caCORE

Software Development Kit 3.2.1

Installation and Basic Test Guide

Center for Bioinformatics

This is a U.S. Government Work

Revised May 10, 2007

Table of Contents

Table of Figures

Introduction

This installation guide outlines the supported configurations and technical installation instructions for the caCORE Software Development Kit (SDK) 3.2.1. Directions for testing the SDK are also included here.

All of the examples and screenshots included in this guide are Windows specific. If you are using a different platform, then modify the information as appropriate for your system.

Overview of caCORE SDK

The caCORE [http://ncicb.nci.nih.gov/infrastructure/cacore_overview] Software Development Kit (SDK) is designed to aid intermediate-level Java programmers with some life science background who are interested in using or extending the capabilities of caCORE. The caCORE SDK is a set of development resources that allows you to create, compile, and run caCORE-like software. For more information about the caCORE SDK, see http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoresdk#documentation.

NOTE:	The caCORE SDK development uses object-oriented programming, primarily Unified Modeling Language (UML). Although installation of the SDK can be performed without an underlying knowledge of UML, you may wish to refer to Appendix A of the caCORE SDK 3.2.1 Programmer's Guide for a brief review of UML in the context of the SDK: http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoresdk#documentation.

caCORE 3.2.1 SDK Minimal System Requirements

Minimal System Requirements

- Internet connection
- Tested Platforms

The caCORE SDK 3.2.1 has been tested on the platforms shown in Table 1.

	Linux Server	Solaris	Windows
Model	HP Proliant ML 330	Sunfire 480R	Dell GX 270
CPU	1 x Intel® Xeon™ Processor 2.80GHz	2 x 1050MHz	1 x Intel® Pentium™ Processor 2.80GHz
Memory	4 GB	4 GB	1 GB
Local Disk	System 2 x 36GB (RAID 1) Data = 2 x 146 (RAID 1)	System 2 x 72GB	System 1 x 36GB
OS	Red Hat Linux ES 3 (RPM 2.4.21-20.0.1)	Solaris 8	Windows XP/2000 Professional

Table 1 Platform testing environment

caCORE SDK Software and Technology Requirements

Software Requirements

Required Software—Not Included in the SDK

You must download and install the required software that is listed in Table 2. This software is *not* included with the caCORE SDK. The software name, version, description, and URL hyperlinks (for download) are indicated in the table.

Required Software—Included in the SDK

See Appendix I caCORE SDK Software and Technology Requirements on page 31.

Software Name	Version	Description	URL
Java 2 Platform Standard Edition (J2SE) 1.4.2 Development Kit (JDK 1.4.2)	jdk1.4.2	The J2SE Software Development Kit (SDK) supports creating J2SE applications	http://java.sun.com/j2se/1.4.2/download.html
UML 1.3 Modeling Tool that produces XMI 1.1 output format	EA 4.50.744 or higher	We recommend using Enterprise Architect (EA)	http://www.sparxsystems.com.au/

Table 2 Required software and technology for the SDK

Optional Software

Optional software to use with the caCORE SDK is listed in Table 3. The included (**Incl.**) column indicates (with a **Yes**) if the software is packaged with the SDK; **No** indicates that you must supply the software. A hyperlink is included for your reference to appropriate sources.

Software Name	*Version*	*Description*	*URL*	*Incl.*
Eclipse IDE	3.0 or higher	An open platform integration tool that provides tool developers with flexibility and control over their software technology.	http://www.eclipse.org/downloads/index.php	No
jeteditor-eclipse plugin	0.0.1-alpha-2004-07-22	JET-Editor is an Eclipse-based Editor for JET-templates (templates used by EMF). It is intended to support the development of JET-templates in a quality that is adequate to other eclipse language support.	http://sourceforge.net/projects/jet-editor	Yes
Oracle 9i Database	9i	Oracle is a commercially-available, alternative relational database management system (RDBMS) that can be used in place of MySQL. Release 9i is currently the only release supported by the SDK.	http://www.oracle.com/technology/products/oracle9i/index.html	No
IBM DB2	8.2	Like Oracle, DB2 is another commercially-available alternative to MySQL.	http://www-306.ibm.com/software/data/db2/	No

Table 3 Optional software and technology for the SDK

Additional Software

The caCORE SDK requires a Java 2 container in which the server component can run. Several different open source and commercial applications are available. Table 4 lists the applications that have been tested and are known to work with the SDK.

Software Name	*Version*	*URL*	*Notes*
Apache Tomcat	5.5.20	http://tomcat.apache.org	The SDK can install Tomcat as part of the build process.
JBoss Application Server	4.0.4.GA	http://www.jboss.org	See the JBoss documentation for instructions on installation and configuration.

Table 4 Additional software for the SDK

RDBMS Software

The caCORE SDK typically also requires a relational database management system (RDBMS) for data persistence. The SDK provides support for the RDBMS software listed in Table 5.

Software Name	Version	URL	Notes
MySQL	4.1.19	http://dev.mysql.com/downloads/mysql/4.1.html	The SDK can install MySQL as part of the build process.
Oracle 9i	9i	http://www.oracle.com/technology/software/products/oracle9i/index.html	Oracle is commercial software. Release 9i is currently the only release supported by the SDK.
IBM DB2	8.2	http://www-306.ibm.com/software/data/db2/	DB2 is commercial software.

Table 5 RDBMS software for the SDK

NOTE:

Drivers for MySQL, Oracle 9i and DB2 are included with the SDK. If you are using a different version of Oracle or DB2, you must obtain the appropriate drivers. JDBC drivers can be downloaded from the Sun Developer Network at http://developers.sun.com/product/jdbc/drivers/index.html, or from the individual vendors' sites (for example, the Oracle 8i driver `classes12.zip` can be downloaded from http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html). These drivers should be placed in the `{project_home}/lib` directory and the appropriate directory in your J2SE container (e.g., Tomcat, JBoss) to enable connection to the appropriate database. In addition, some manual modification of the Hibernate configuration files may be necessary.

Additional Optional Software

Additional optional software for the SDK is listed in Table 6. The **included** column indicates whether the software is packaged with the SDK. A hyperlink is included for your reference to appropriate sources.

Software Name	Version	Description	URL	Incl.
Eclipse IDE	3.0 or higher	An open platform for tool integration that provides developers with flexibility and control over their software technology.	http://www.eclipse.org	No
jeteditor-eclipse plugin	0.0.1-alpha-2004-07-22	JET-Editor is an Eclipse-based editor for Java Emitter Template (JET) technology.	http://jet-editor.sourceforge.net/	No

Table 6 Additional optional software for the SDK

Documentation and Style Tools

The following tools, useful for documentation and source code styling, are part of the SDK framework.

Useful tools included in the SDK

- **Javadoc** – Execute the Ant task `doc` to generate Javadocs for your beans. Your javadocs will be generated to the `{home_directory}/output/{project_name}/doc` directory. For more information on Javadoc, see http://java.sun.com/j2se/javadoc/.
- **Jalopy** – Execute the Ant task `format` to make your code well formatted. The default indentation format is used in the SDK. This task is configurable to enforce coding standards that you wish to adhere to for your project.
  See http:// jalopy.sourceforge.net/manual.html for information on how to customize this task.

NOTE:	See also Appendix II caCORE SDK Components on page 37.

Upgrading to caCORE SDK 3.2.1 from Previous SDK Versions*

caCORE SDK Users Upgrading to SDK 3.2.1	*All users of previous SDK versions are encouraged to upgrade to the SDK 3.2.1. For details on the new features in this version, see the Release Notes that accompany the downloadable archive.

Use the following steps to download and install the caCORE SDK 3.2.1 upgrade:

Step	Action
1	Follow the directions for downloading the caCORE SDK 3.2.1 on page 11. If you have previously installed SDK 1.0.x or 3.1, to upgrade to SDK 3.2.1, unzip the SDK 3.2.1 to a new directory.

Step	Action
2	If you are using an existing MySQL database, be sure to set the parameters in the `deploy.properties` file as described in Table 7 so you do not force a reinstallation of MySQL and reloading of the example database. **Note:** The default `project_name` for the SDK 3.2.1 is `example`. If you choose to replace that name with one used in previous SDK versions, ensure that the `project_name` is consistent for all parameters described in this installation and test guide.
3	Follow installation and testing instructions for the new SDK beginning on page 6.
4	**Note:** The `output` directory in SDK 3.2.1 is not created until after you run `build-system` or a `similar Ant target`, so do not delete SDK 1.0.x or 3.1 files until you know the SDK 3.2.1 has a successful build. • If you have created your own models in the previous SDK, copy all of your models within your `{OLD_SDK_HOME}/models` directory to your `{NEW_SDK_HOME}/models` directory and all XMI files from `{OLD_SDK_HOME}/models/xmi` to `{NEW_SDK_HOME}/models/xmi`. • Copy any EVS reports or annotated XMI files you want to save that are in the old version. After you have successfully installed SDK 3.2.1, you can delete the SDK 1.0.x directory from your computer.

Installing the caCORE SDK

Preliminary Considerations

⚠ **BEFORE YOU BEGIN**	The SDK has been tested with the operating systems and hardware specified on pages 1 and 3 of this guide. We cannot guarantee the SDK will work if you are using variations of these operating systems and/or hardware.

caCORE Sample Object Model

To demonstrate the SDK, a small caCORE sample object model is included with the SDK. The sample model, as shown in Figure 1, contains seven of the core objects from the `gov.nih.nci.cabio.domain` package.

NOTE:	Figure 1 is a UML class diagram. For more information about UML, see Appendix A in the *caCORE SDK 3.2.1 Programmer's Guide*, available from: http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoresdk#documentation.

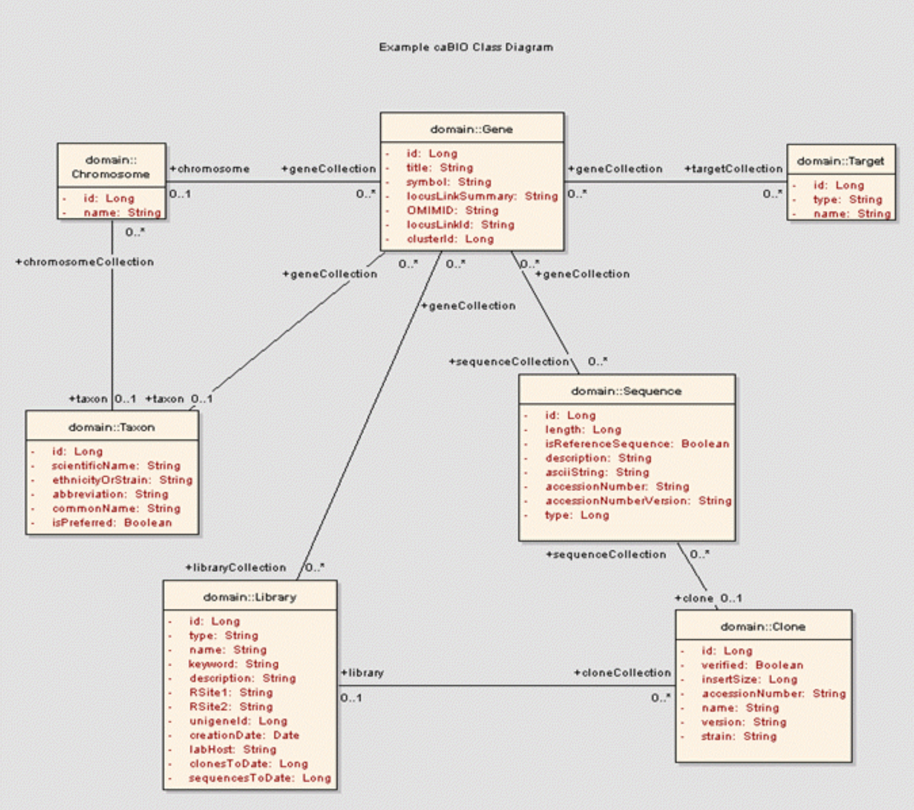


Figure 1 caBIO class diagram

Also, as of v3.2.1 of the SDK, there is now a second sample EA model file, SDKTestModel.eap that illustrates most of the various allowable class relationships supported by the SDK code generator. This includes generalization (inheritance) and uni- and bi-directional associations (many-to-many, many-to-one, one-to-many, and one-to-one), as illustrated in the following figures:

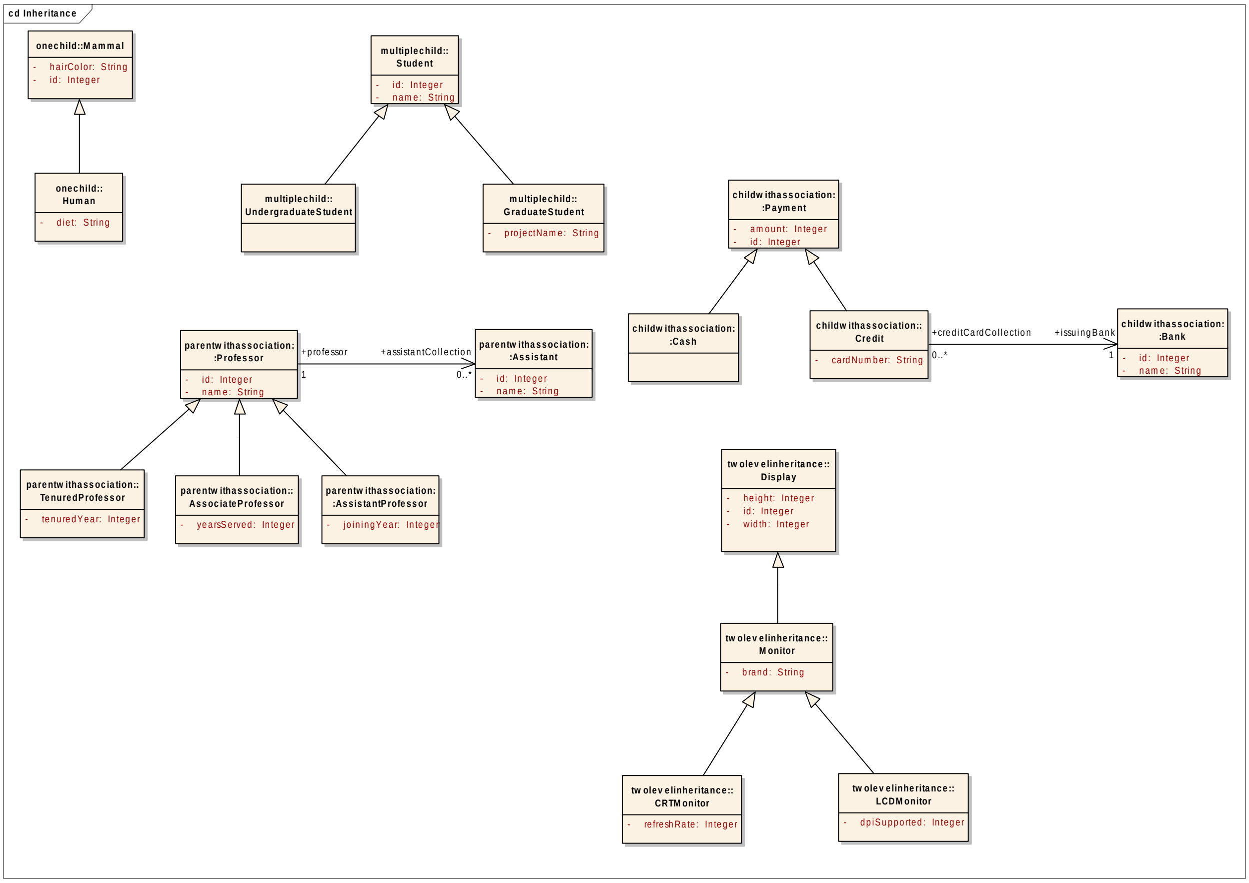


Figure 2 Inheritance Sample class diagram

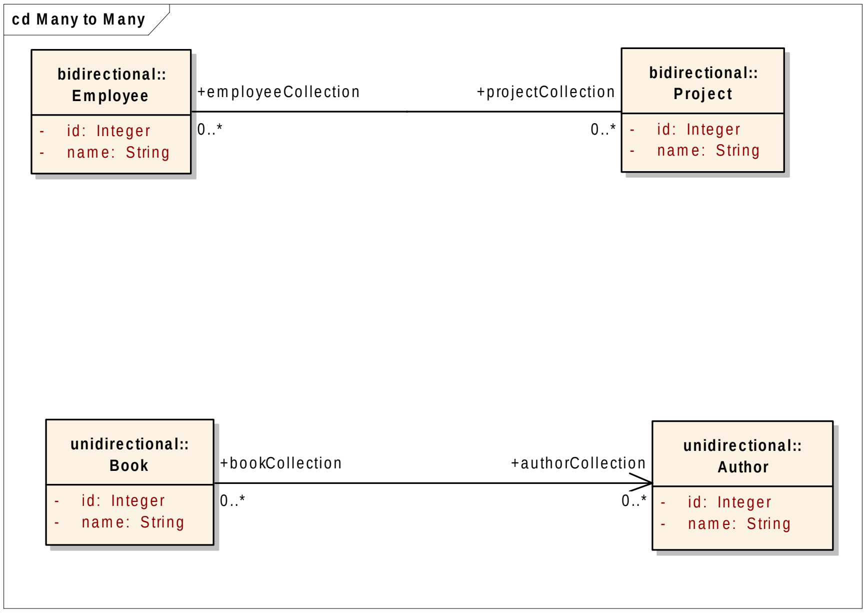


Figure 3 Many-to-Many Sample class diagram

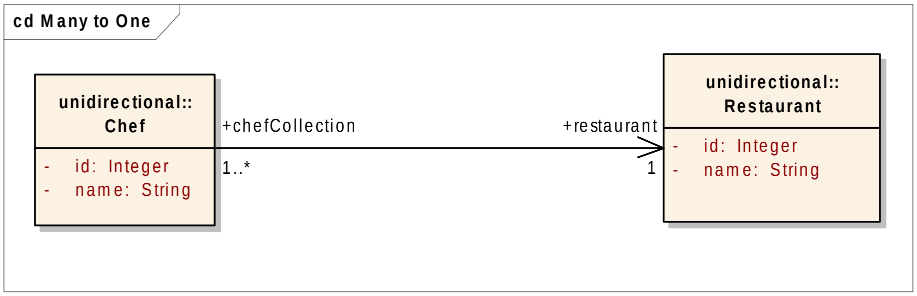


Figure 4 Many-to-One sample class diagram

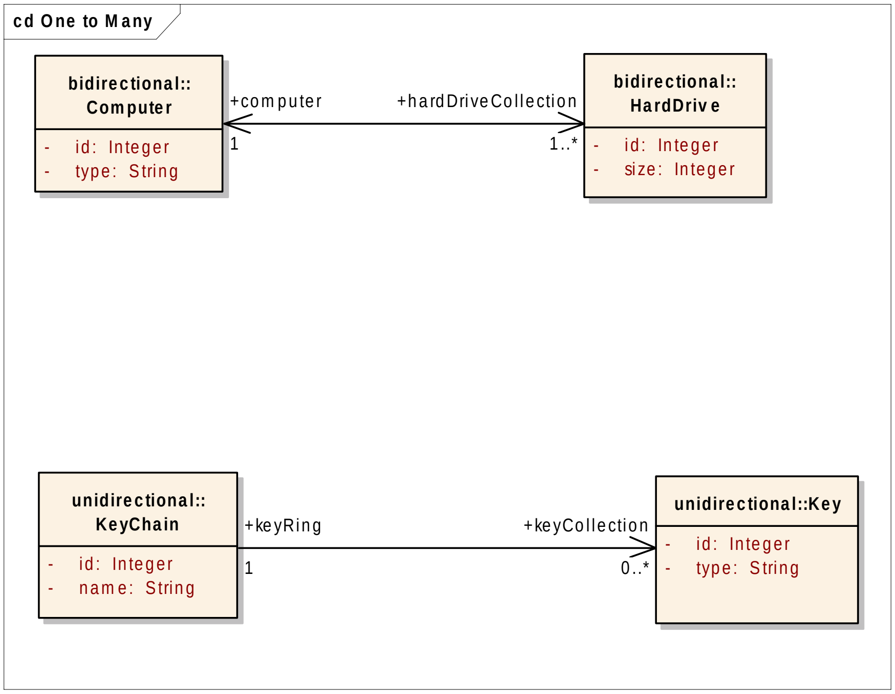


Figure 5 One-to-Many sample class diagram

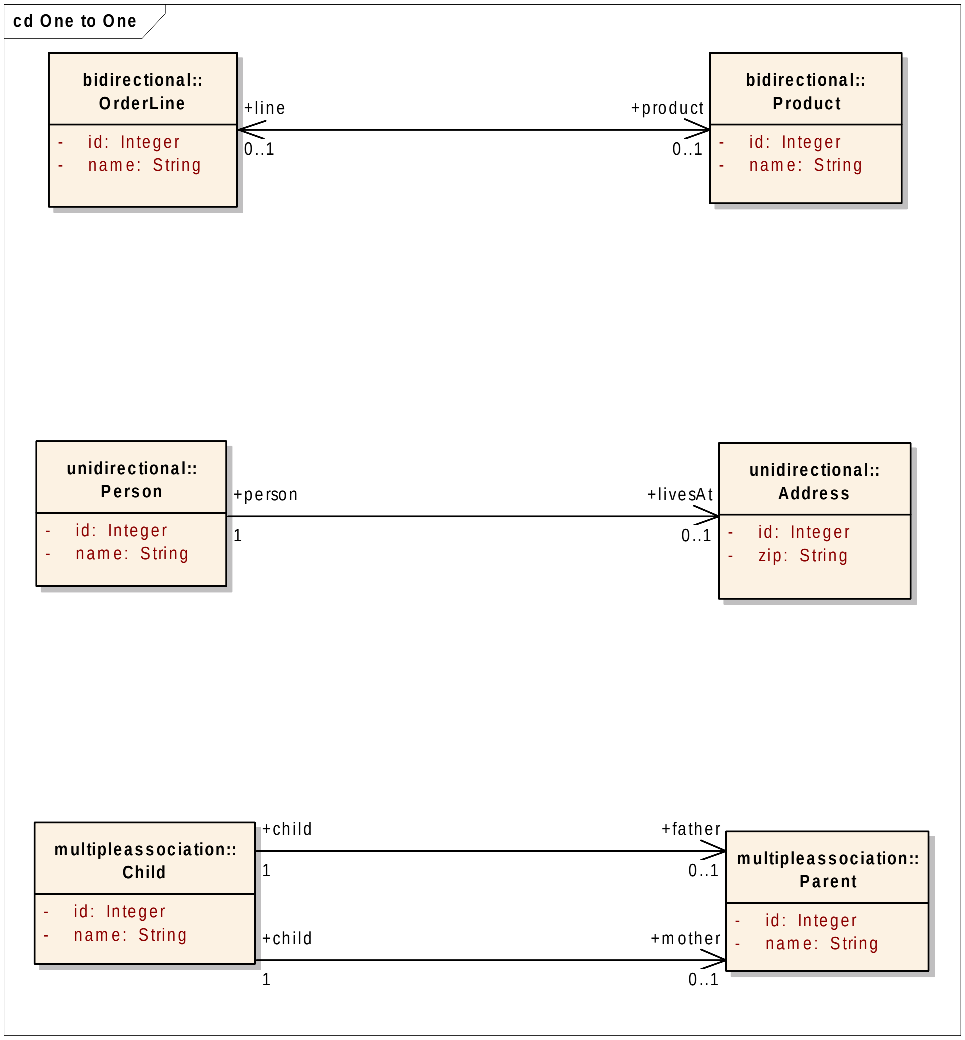


Figure 6 One-to-One sample class diagram

Also included in this Test model are examples of the supported data types, including Boolean, Date, Double, Float, Integer, and String. There are also examples of how to model Primary Keys using Double, Float, Integer, or String ID, as illustrated in the following figure:

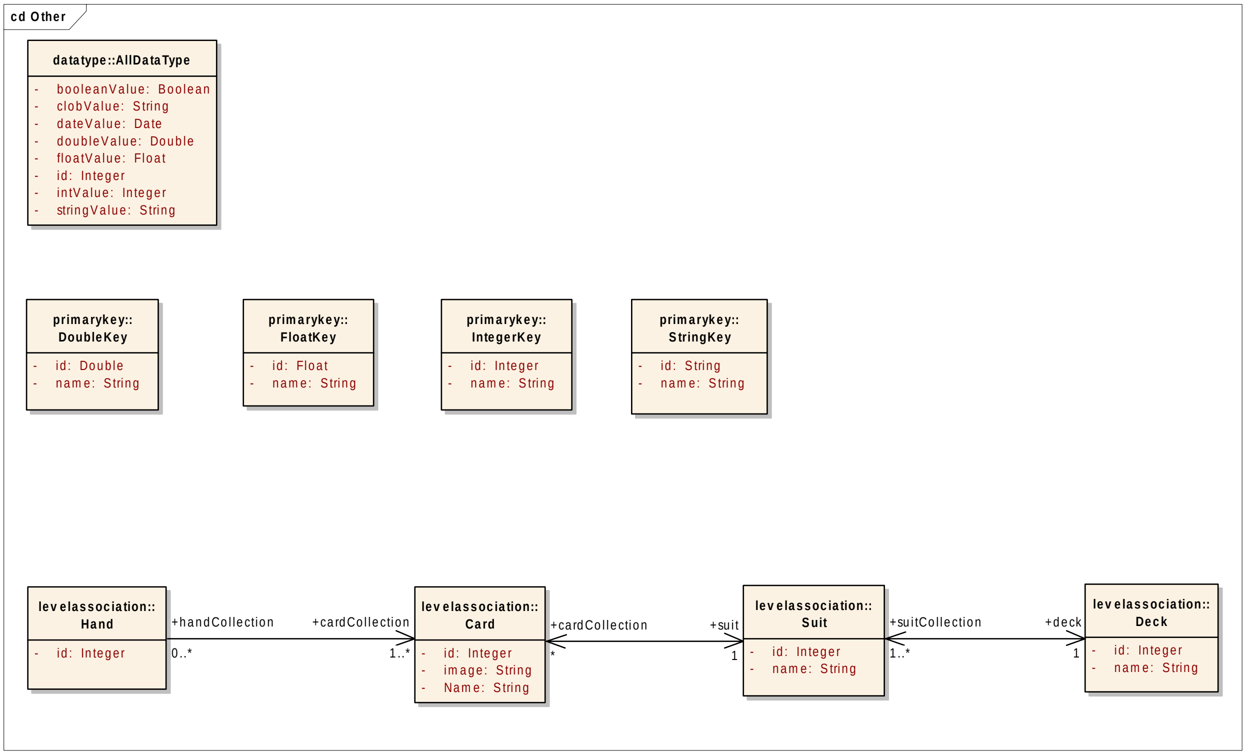


Figure 7 Supported Data Types, and Primary Key sample class diagram

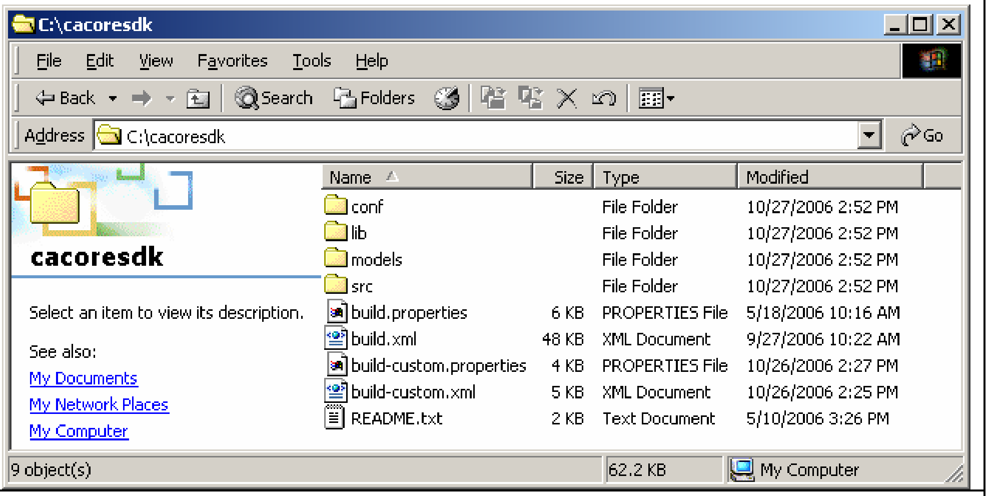
NOTE:	For more information about relationships between classes, as well as and UML in general, see Chapter 5, Creating UML Models, as well as Appendix A in the *caCORE SDK 3.2.1 Programmer's Guide*, available from: http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoresdk#documentation.

After successfully installing the SDK and executing the sample demonstration, as described in this guide, you should be ready to start customizing the SDK to produce output tailored for your specific needs with your specific database connections. If you have the required software installed on your system (see the previous section), then it should not take more than ten minutes to install and run this example.

Downloading the caCORE SDK

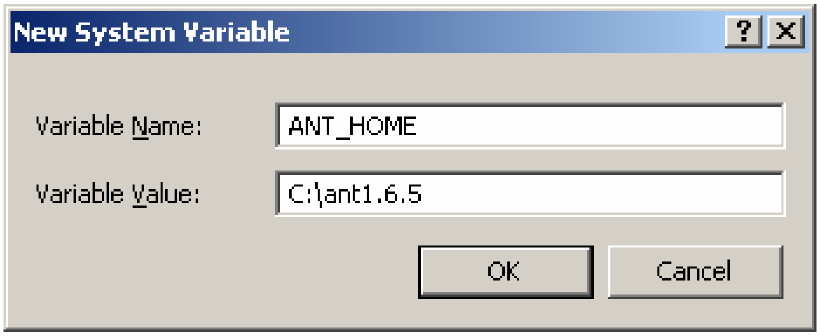
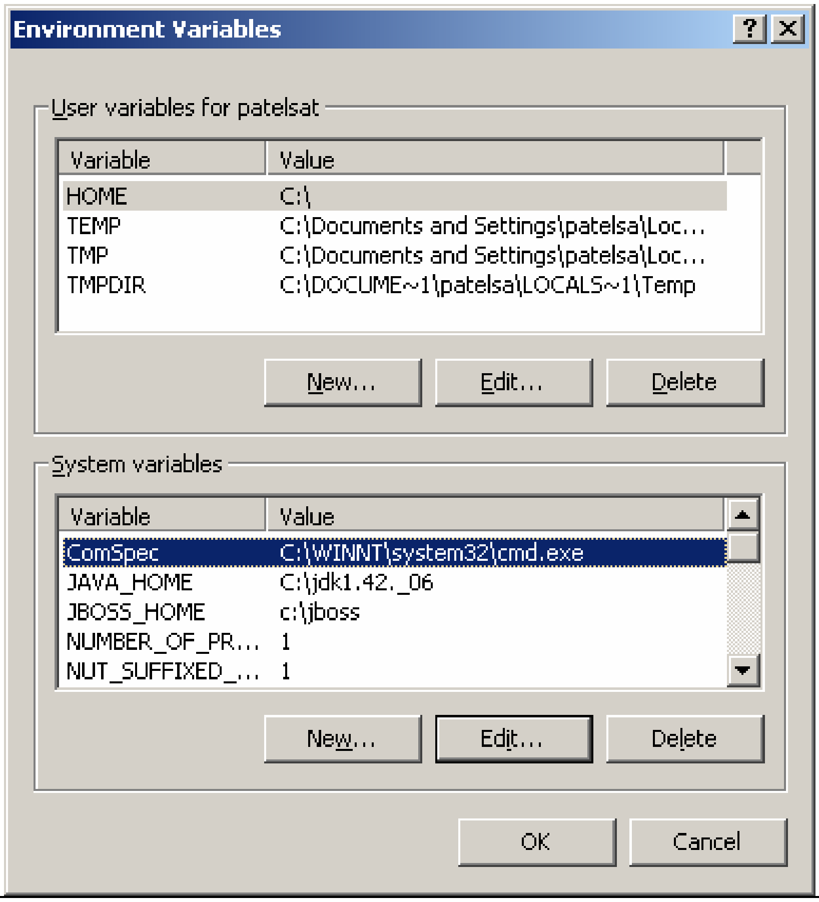
SUGGESTION	To best understand the installation and testing procedures for the SDK, it is recommended that you follow the procedures described in this section with minimal deviation.

Complete the following steps to download the caCORE SDK:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Go to the NCICB download web site http://ncicb.nci.nih.gov/download/ and click on the link to download the caCORE SDK.</td></tr>
<tr><td>2</td><td>Provide your email, name, and institution. Click <b>Enter the Download Center</b>.</td></tr>
<tr><td>3</td><td>Select <b>caCORE SDK</b> and agree to the caCORE software license by selecting <b>Checking this box indicates that you agree to the above terms</b>.</td></tr>
<tr><td>4</td><td>Under <b>Source Distribution</b>, select the appropriate caCORE SDK file for your platform (<b>Windows</b> or <b>Unix/Linux</b>) and save it to a temporary directory on your computer (for example, <code>C:\temp</code> in Windows). This may take several minutes.</td></tr>
<tr><td>5</td><td>Extract the contents of the caCORE distribution zip file to your root directory. For example, if your root directory in Windows is <code>C:\</code>, then your caCORE home directory is <code>C:\cacoresdk</code>.
<b>Note:</b> This directory name cannot contain any spaces or this will cause errors in Windows. For instance, do not place the caCORE SDK directory in the Program Files directory.
The following figure displays the extracted SDK.


Address: C:\cacoresdk
<table>
<tr><th>Name</th><th>Size</th><th>Type</th><th>Modified</th></tr>
<tr><td>conf</td><td></td><td>File Folder</td><td>10/27/2006 2:52 PM</td></tr>
<tr><td>lib</td><td></td><td>File Folder</td><td>10/27/2006 2:52 PM</td></tr>
<tr><td>models</td><td></td><td>File Folder</td><td>10/27/2006 2:52 PM</td></tr>
<tr><td>src</td><td></td><td>File Folder</td><td>10/27/2006 2:52 PM</td></tr>
<tr><td>build.properties</td><td>6 KB</td><td>PROPERTIES File</td><td>5/18/2006 10:16 AM</td></tr>
<tr><td>build.xml</td><td>48 KB</td><td>XML Document</td><td>9/27/2006 10:22 AM</td></tr>
<tr><td>build-custom.properties</td><td>4 KB</td><td>PROPERTIES File</td><td>10/26/2006 2:27 PM</td></tr>
<tr><td>build-custom.xml</td><td>5 KB</td><td>XML Document</td><td>10/26/2006 2:25 PM</td></tr>
<tr><td>README.txt</td><td>2 KB</td><td>Text Document</td><td>5/10/2006 3:26 PM</td></tr>
</table>
9 object(s) 62.2 KB</td></tr>
<tr><td>6</td><td>If you do not already have Apache Ant installed, extract the contents of the Ant zip file, <code>apache-ant-1.6.5-bin.zip</code> and save it to your computer. For example, if your root directory in Windows is <code>C:\</code>, then extract your Ant home directory to <code>C:\apache-ant-1.6.5</code>.</td></tr>
</table>

Step	Action
7	Verify that the environment variables ANT_HOME and JAVA_HOME are set and that your PATH statement includes the locations of the Ant and Java binaries. For example, PATH should include `C:\apache-ant-1.6.5\bin; C:\j2sdk1.4.2_06\bin`. To do this in Windows: a. Select **My Computer > Properties** (from the shortcut menu) **> Advanced** tab. b. Click the **Environment_Variables...** button. c. **ANT_HOME** and **JAVA_HOME** must each be listed in one or the other of the panels of the dialog box. To add either, below one of the panels, click the **New** button. d. In the New User Variable dialog box, add the **Variable** and **Variable Value** (see example New User Variable dialog box below). Examples: **Variable** = ***ANT_HOME***; **Variable Value** = ***C:\apache-ant-1.6.5*** **Variable** = ***JAVA_HOME***; **Variable Value** = ***C:\j2sdk1.4.2_06*** f. Verify that the **PATH** statement listed in the Environment Variables dialog box includes both the **ANT_HOME** and **JAVA_HOME** binaries, for example: ***C:\j2sdk1.4_2_06\bin;C:\apache-ant-1.6.5\bin***. (To easily review these, copy them to a text editor.) If necessary, click the **Edit** button and enter the correct information (see example Environmental Variables dialog box below).

Step	Action
7 (con't)	 

Variable Name: ANT_HOME

Variable Value: C:\ant1.6.5

Variable	Value
HOME	C:\
TEMP	C:\Documents and Settings\patelsa\Loc...
TMP	C:\Documents and Settings\patelsa\Loc...
TMPDIR	C:\DOCUME~1\patelsa\LOCALS~1\Temp

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
JAVA_HOME	C:\jdk1.42._06
JBOSS_HOME	c:\jboss
NUMBER_OF_PR...	1
NUT_SUFFIXED_...	1

Running the Example

The SDK example is configured to work with MySQL 4.1.19; however, you must assign the appropriate values for the database properties in the `deploy.properties` file.

NOTE:

When entering information for use by the SDK, no spaces are allowed in filenames or directory names.

Modifying the deploy.properties File in Windows

- The file `{home_directory}/conf/deploy.properties` contains properties values that must be updated for your system environment. These values are used during the installation step to indicate the location and version of software.
- For the purpose of running the example, we recommend that you only modify the properties listed in both Table 7 and Table 8, as described. Otherwise, leave the `deploy.properties` unaltered.
- **Note:** Do not modify the forward slashes (/) in the `deploy.properties` file, even when installing on Windows.

Property Name	***Description***
`install_tomcat`	Specify **yes** to install Tomcat 5.5.20 (http://jakarta.apache.org/tomcat/) or specify **no** if Tomcat or JBoss is already installed. **Note:** If you specify **yes**, any previous versions of Tomcat may be overwritten, which could adversely affect programs running on your computer.
`j2se_container_home`	Provide the home directory for your J2SE container (for example, in Windows, `C:\tomcat or c:\jboss-4.0.4.GA`). If you instruct the SDK to install Tomcat, this property defines the directory in which it will be installed. **Note:** This directory name cannot contain any spaces or this will cause errors.
`mysql_home`	Provide the home directory for MySQL (for example, in Windows, `C:\mysql`). If you instruct the SDK to install MySQL, this property defines the directory in which it will be installed. **Note:** This directory cannot contain any spaces or this will cause errors. See Table 8 and User MySQL Download on page 18 for configuring other MySQL properties.
`project_name`	The default `project_name` for the SDK 3.2.1 is `example`. If you choose to replace the name, be sure and use your user-defined name for all `project_name` parameters described in this installation and test guide.

Property Name	Description
`default_security_level`	Determines whether the security provided through CSM is enabled or disabled. 1 indicates security on by default; 0 indicates security off by default.
`application_name`	Application Context Name used for CSM security. This should be the same name used in the UPT.
`default_session_timeout`	Determines the default timeout, in milliseconds, for client sessions when security is enabled.
`disable_writable_api_generation`	Determines whether writable APIs should be disabled or enabled for this application. If this value is set to “yes”, writable APIs are disabled. To turn write functionality on, set this property value to “no”.
`authentication_disabled`	Determines if authentication should be used when security is enabled. If turned “ON” it disables the authentication when security is turned on. The password supplied by the client while calling the startSession method on ClientSession object is not used anymore. Hence the client can pass any string as password. However the user name is needed to establish the session for Authorization.
`csm_database_url`	The fully qualified JDBC URL to database/schema hosting the CSM Authorization Schema for the application. Note: this database could be different than the actual application database/schema hosting the application data or could be the same if the CSM tables are hosted in the same database/schema. For example, jdbc:oracle:thin:@foo.bar.com:1521:foodatabase
`csm_database_user_name`	The user name for the CSM’s Authorization database.
`csm_database_password`	The password for the CSM’s Authorization database.
`csm_database_dialect`	The hibernate dialect to be used while connecting to the CSM’s Authorization Database, for example org.hibernate.dialect.OracleDialect.
`csm_database_driver`	The JDBC driver to be used while connecting to the CSM’s Authorization Database. For example oracle.jdbc.driver.OracleDriver.

Table 7 Required deploy.properties values

NOTE:	When entering information for use by the SDK, no spaces are allowed in filenames or directory names.

Modifying the deploy.properties File in UNIX or Linux

The file `{home_directory}/conf/deploy.properties` contains property values that must be updated for your UNIX or Linux system environment. These values are used during the installation step to indicate the location and version of software. For example, make sure to update the pathnames to point to the location of MySQL, etc., on your computer.

MySQL Database Issues

Select from the four scenarios described below regarding MySQL, and modify the `deploy.properties` file appropriately. (We recommend you use scenario 1a, allowing the SDK to download and install MySQL.)

Scenario	*1* *a. You have not downloaded MySQL,* *or* *b. You have an installed MySQL and want the SDK to reinstall it.*	*2* *You do have an installed MySQL but want to create a new user and schema*	*3* *You do have an installed MySQL and want to use an existing user and schema.*
`install_mysql`	Yes	No	No
`db_user`	User name to create for your MySQL database	User name to create for your MySQL database	Existing user name
`db_password`	Password to create for your MySQL database	Password to create for your MySQL database	Existing password
`schema_name`	Name of schema to create	Name of schema to create	Name of existing schema
`db_server_name`	Fully qualified server name of the machine where mySQL will be installed. *Example:* `localhost`	Fully qualified server name of the machine where mySQL is installed. *Example:* `localhost`	Fully qualified server name of the machine where mySQL is installed. *Example:* `localhost`

Scenario	*1* *a. You have not downloaded MySQL,* *or* *b. You have an installed MySQL and want the SDK to reinstall it.*	*2* *You do have an installed MySQL but want to create a new user and schema*	*3* *You do have an installed MySQL and want to use an existing user and schema.*
`create_mysql_user`	Yes	Yes	No
`create_schema`	Yes	Yes	No
`install_data`	Yes	Yes	No

Table 8 MySQL properties in the deploy.properties file

User MySQL Download

If you would like to download and install MySQL yourself and are using Microsoft Windows, observe the following guidelines to ensure that your installation is compatible with the caCORE SDK.

Perform the following steps to download and install MySQL:

Step	Action
1	When using the MySQL installation wizard, select the **Custom** setup type (do NOT choose **Typical**).
2	In the Custom Setup screen of the wizard, change the installation path to a location that does not include any spaces, for example `C:\mysql`.
3	After the wizard has finished installing MySQL, you will be prompted to sign up for a **MySQL.com** account. This step is optional (most users will want to choose **Skip Sign-Up**).
4	The final screen of the setup wizard gives you the option to configure the MySQL server. Make sure this box is checked before pressing **Finish**.
5	If you are not sure about how to configure your instance of MySQL, step through the configuration wizard and use the default values already selected. However, on the screen allowing you to choose the Windows options, UNCHECK the option to Install as Windows Service.

If you have MySQL installed but want to create a new user or use an existing user, make sure that you have granted privileges to your user to access the desired schema:

Step	Action
1	In a Command Prompt window, enter ***cd {mysql_home}*** to go to your MySQL home directory. **Note:** `mysql_home` is defined in the `deploy.properties` file. *Example:* ***cd C:\mysql***
2	Enter ***cd bin*** to go to the bin directory.
3	Enter ***mysql*** to execute the MySQL monitor.
4	Enter ***grant all privileges on {schema_name}.* to '{db_user}@'{web_server_name}' identified by '{db_password}' with grant option;*** **Notes:** • `schema_name, db_user, web_server_name` and `db_password` are defined in the `deploy.properties` file. • The semi-colon is an important part of the command--it prompts its execution.
5	For example, for the example system, enter the following commands: ***grant all privileges on cacore.* to 'cabiouser'@'localhost' identified by 'cabio' with grant option;*** ***grant all privileges on cacore.* to 'cabiouser'@'*' identified by 'cabio' with grant option;***

Building a Demonstration System

Perform the following steps to build your system.

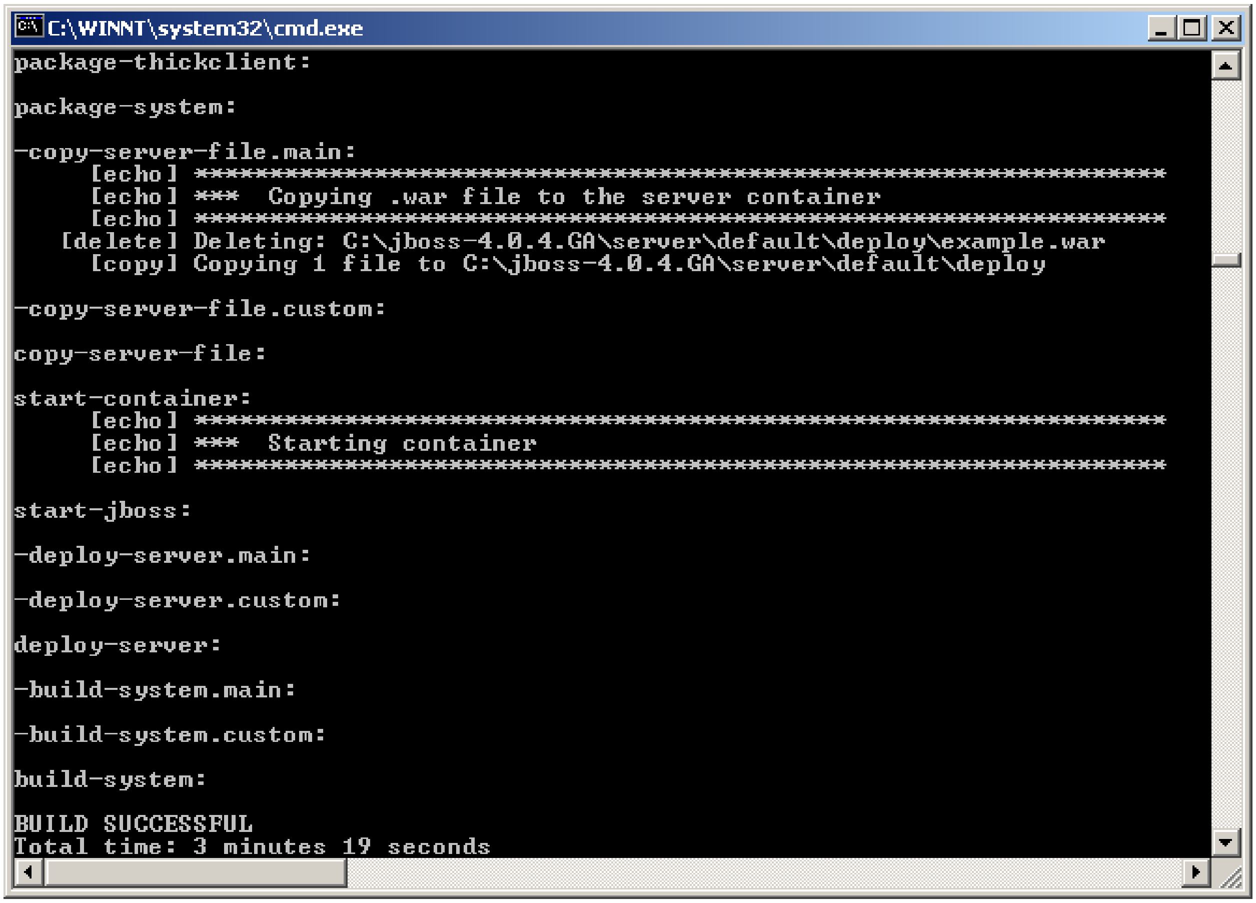
Step	Action
1	In a **Command Prompt** window, enter ***cd {home_directory}*** to go to your home directory (for example, in Windows: `C:\cacoretoolkit`).
2	Enter ***ant build-system***.

3	After your web application server has completely finished starting, run the following command to deploy web services: ***`ant deployWS`***.

NOTE:	These commands should only be run once with the `deploy.properties` file as you have configured it. After you have run `build-system` the first time, you must update the `deploy.properties` file so that software is not reinstalled on subsequent executions of `build-system`.

Ant messages display as each task is processing. The `build-system` task builds the entire system and deploys the software to the webapp directory of the web application server installation specified in the `deploy.properties` file.

Figure 8 displays the last few tasks of a successful build.

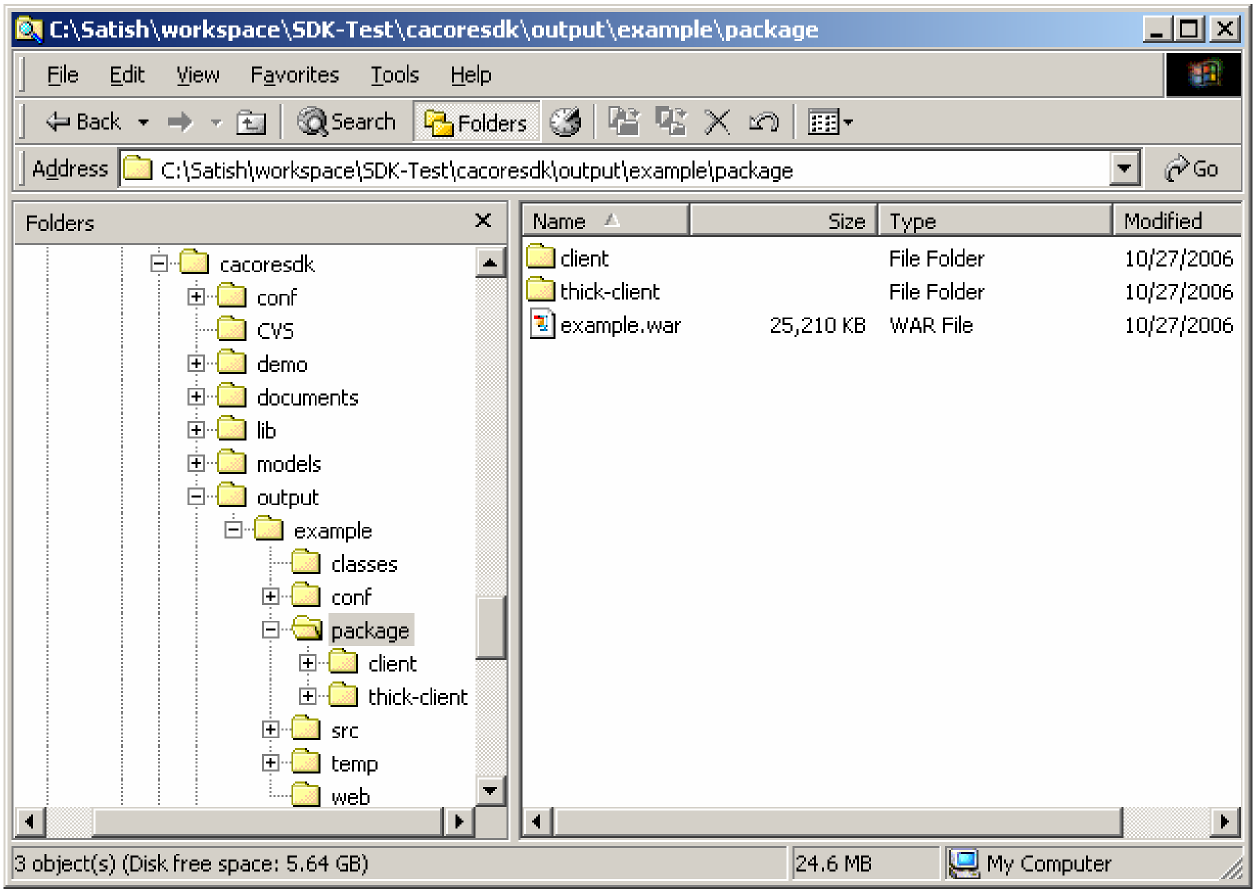


```
package-thickclient:

package-system:

-copy-server-file.main:
     [echo] *****************************************************************
     [echo] ***  Copying .war file to the server container
     [echo] *****************************************************************
   [delete] Deleting: C:\jboss-4.0.4.GA\server\default\deploy\example.war
     [copy] Copying 1 file to C:\jboss-4.0.4.GA\server\default\deploy

-copy-server-file.custom:

copy-server-file:

start-container:
     [echo] *****************************************************************
     [echo] ***  Starting container
     [echo] *****************************************************************

start-jboss:

-deploy-server.main:

-deploy-server.custom:

deploy-server:

-build-system.main:

-build-system.custom:

build-system:

BUILD SUCCESSFUL
Total time: 3 minutes 19 seconds
```

Figure 8 Successful build-system task

NOTE TO WINDOWS USERS :	Command screens may pop up during the build that indicate that MySQL and Tomcat or JBoss are running. If you plan to continue working with the SDK, or proceed through the testing procedures in this chapter, be sure and leave those windows open. Closing them kills the associated applications and prevents the generated system from functioning.

Figure 9 illustrates the directory structure resulting from running the `build-system` task. This task compiles the SDK framework classes, generates and compiles the domain objects represented by the UML model of the system, generates the Object Relational Mapping (ORM) configuration files and a number of other system artifacts, and finally packages files for the server-side (`server.war`) and client side (`client.jar`) and its resources, `thick-client.jar` and its resources). Most of these files are found in the output directory and subdirectories shown in Figure 9.



C:\Satish\workspace\SDK-Test\cacoresdk\output\example\package

Address: C:\Satish\workspace\SDK-Test\cacoresdk\output\example\package

Name	Size	Type	Modified
client		File Folder	10/27/2006
thick-client		File Folder	10/27/2006
example.war	25,210 KB	WAR File	10/27/2006

3 object(s) (Disk free space: 5.64 GB) | 24.6 MB | My Computer

Figure 9 Generated directory structure

Executing Tests

Tests must be performed to make sure your system is functioning properly. Perform the following system tests to verify a successful installation.

WARNING TO WINDOWS USERS:	Command screens may pop up during the build that indicate that MySQL and Tomcat or JBoss are running. You must leave those windows open as you execute the SDK system tests. Closing them kills the associated applications.

Testing the Database

Perform the following steps to the database schema and verify data was loaded.

Step	Action
1	In a Command Prompt window, enter ***cd {mysql_home}*** to go to your MySQL home directory. **Note:** mysql_home is defined in the deploy.properties file. *Example:* ***cd C:\mysql***
2	Enter ***cd bin*** to go to the bin directory.
3	Enter ***mysql*** to execute the MySQL monitor.
4	Enter ***select * from {schema_name}.{table_name};.*** **Notes:** • schema_name is defined in the deploy.properties file. *Example:* ***cacore*** • The semi-colon is an important part of the command--it prompts its execution.
5	For the initial example, enter ***select * from cacore.gene where gene_symbol like 'IL%';***. There should be 9 rows displayed.

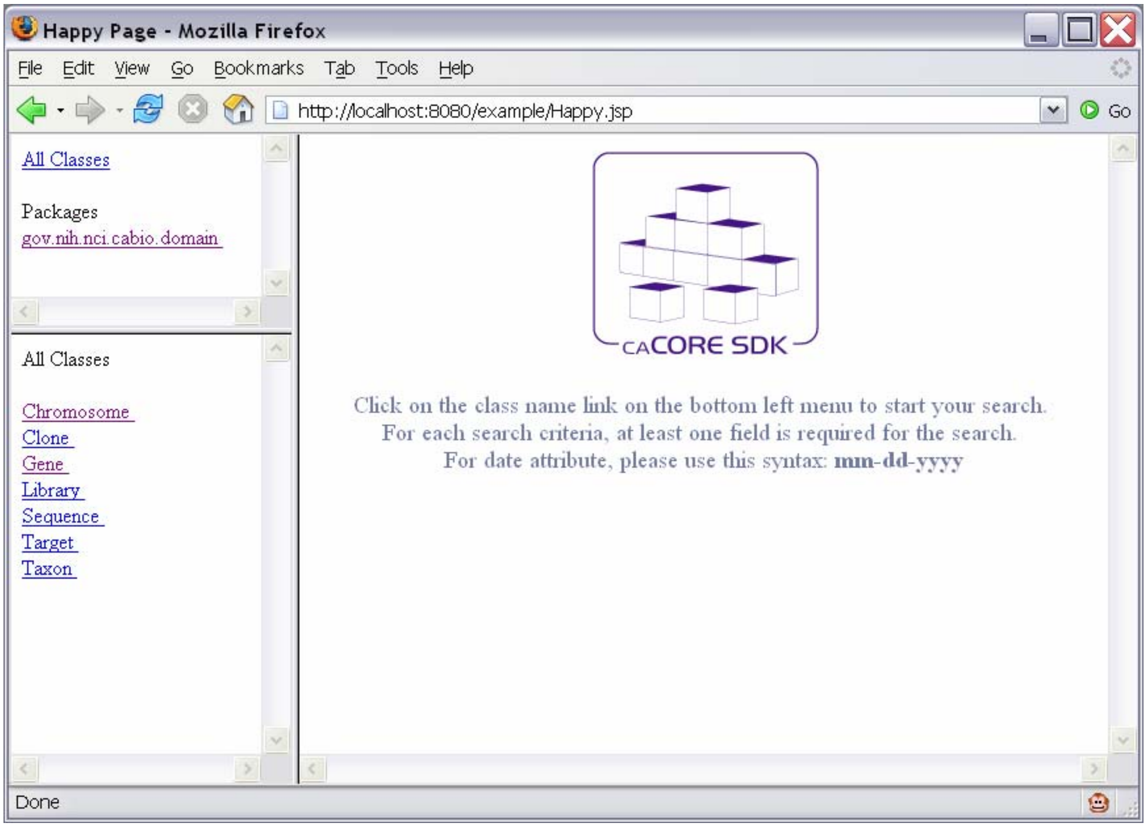
If you see data, then your database load was successful

Testing the System

Perform the following steps to test the system.

Step	Action
1	Enter the following URL in your browser to verify all your required system resources are available: http://{web_server_name}:8080/{project_name}/Happy.jsp. **Note:** project_name and web_server_name are defined in the deploy.properties file. *Example:* project_name = ***example;*** web_server_name = localhost
2	For the getting started example, enter: http://localhost:8080/example/Happy.jsp

Figure 10 displays the browser window that opens when the system has been properly built.


Happy Page - Mozilla Firefox
File Edit View Go Bookmarks Tab Tools Help
http://localhost:8080/example/Happy.jsp
Go
caCORE SDK
Done

All Classes

Packages
gov.nih.nci.cabio.domain

All Classes

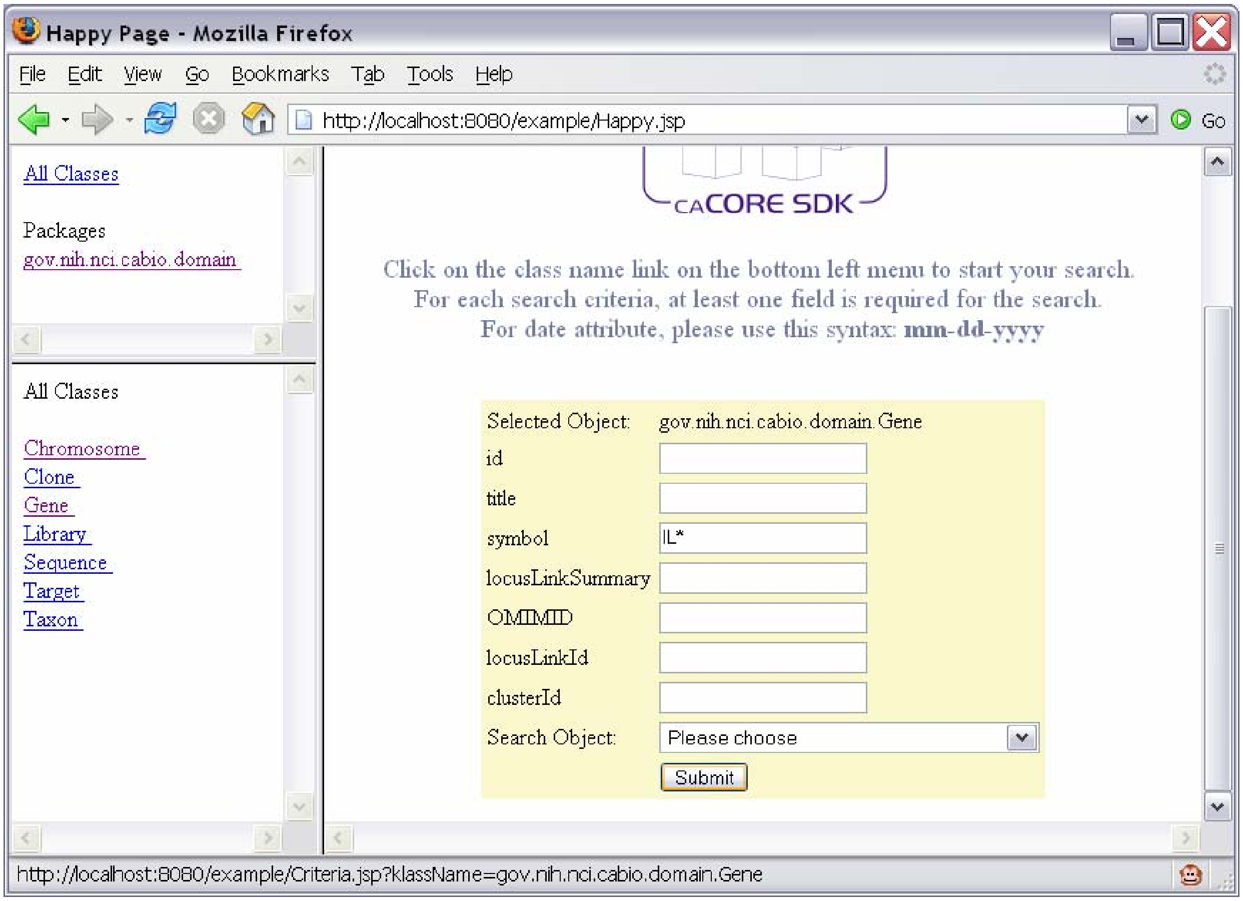
Chromosome
Clone
Gene
Library
Sequence
Target
Taxon

Click on the class name link on the bottom left menu to start your search.
For each search criteria, at least one field is required for the search.
For date attribute, please use this syntax: **mm-dd-yyyy**

Figure 10 Browser display with successful system test

The Happy.jsp page provides a simple query interface that can be used to test the system and ensure that data has been correctly loaded. Perform the following steps to test the system:

Step	Action
1	In the lower-left window, select the Gene link
2	A query form displays in the main window. Enter **IL*** in the Symbol field and press Submit (Figure 11).
3	A new window should appear that displays nine objects that match the query you just submitted (Figure 12). In addition to displaying the attributes of each of these, you can also navigate to associated objects by clicking the links in each row.



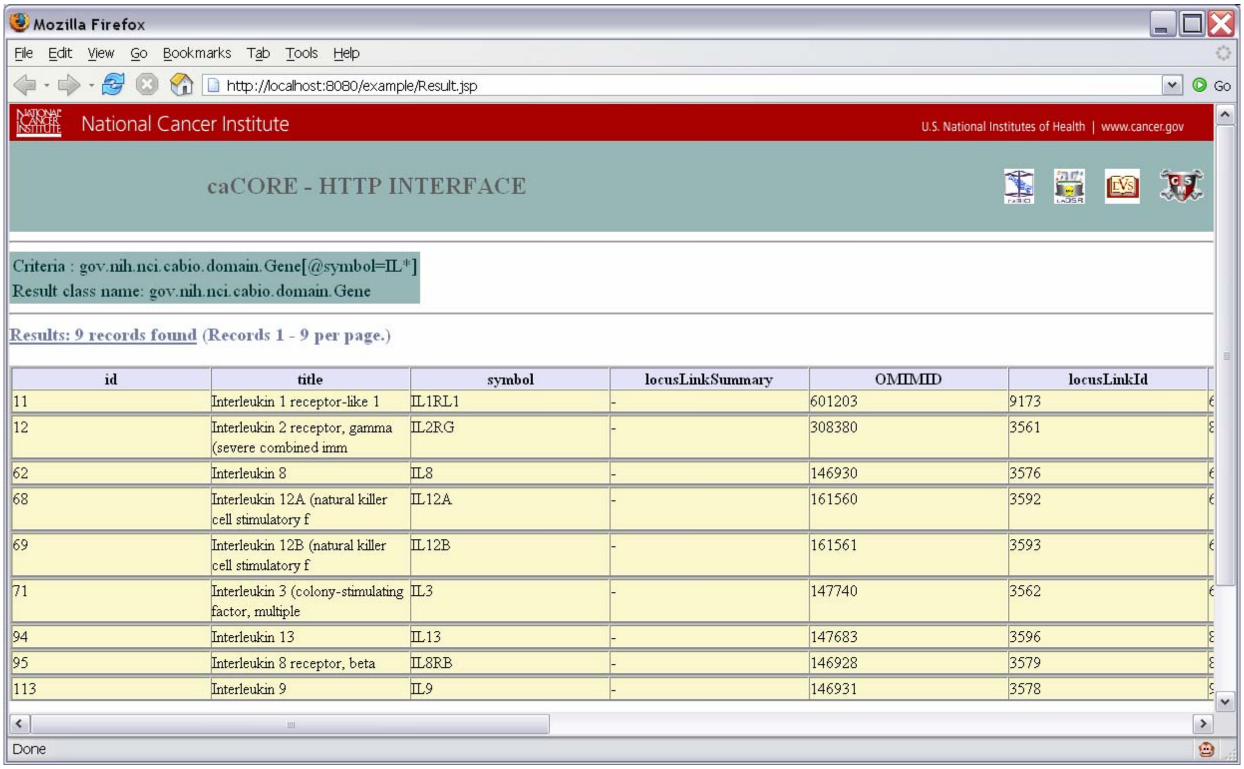
http://localhost:8080/example/Happy.jsp

Click on the class name link on the bottom left menu to start your search.
For each search criteria, at least one field is required for the search.
For date attribute, please use this syntax: **mm-dd-yyyy**

Selected Object:	gov.nih.nci.cabio.domain.Gene
id	
title	
symbol	IL*
locusLinkSummary	
OMIMID	
locusLinkId	
clusterId	
Search Object:	Please choose

http://localhost:8080/example/Criteria.jsp?klassName=gov.nih.nci.cabio.domain.Gene

Figure 11 Browser display with Gene as selected search object


Mozilla Firefox
File Edit View Go Bookmarks Tab Tools Help
http://localhost:8080/example/Result.jsp
Go
National Cancer Institute
U.S. National Institutes of Health | www.cancer.gov
caCORE - HTTP INTERFACE
Done

Criteria : gov.nih.nci.cabio.domain.Gene[@symbol=IL*]
Result class name: gov.nih.nci.cabio.domain.Gene

Results: 9 records found (Records 1 - 9 per page.)

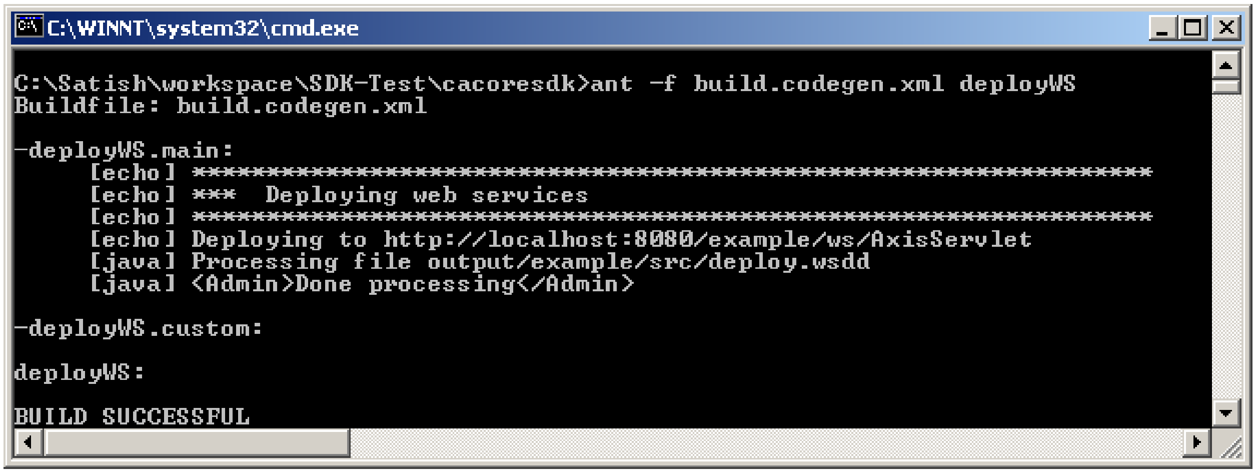
id	title	symbol	locusLinkSummary	OMIMID	locusLinkId
11	Interleukin 1 receptor-like 1	IL1RL1	-	601203	9173
12	Interleukin 2 receptor, gamma (severe combined imm	IL2RG	-	308380	3561
62	Interleukin 8	IL8	-	146930	3576
68	Interleukin 12A (natural killer cell stimulatory f	IL12A	-	161560	3592
69	Interleukin 12B (natural killer cell stimulatory f	IL12B	-	161561	3593
71	Interleukin 3 (colony-stimulating factor, multiple	IL3	-	147740	3562
94	Interleukin 13	IL13	-	147683	3596
95	Interleukin 8 receptor, beta	IL8RB	-	146928	3579
113	Interleukin 9	IL9	-	146931	3578

Figure 12 Browser displaying search results for Gene

Deploying WebServices

Perform the following steps to deploy webservices.

Step	Action
1	In a Command Prompt window, enter ***cd {home_directory}*** to go to your home directory. *Example:* ***cd C:\cacoresdk***
2	To deploy webservices, enter ***ant deployWS*** (Figure 13).
3	To test the Webservices interface, refer to “Running the Demonstration Programs” on page 27.



```
C:\Satish\workspace\SDK-Test\cacoresdk>ant -f build.codegen.xml deployWS
Buildfile: build.codegen.xml

-deployWS.main:
     [echo] ****************************************************************
     [echo] ***  Deploying web services
     [echo] ****************************************************************
     [echo] Deploying to http://localhost:8080/example/ws/AxisServlet
     [java] Processing file output/example/src/deploy.wsdd
     [java] <Admin>Done processing</Admin>

-deployWS.custom:

deployWS:

BUILD SUCCESSFUL
```

Figure 13 Output of deploy Webservices

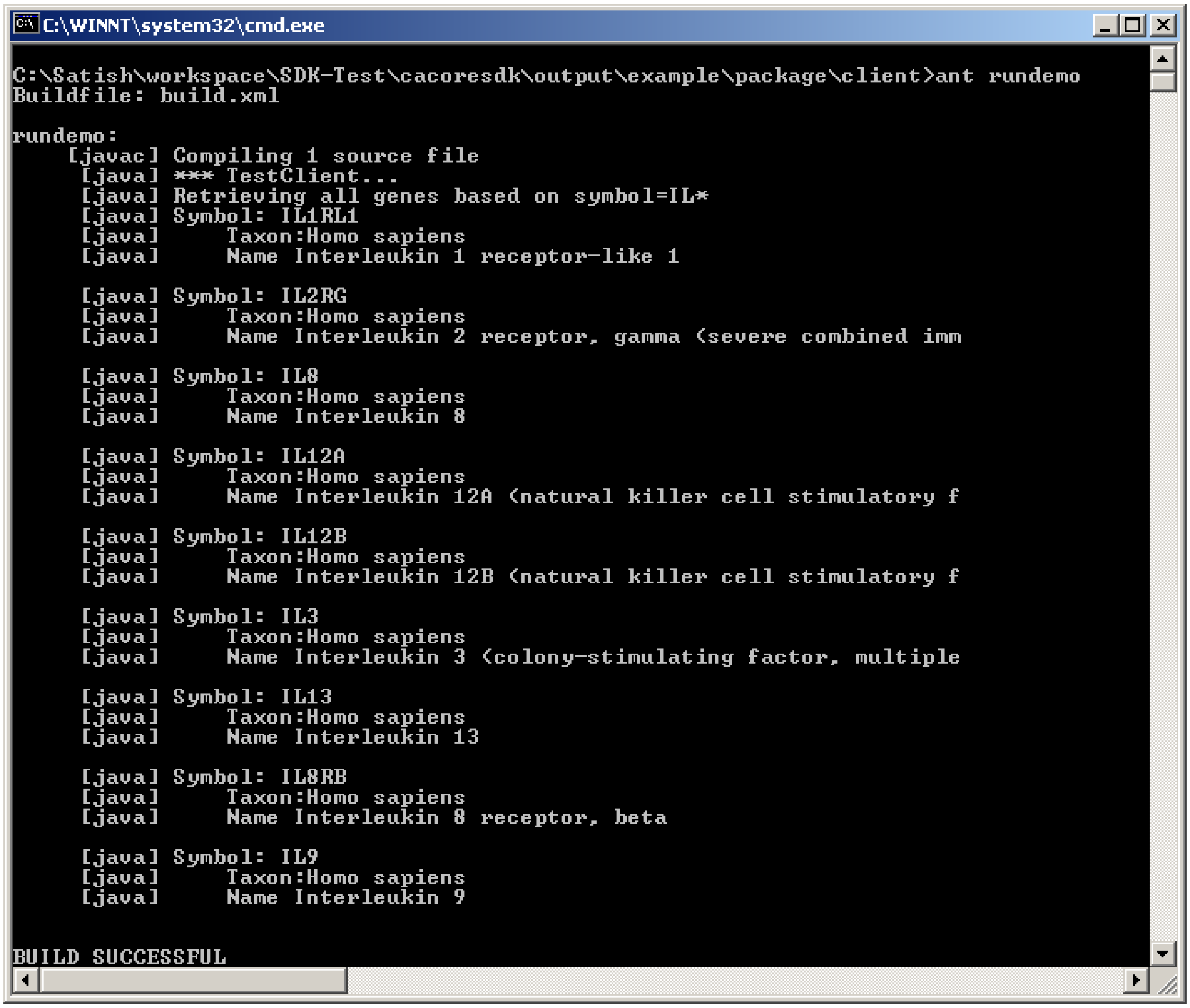
Running the Demonstration Programs

Three test classes are provided with the example system that demonstrate the use of the Java interface, web services interface, and the XML conversion utility. Perform the following steps to run the demonstration programs.

Step	Action
1	In a Command Prompt window, enter ***cd {home_directory}\output\{project_name}\package\client*** to go to your client’s home directory. *Example:* ***cd C:\cacoresdk\output\example\package\client***
2	To test the Java interface, enter ***ant rundemo*** at the command prompt
3	To test the web services interface, enter ***ant runWSdemo*** at the command prompt

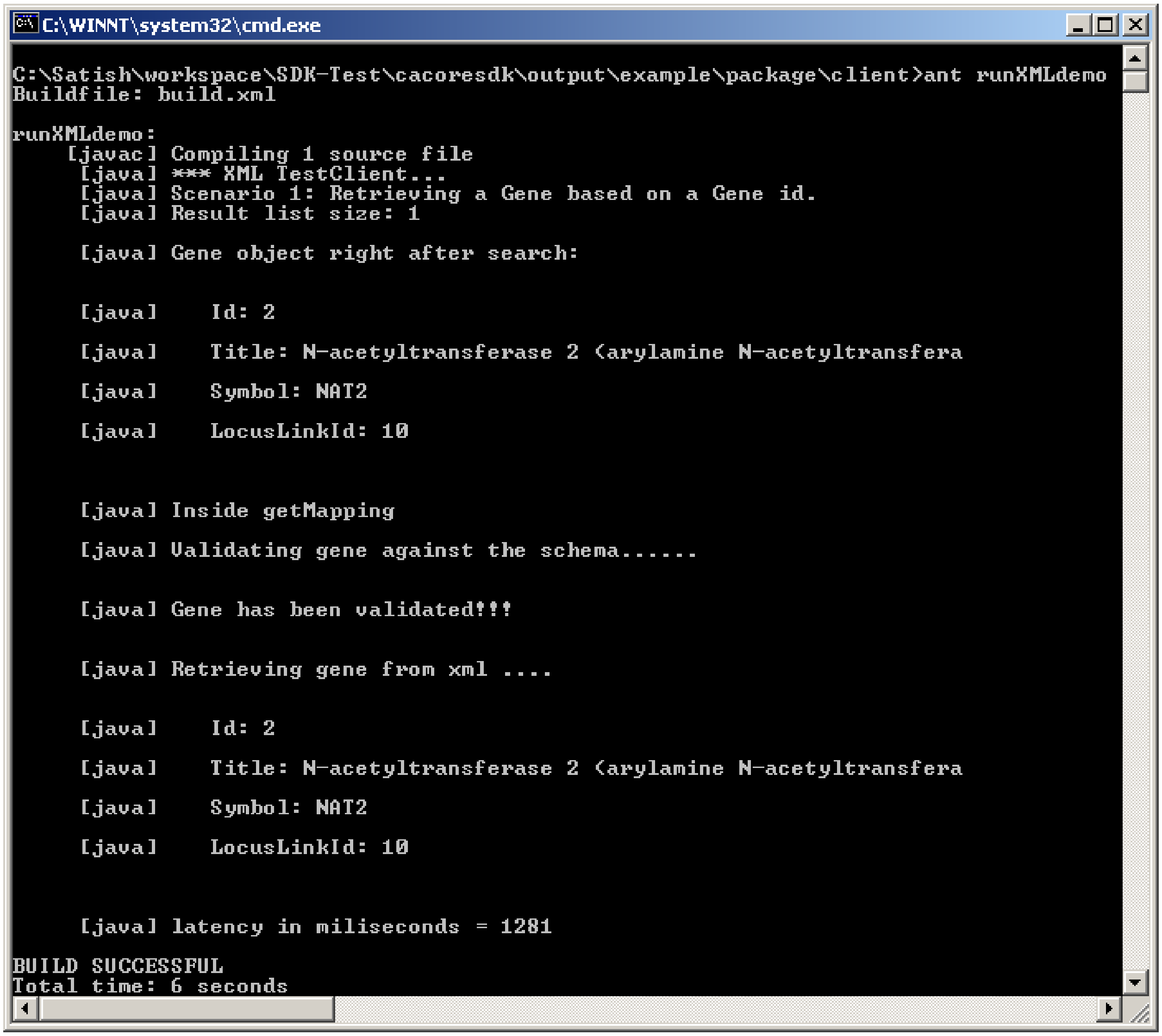
4	To test the XML conversion utility, enter ***ant runXMLdemo*** at the command prompt

The following figures display example outputs of running the demonstrations.



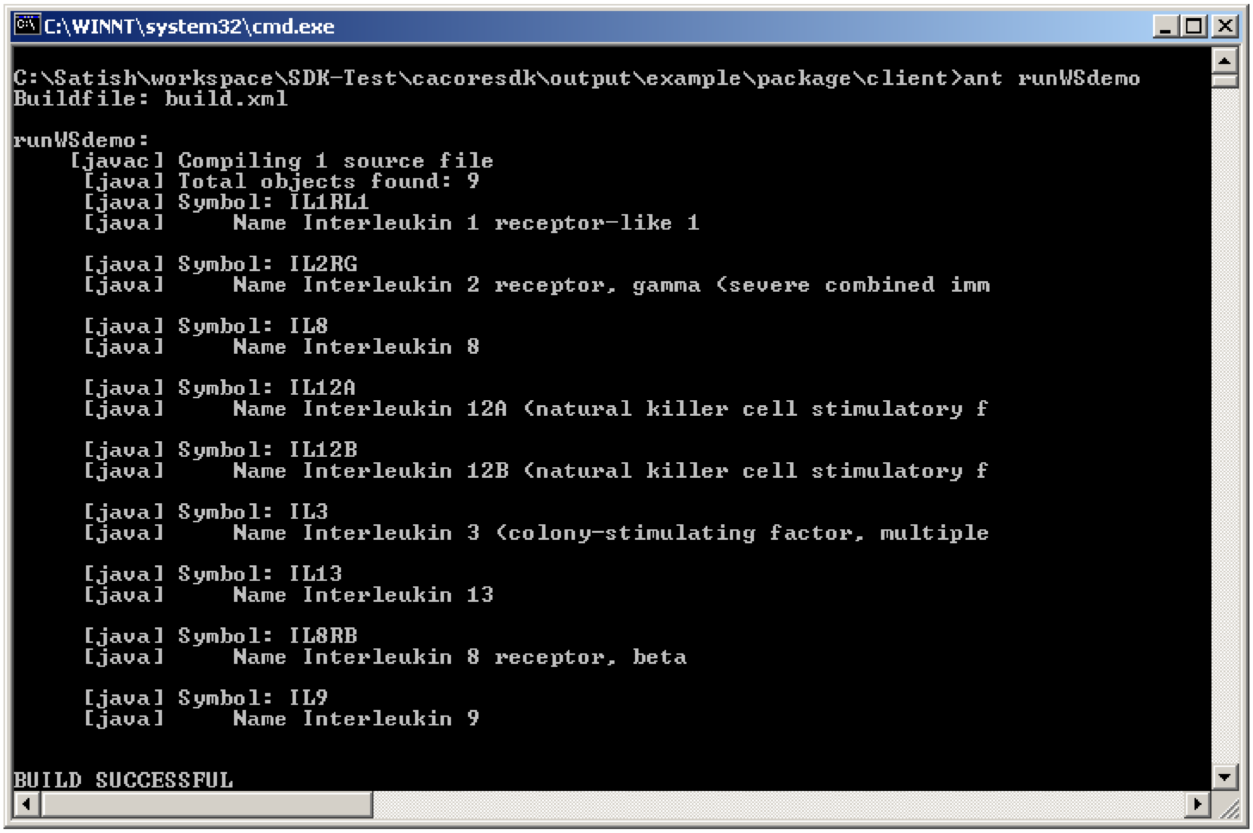
```
C:\Satish\workspace\SDK-Test\cacoresdk\output\example\package\client>ant rundemo
Buildfile: build.xml

rundemo:
    [javac] Compiling 1 source file
     [java] *** TestClient...
     [java] Retrieving all genes based on symbol=IL*
     [java] Symbol: IL1RL1
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 1 receptor-like 1

     [java] Symbol: IL2RG
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 2 receptor, gamma (severe combined imm

     [java] Symbol: IL8
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 8

     [java] Symbol: IL12A
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 12A (natural killer cell stimulatory f

     [java] Symbol: IL12B
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 12B (natural killer cell stimulatory f

     [java] Symbol: IL3
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 3 (colony-stimulating factor, multiple

     [java] Symbol: IL13
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 13

     [java] Symbol: IL8RB
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 8 receptor, beta

     [java] Symbol: IL9
     [java]     Taxon:Homo sapiens
     [java]     Name Interleukin 9


BUILD SUCCESSFUL
```

Figure 14 Example output of rundemo



```
C:\Satish\workspace\SDK-Test\cacoresdk\output\example\package\client>ant runXMLdemo
Buildfile: build.xml

runXMLdemo:
    [javac] Compiling 1 source file
     [java] *** XML TestClient...
     [java] Scenario 1: Retrieving a Gene based on a Gene id.
     [java] Result list size: 1

     [java] Gene object right after search:


     [java]    Id: 2

     [java]    Title: N-acetyltransferase 2 (arylamine N-acetyltransfera

     [java]    Symbol: NAT2

     [java]    LocusLinkId: 10



     [java] Inside getMapping

     [java] Validating gene against the schema......


     [java] Gene has been validated!!!


     [java] Retrieving gene from xml ....


     [java]    Id: 2

     [java]    Title: N-acetyltransferase 2 (arylamine N-acetyltransfera

     [java]    Symbol: NAT2

     [java]    LocusLinkId: 10



     [java] latency in miliseconds = 1281

BUILD SUCCESSFUL
Total time: 6 seconds
```

Figure 15 Example output of runXMLdemo



```
C:\Satish\workspace\SDK-Test\cacoresdk\output\example\package\client>ant runWSdemo
Buildfile: build.xml

runWSdemo:
    [javac] Compiling 1 source file
     [java] Total objects found: 9
     [java] Symbol: IL1RL1
     [java]     Name Interleukin 1 receptor-like 1

     [java] Symbol: IL2RG
     [java]     Name Interleukin 2 receptor, gamma (severe combined imm

     [java] Symbol: IL8
     [java]     Name Interleukin 8

     [java] Symbol: IL12A
     [java]     Name Interleukin 12A (natural killer cell stimulatory f

     [java] Symbol: IL12B
     [java]     Name Interleukin 12B (natural killer cell stimulatory f

     [java] Symbol: IL3
     [java]     Name Interleukin 3 (colony-stimulating factor, multiple

     [java] Symbol: IL13
     [java]     Name Interleukin 13

     [java] Symbol: IL8RB
     [java]     Name Interleukin 8 receptor, beta

     [java] Symbol: IL9
     [java]     Name Interleukin 9


BUILD SUCCESSFUL
```

Figure 16 Example output of runWSdemo

NOTES:	• The `rundemo and runWSdemo` tasks execute a default query that searches for Gene objects whose symbol begins with the letters ‘IL’. • The runXMLdemo task performs a query for a certain Gene object, saves an XML serialization of that object to disk, then deserializes the same object and displays its attributes. • These successful tests indicate that the server and client files have been properly generated and that the system can communicate with the database and servlet environments and “data in objects”.

Congratulations! If your installation passed these tests, you have successfully installed the infrastructure needed for a caCORE-like system and built a demonstration application. Refer to the *caCORE SDK 3.2.1 Programmer’s Guide* for the detailed procedures necessary to generate your own caCORE-like system.

Appendix I caCORE SDK Software and Technology Requirements

Table 9 lists required software for utilizing the caCORE SDK that is included with the SDK. The software name, version, description, URL, and whether it is included (Incl.) in the distribution are indicated. The included (**Incl.**) column indicates (with a **Yes**) if the software is packaged with the SDK. **No** indicates that you must supply the software.

Software Name	***Version***	***Description***	***URL***	***Incl***
Java 2 Platform Standard Edition (J2SE) 5.0 Development Kit (JDK 5.0)	jdk1.4.2	The J2SE Software Development Kit (SDK) supports creating J2SE applications	http://java.sun.com/j2se/1.4/download.jsp	No
UML 1.3 Modeling Tool that produces XMI 1.1 output format	EA 4.50.744 or higher	We recommend using Enterprise Architect (EA)	http://www.sparxsystems.com.au	No
Ant.jar	1.6.5	Apache Ant is a Java-based build tool	http://ant.apache.org/bindownload.cgi	Yes
activation.jar		The classes that make up the JavaBeans Activation Framework (JAF) standard extension are contained in the included Java Archive (JAR) file, "activation.jar"	http://java.sun.com/products/javabeans/glasgow/jaf.html	Yes
antlr-2.7.6.jar	2.7.6	Query parser used by Hibernate 3	http://www.antlr.org/download.html	Yes
ant-testunit.jar	1.6.5	For running JUnit tests via the Ant junit task.	Not available as a standalone binary download. You must download the ANT sources and then build it.	Yes
asm.jar	1.5.3	ASM is a Java bytecode manipulation framework. It can be used to dynamically generate stub classes or other proxy classes, directly in binary form, or to dynamically modify classes at load time, i.e., just before they are loaded into the Java Virtual Machine.	http://asm.objectweb.org/index.html	Yes
asm-attrs.jar	1.5.3	Part of the ASM bytecode manipulation framework.	http://www.hibernate.org	Yes
axis-ant.jar	1.4	Ant tasks for building axis.	http://ws.apache.org/axis/releases.html	Yes
axis.jar	1.4	Apache Axis is an implementation of the SOAP (Simple Object Access Protocol)	http://ws.apache.org/axis/releases.html	Yes
c3p0-0.9.0.jar	0.9.0	Connection pool used by Hibernate	http://sourceforge.net/projects/c3p0	Yes
castor-1.0.2.jar	1.0.2	XML processor for marshalling and unmarshalling Java objects	http://www.castor.org/download.html	Yes

Software Name	Version	Description	URL	Incl
cglib-full-2.1.3.jar	2.0.1	Dynamic Java byte code generator	http://sourceforge.net/project/showfiles.php?group_id=56933	Yes
classes12.jar		Oracle JDBC Drivers. Classes for use with JDK 1.2 and JDK 1.3. It contains the JDBC driver classes, except classes for NLS support in Oracle Object and Collection types.		Yes
clm.jar	3.2	Common Logging Module (CLM), which provides a separate service under caCORE for Audit and Logging Capabilities.		Yes
codegen.jar		Classes required for Java Emitter Template (JET) compilation.	http://www.eclipse.org/	Yes
commons-beanutils.jar	1.6	Apache Jakarta Commons utilities	http://apache.bestwebcover.com/java-repository/commons-collections/jars/	Yes
commons-collections-3.2.jar	2.1	Apache Jakarta Commons utilities	http://apache.bestwebcover.com/java-repository/commons-collections/jars/	Yes
commons-dbcp-1.2.1.jar	1.1	The Jakarta Commons DBCP Component provides database connection pooling.	http://archive.apache.org/dist/java-repository/commons-dbcp/jars/?C=S;O=A	Yes
commons-discovery-0.2.jar	0.2	Apache Jakarta Commons discovery utilities	http://jakarta.apache.org/commons/discovery/	Yes
commons-lang-2.1.jar	2.1	Provides helper utilities for the java.lang API.	http://linux.cs.lewisu.edu/apache/java-repository/commons-lang/jars/?C=N;O=D	Yes
commons-logging-1.1.jar	1.1	Provides a helper utilities logging.	http://public.planetmirror.com/pub/maven/commons-logging/jars/	Yes
commons-logging.jar		Apache Jakarta Commons logging utilities	http://jakarta.apache.org/commons/logging/	Yes
commons-pool-1.2.jar	1.1	The Jakarta Commons Pool Component provides a generic object pooling API.	http://apache.intissite.com/java-repository/commons-pool/jars/	Yes
csmapi.jar	3.2	Common Security Module		Yes

Software Name	*Version*	*Description*	*URL*	*Incl*
datafile.jar	1.3.2	Java data file read/write utility that provides a convenient set of interfaces for reading and writing data to and from files in widely accepted format such as comma separated values (CSV), fixed width, tab separated, as well as others	http://datafile.sourceforge.net/	Yes
db2java.jar		Contains classes to support connections to DB2 databases.	http://www-306.ibm.com/software/data/db2/udb/	Yes
dom4j-1.6.1.jar		Contains classes that allow you to read, write, navigate, create and modify XML documents.	http://public.planetmirror.com/pub/maven/dom4j/jars/	Yes
ehcache-1.2.2.jar	1.2.2	Ehcache is a pure Java, in-process cache.	http://smokeping.planetmirror.com/pub/maven/ehcache/jars/	Yes
freemarker.jar	2.3	FreeMarker is a "template engine"; a generic tool to generate text output (anything from HTML or RTF to auto generated source code) based on templates.	http://freemarker.sourceforge.net/freemarkerdownload.html	Yes
hibernate3.jar	3.1.3	Hibernate is used for the server-side object-relational mapping (ORM)	http://www.hibernate.org	Yes
jakarta-oro-2.0.8.jar	2.0.8	The Jakarta-ORO Java classes are a set of text-processing Java classes that provide Perl5 compatible regular expressions, AWK-like regular expressions, glob expressions, and utility classes for performing substitutions, splits, filtering filenames, etc.	http://jakarta.apache.org/site/binindex.cgi	Yes
jalopy-1.0b11.jar	1.0b11	Source code formatter.	http://public.planetmirror.com/pub/maven/jalopy/jars/	Yes
jalopy-ant-0.6.2.jar	0.62	Ant task for building jalopy.	http://public.planetmirror.com/pub/maven/jalopy/jars/	Yes
jaxen-core.jar		The jaxen project is a Java XPath Engine. jaxen is a universal object model walker, capable of evaluating XPath expressions across multiple models.	http://jaxen.org/releases.html	Yes

Software Name	Version	Description	URL	Incl
jaxen-jdom.jar		The jaxen project is a Java XPath Engine. jaxen is a universal object model walker, capable of evaluating XPath expressions across multiple models.	http://jaxen.org/releases.html	Yes
jaxrpc.jar		Java API for XML-based RPC		Yes
Jboss-client.jar	4.0.4		http://www.jboss.org	Yes
jdom.jar	1.0	Java-based solution for accessing, manipulating, and outputting XML data from Java code.	http://www.jdom.org/downloads/index.html	Yes
jdtcore.jar		Eclipse Tomcat Plugin		Yes
jetc-task.jar		An ANT task for translating JET templates outside of Eclipse		Yes
jmi.jar		JMI is a standards-based, platform independent, vendor-neutral specification for modeling, creating, storing, accessing, querying, and interchanging metadata using UML, XML, and Java.	http://mdr.netbeans.org/download/daily.html	Yes
jmiutils.jar			http://mdr.netbeans.org/download/daily.html	Yes
jta.jar		JTA specifies standard Java interfaces between a transaction manager and the parties involved in a distributed transaction system	http://java.sun.com/products/jta/	Yes
junit-3.8.1.jar	3.8.1	JUnit is a regression testing framework that is used by the developer who implements unit tests in Java	http://www.junit.org/index.htm	Yes
log4j-1.2.13.jar	1.2.13	Log4j is an open source tool developed for putting log statements into your application. With log4j you can enable logging at runtime without modifying the application binary.	http://logging.apache.org/log4j/docs/download.html	Yes
mail.jar	1.2	JavaMail API		Yes
mdrant.jar		Ant tasks for building MDR.	http://mdr.netbeans.org/download/daily.html	Yes

Software Name	***Version***	***Description***	***URL***	***Incl***
mdrapi.jar		MDR implements the OMG's MOF (Meta Object Facility) standard based metadata repository and integrates it into the NetBeans Tools Platform. It contains implementation of MOF repository including persistent storage mechanism for storing the metadata. The interface of the MOF repository is based on (and fully compliant with) JMI (Java Metadata Interface - JSR-40).	http://mdr.netbeans.org/download/daily.html	Yes
mof.jar			http://mdr.netbeans.org/download/daily.html	Yes
mysql-connector-java-3.1.13-bin.jar	3.1.13	My SQL database driver	http://www.mysql.com/products/connector/j/	Yes
nbmdr.jar		http://jaxen.codehaus.org/	http://mdr.netbeans.org/download/daily.html	Yes
ojdbc14.jar		ODBC driver of Oracle database		
openide-util.jar		Contains low level basic support classes that MDR depends on.	http://mdr.netbeans.org/download/daily.html	Yes
osgi.jar			http://www.osgi.org/osgi_technology/download_specs.asp?section=2	Yes
p6spy.jar		An open source framework for applications that intercept and optionally modify database statements.	http://www.p6spy.com	Yes
resources.jar				Yes
runtime.jar				Yes
saaj.jar	1.2			Yes
saxpath.jar	1.0-FCS	SAXPath is an event-based API for XPath parsers, that is, for parsers that parse XPath expressions.		Yes
serializer.jar		Contains the serializer classes of Xalan-Java2.		Yes
servlet.jar		HTTP Servlet classes		Yes
spring.jar	1.2.7	Spring framework	http://www.springframework.org	Yes
spring-richclient-full-0.1.0.jar	0.1	Spring framework remote client	http://www.springframework.org	Yes
uml-1.3.jar				Yes
uuid-key-generator.jar		JBoss UUID key generator		
wsdl4j-1.5.1.jar	1.5.1	Web Services Description Language support for Java		Yes

Software Name	Version	Description	URL	Incl
xalan.jar		An XSLT (Extensible Stylesheet Language Transformation) processor for transforming XML documents into HTML, text, or other XML document types.	http://xalan.apache.org/	Yes
xercesImpl.jar	2.7.1	Xerces Java Parser	http://xml.apache.org/xerces-j/	Yes
xml-apis.jar	2.0.2	XSLT processor for transforming XML documents into HTML, text, or other XML document types.	http://xml.apache.org/xalan-j/	Yes
Xmlrpc-2.0.jar	2.0	Apache XML-RPC is a Java implementation of XML-RPC, a popular protocol that uses XML over HTTP to implement remote procedure calls.	http://www.apache.org/	Yes

Table 9 Required software for the caCORE SDK 3.2

NOTE:

Drivers for MySQL database are included in the SDK. For other databases, you must download their drivers (for example, the Oracle 8i driver `classes12.zip` can be downloaded from http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html).

Appendix II caCORE SDK Components

The caCORE SDK includes the following components:

- Sample UML object/data model to use with the development kit
    - cabio.eap
- XML Metadata Interchange (XMI) Version of the sample model
    - cabioExampleModel.xmi
- Framework packages
    - gov.nih.nci.system
    - gov.nih.nci.common
    - org.hibernate
- Configuration files to enable you to customize your installation to meet your specific database, server, and other network needs
    - deploy.properties
    - download.properties
- Ant buildfile
- Code generator package
    - gov.nih.nci.codegen.core
    - gov.nih.nci.codegen.framework
    - Java JET templates for generating caCORE like APIs
- MySQL database
- Demo package with examples of how to leverage the code generation framework (for advanced users)

Contacting Technical Support

NCICB Application Support	http://ncicb.nci.nih.gov/NCICB/support Telephone: 301-451-4384 Toll free: 888-478-4423