1 First Steps

1.1 Introduction

Because you are reading this book, we assume you are interested in object-oriented application development in general and the Caché postrelational database from InterSystems in particular. And because you have chosen a practical manual with a CD-ROM, we further assume that your interest is more than theoretical and that you wish to use it to gain practical experience as fast as possible.

Installing Caché and developing the first object-oriented application

Therefore, let us postpone, for the moment, a detailed description of object-oriented principles and go straightaway into application development. In this chapter we install Caché and other software products from the accompanying CD-ROM, become acquainted with basic system administration, and develop a small object-oriented application with Caché.

In Chapter 2 we shall return to the principles of object-oriented development. If you wish to skip this introductory chapter, you can jump to Chapter 2. Even so, you should install Caché before proceeding to later chapters that cover in detail the specific aspects of application development with Caché. You will find it easier to follow the examples given in this book if you run them for yourself with the software installed from the accompanying CD-ROM.

But before you unpack the CD-ROM and install the software, we must first answer one basic question:

What is Caché—the postrelational database?

Postrelational database models

InterSystems Corporation—a database vendor headquartered in Cambridge, Massachusetts—introduced Caché at the end of 1997. InterSystems refers to Caché as a "postrelational" database, which is primarily a time reference. In this sense, all databases designed and developed after the relational model could be called postrelational. The term is often used in this manner. For example, Vossen [1994] uses "postrelational" to group all databases no longer based on normalized, flat tables, and mentions nested-relational, multidimensional, and object-oriented databases as examples.

But a definition based on time alone does not suffice. To understand the content of a postrelational database, you must appreciate the changed demands made on modern database systems, which, for example, Currier [1997] describes impressively. Current applications are normally developed to support complex information flow and comprehensive work processes in companies. A developer's principal task here is to correctly

determine the structures of the macro-cosmos and represent them as appropriate models in the micro-cosmos of information technology.

It is obvious that corporate applications are affected by the business processes they model. Object-oriented technologies are ideal for real-world business process modeling, and this is the main reason for their increasing popularity. (Chapter 2 provides a detailed description of the object-oriented technologies.)

It became apparent only later that the models in which information is stored also needed to be updated. Vaskevitch [1992] wrote in an article for BYTE magazine:

Postrelational databases reflect an image of the real world

> *Relational databases still make sense for many kinds of applications that lend themselves to expression as simple sets of simple tables. However, to build applications that reflect the real world, you will have to find a way to take another two steps forward this time to a postrelational world.*

In this sense, the postrelational model from Caché helps software developers represent the relationships of "real world" data in information technology without oversimplification. This book shows how you can quickly develop reliable, sophisticated applications using Caché.

1.2 System Requirements

To successfully install and run the software on the accompanying CD-ROM, your system should satisfy the following minimal requirements:

- Intel Pentium CPU
- Windows 95, 98, ME, NT 4.0 (SP4, 5 or 6), 2000 (SP2), or XP
- 128 MB main memory
- 100 MB free disk space
- Configured TCP/IP stack
- CD-ROM drive

1.3 Installation

The accompanying CD-ROM contains an automatic setup program that provides the following options:

- Read Book Online
- Install Acrobat Reader
- Install Caché 5
- Caché Documentation and Tutorials
- Book Examples

If the CD-ROM does not start automatically when you insert it, the autostart function for your CD-ROM drive has been deactivated. You can manually start the installation program by double-clicking the specified items in the following sequence: *My Computer*, the CD-ROM drive symbol, and the *autostart.exe* program.

1.3.1 Read Book Online

Read Book Online

The enclosed CD-ROM contains the complete text of this book, including the code examples. Due to copyright restrictions, you cannot print the book from the CD-ROM; however, you can read and search the text with the Adobe Acrobat Reader. If Adobe Acrobat Reader is not installed on your system, you can install it from the CD-ROM.

1.3.2 Installing Caché

Install Caché 5

To install Caché, select the *Install Caché 5* option from the menu. (Note: to install Caché under Windows NT, 2000, or XP, you must be logged on as administrator.)

Install to the default path and select *8-bit* as the installation type when prompted. Be sure that all Caché components needed for the examples in this book are installed.

The installation is complete when the following display appears on the screen. Under some circumstances, the system will need to restart; the installation routine will tell you if this is the case.

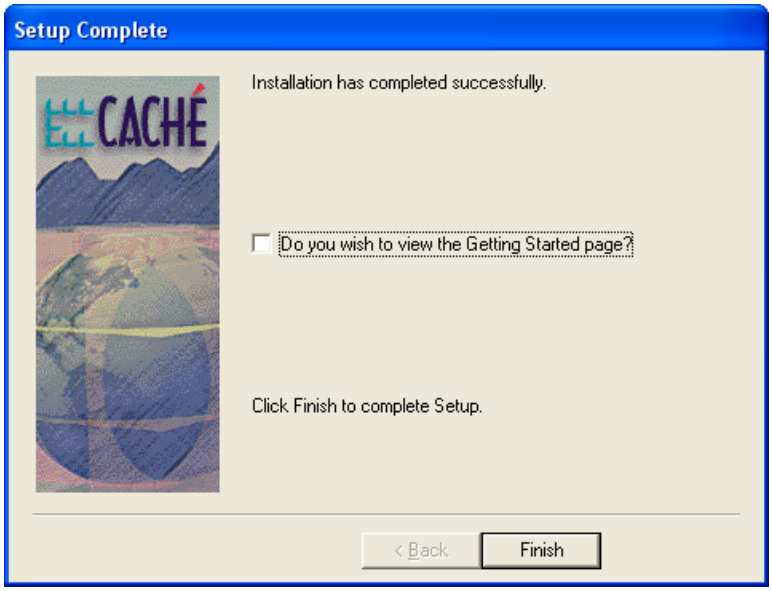


Figure 1.1 Successful installation

1.3.3 The Book Examples

The code examples used in the book can be imported as files from the CD-ROM. Use the setup program's menu option *Book Examples* to open Windows Explorer, which then shows subdirectories with the example files.

1.4 Using Caché

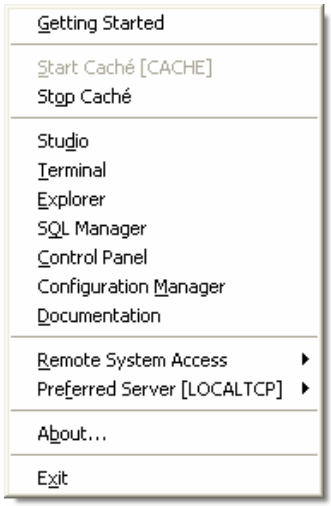
1.4.1 The Caché Cube

After a successful installation, Caché is ready to run. If no restart was necessary, Caché will already be running. To see whether Caché is running, check the system tray—this is the right-hand area of the taskbar at the lower edge of the screen. This displays the Caché Cube, an icon in the form of a cube. If the icon is colored, Caché is already running. If the cube is outlined in gray, Caché needs to be started, which you can do with the *Start Caché* option.

Figure 1.2 The Caché Cube

Click the Caché Cube icon to open a menu in which you can administer Caché. The second and third menu items are used to start and stop Caché.

Figure 1.3 The Caché Cube menu



1.4.2 The Caché Utilities

Management tools for Caché

You can invoke management tools that control important Caché functions directly from the Caché Cube menu.

Table 1.1 The Caché utilities

Menu option	Description
Getting Started	Shows the "Getting Started" document with shortcuts to important Caché online documentation and tutorials.
Start Caché	Starts Caché.
Stop Caché	Stops Caché and optionally initiates a restart.
Studio	Caché Studio is a graphical editor you use to create classes, CSP pages and routines with Caché ObjectScript or Caché Basic.
Terminal	Caché Terminal provides a terminal emulation screen that you can use to logon in command mode to local or remote Caché systems.
Explorer	Caché Explorer manages databases and namespaces together with their associated classes, globals, and routines.

SQL Manager	Caché SQL Manager helps you manage all aspects of relational access to Caché.
Control Panel	Caché Control Panel administers the running Caché system and its processes.
Configuration Manager	The Configuration Manager defines the system and network configurations for Caché.
Documentation	Accesses the Caché online documentation and tutorials.
Remote System Access	When you invoke a utility here, you can specify a server other than the preferred server.
Preferred Server	Sets up the Caché servers and selects the preferred server. For a single-user installation, the local server (LOCALTCP) has been set up automatically during installation.
About	Displays the copyright and version information for Caché.
Exit	If you close the Caché Cube with this option, you can use the Start menu later to reactivate it at any time.

1.5 Step-by-Step Through the First Example

Small object-oriented application with Caché

Once you have installed Caché and are familiar with its basic operations, it is time for an introduction to application development. We will use Caché to design and implement a small object-oriented application. This easy task can be accomplished in just 10 minutes.

1.5.1 The Task

We start with a simple person class as an example. A person should have properties for family name, first name, and date of birth (DOB). We will let Caché create the associated methods that store and retrieve persons in the database. We also want to provide a query for selecting all stored persons. Schematically, our model has the following form:

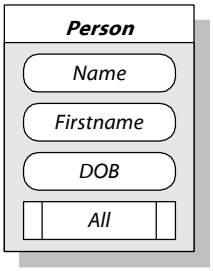


Figure 1.4 Person object class

1.5.2 Caché Studio

We now start Caché Studio to define this object class. Click the Caché Cube in the taskbar and select *Studio*. A window appears with the only existing connection, `Local`, and the namespace `User` already selected. Click *Connect* to accept these defaults.

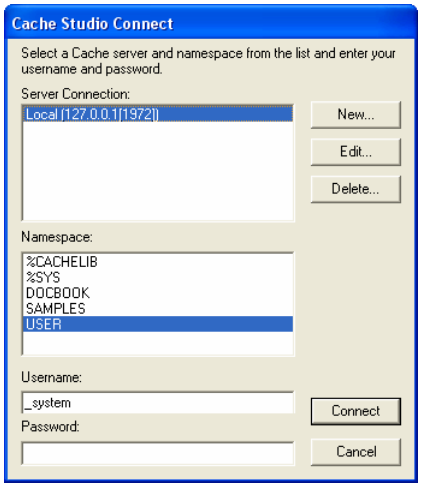


Figure 1.5 Selecting a connection

The Caché Studio main window that now appears shows an empty project with the name *Project1*.

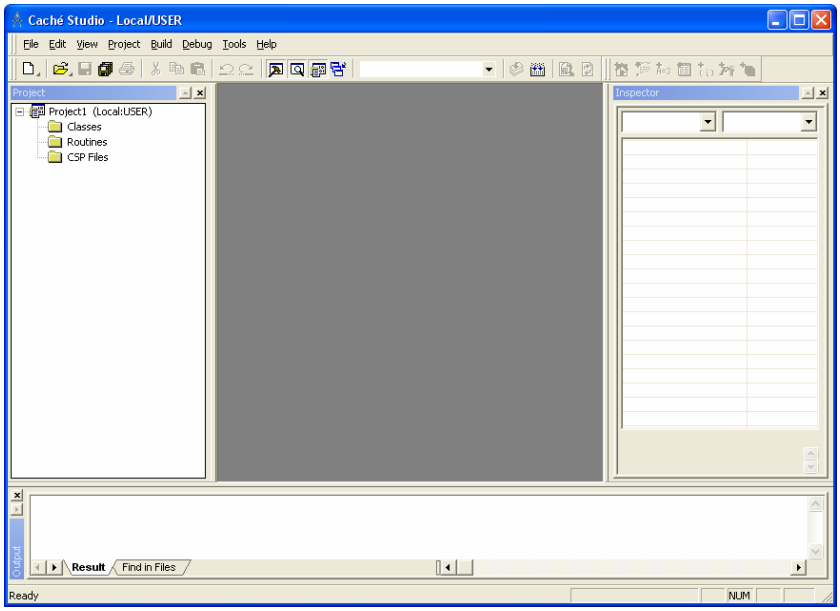


Figure 1.6 Caché Studio

1.5.2.1 Creating a Class Definition

Caché provides a wizard for creating new classes. Right-click *Classes* and select *Create New Class* in the context menu that opens. The *New Class Wizard* appears. Now specify the name and description of the new class.

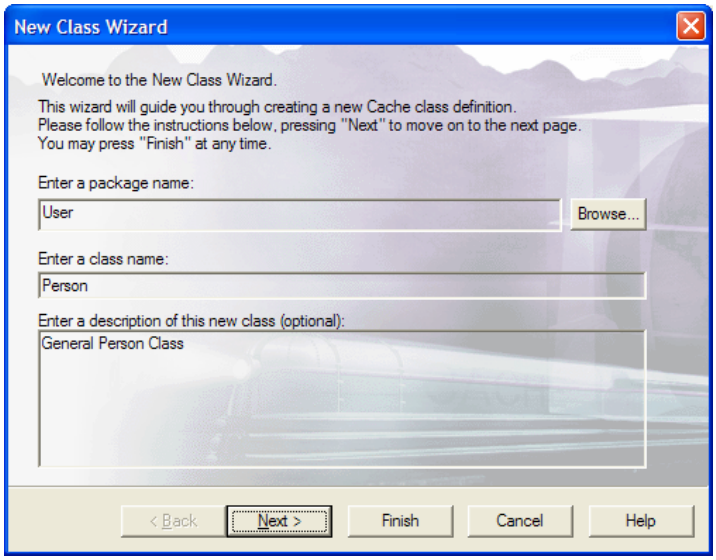


Figure 1.7 Creating a new class in the *New Class Wizard*

Keep the default package name, `User`. Specify `Person` as the name of the class, and enter a short description. Then click *Next* to proceed to the next step in which you specify the class type.

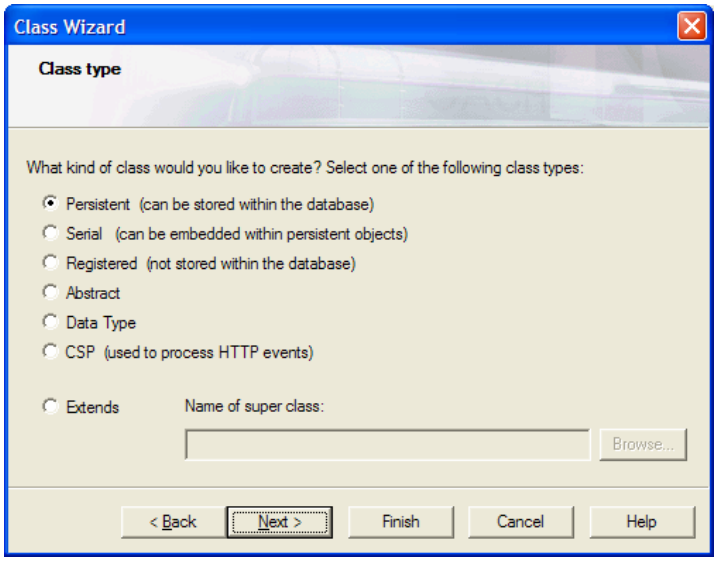


Figure 1.8 *Class type*—Selecting the class type

We keep the default class type, *Persistent*. Caché will now automatically store and fetch objects for this class.

Click *Complete* to skip the remaining steps of the wizard. The *Class Wizard* creates the new class, which now appears in the Caché Studio project hierarchy.

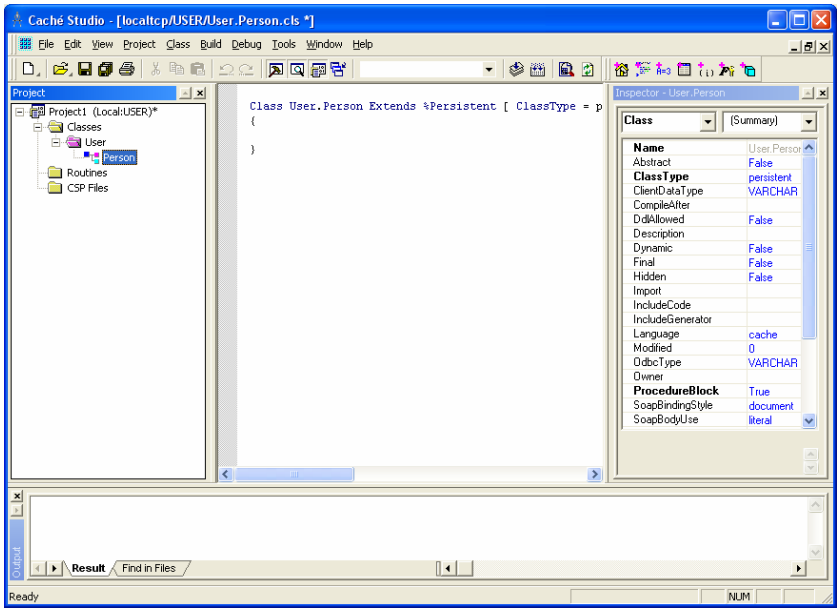


Figure 1.9 The newly-created class

1.5.2.2 Creating Properties

To define the properties of the class, click the *New Property* button.

Figure 1.10 The *New Property* button

The *New Property Wizard* opens.

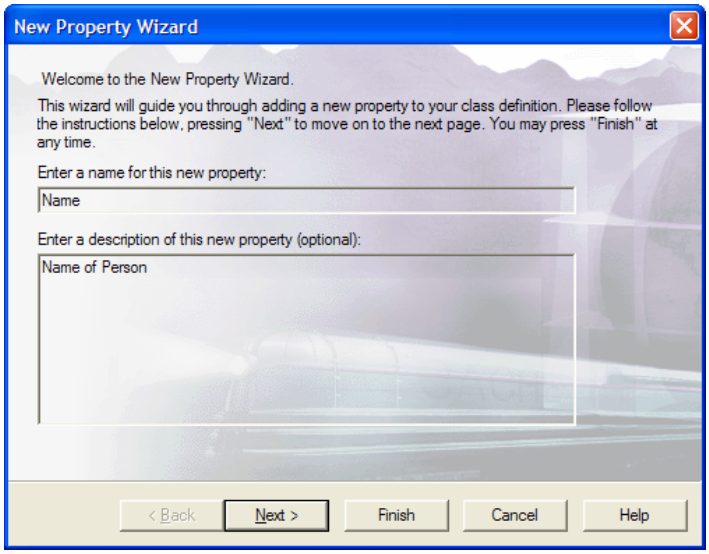


Figure 1.11 Creating a new property with the *New Property Wizard*

First we create a property for the person's name. Enter `Name` in the *Property name* field and specify a short description in the description field.

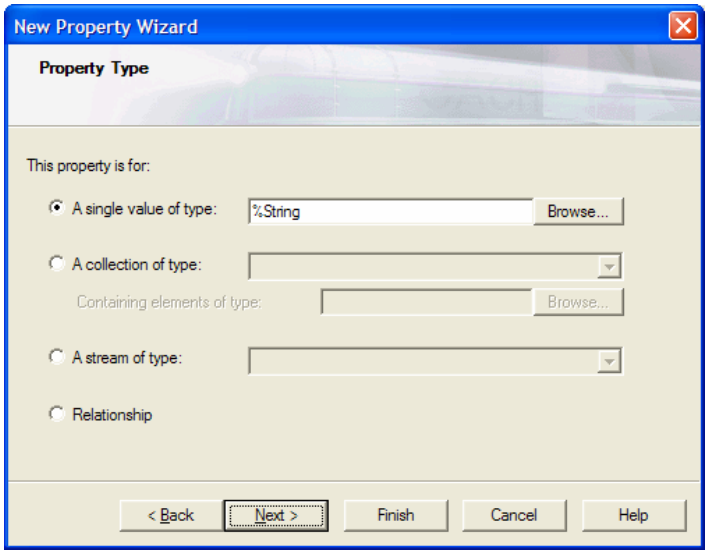


Figure 1.12 *Property Type*—Selection of data type

Retain the default type, `%String`, and click *Next*.

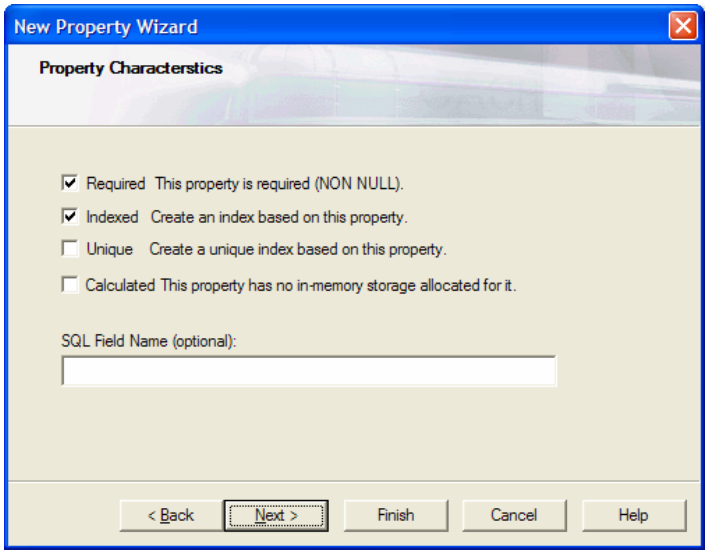


Figure 1.13 *Property Characteristics*—Characteristics of the Property

Check the *Required* and *Indexed* characteristics to specify that a name is mandatory and the name search is to be optimized, respectively. No other characteristics are necessary, so click *Finish*. The new property is now defined.

In the same way, we define the `Firstname` (without the *Required* and *Indexed* characteristics) and `DOB` (with the data type `%Date`) properties. Once we have defined these properties, the class definition has the following appearance:

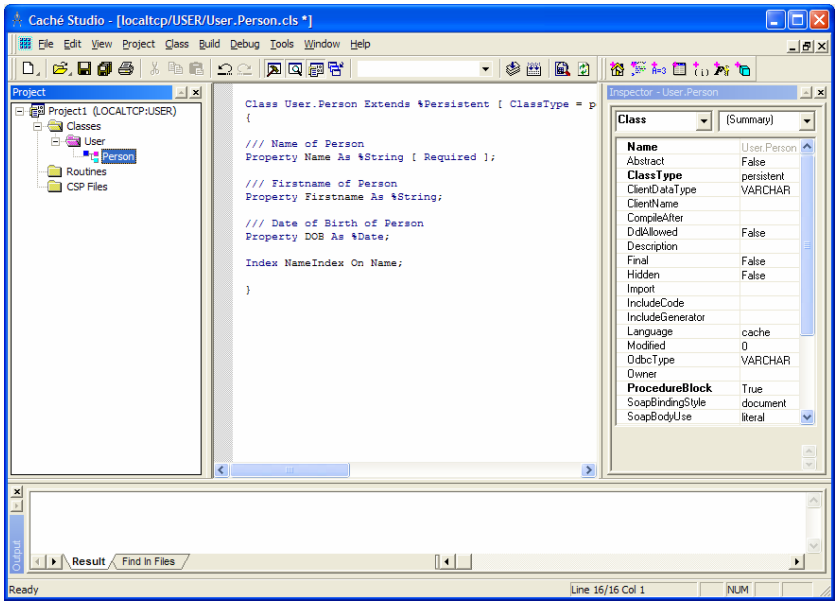


Figure 1.14 The class definition in Caché Studio

1.5.3 Creating the Query

We now define a new query that we can use to find all persons stored in the database. To do this, click the *New Query* button.

Figure 1.15 The *New Query* button

The *New Query Wizard* opens.

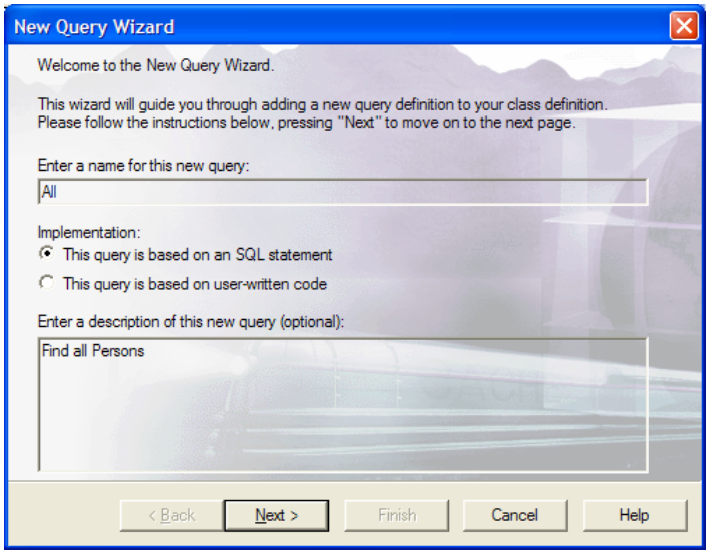


Figure 1.16 Creating a new query with the *New Query Wizard*

We enter `All` as the name of the query, and a short description in the field below. We keep the default, SQL statement, for the Implementation

selection. Clicking *Next* displays the definition for the request parameter, which we do not need, so click *Next* to skip it. In the next step we specify the fields that the query is to supply.

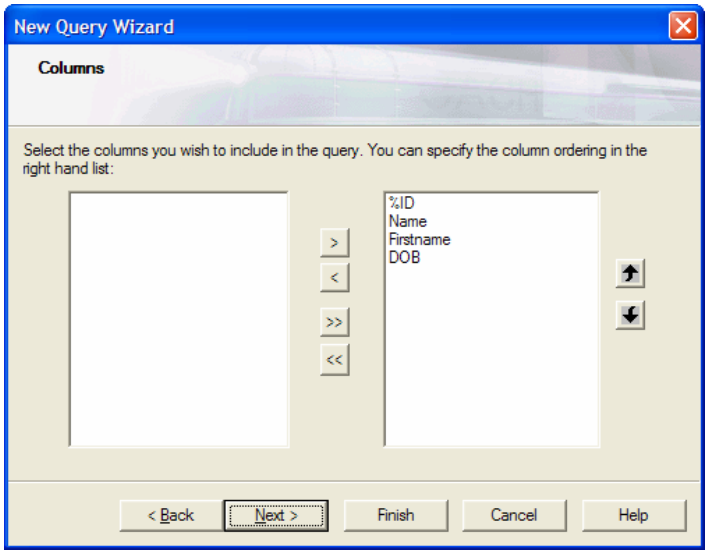


Figure 1.17 *Columns*—Selecting the query fields

On the left-hand side you see a list with all the properties of the object class and an additional ID field. Select each of the `%ID`, `Name`, `Firstname`, and `DOB` fields and click the > arrow to move each field into the right-hand list. When you are finished, click *Next*. The next step sets a query condition. Because we do not need to set this, click *Next* to proceed to make the selection for the sorting sequence.

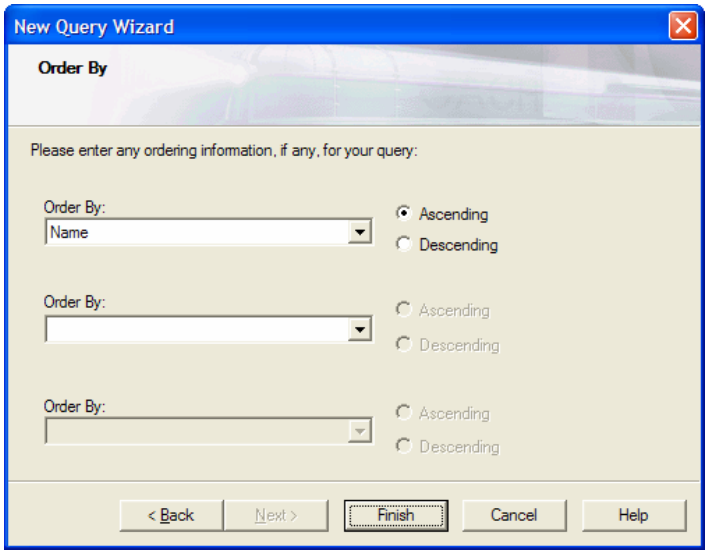
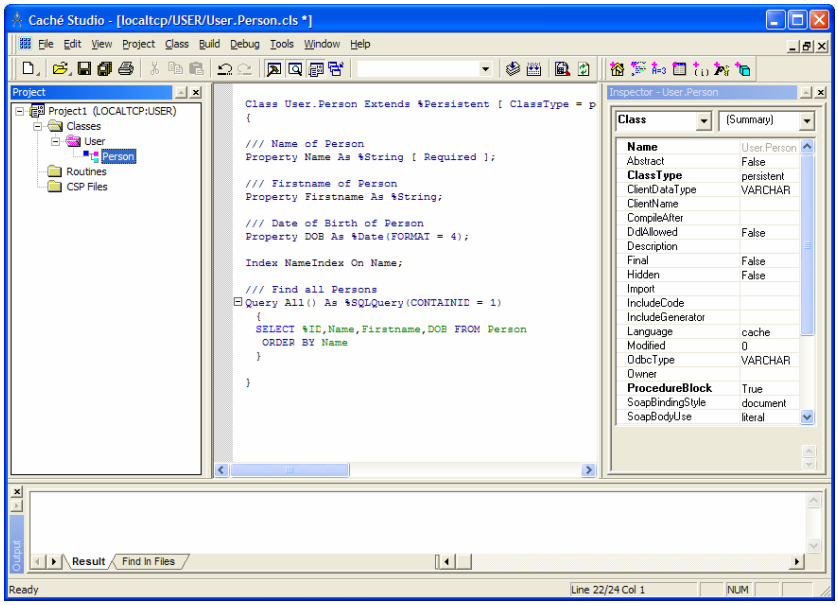


Figure 1.18 *Order By*—Defining the query's sorting sequence

Because we wish to sort the selection of persons by name, we choose the *Name* field from the *Order By* list. When we click *Finish,* Caché Studio shows the Person class with its properties and the new query.

Figure 1.19 Person class with its properties and the new All query



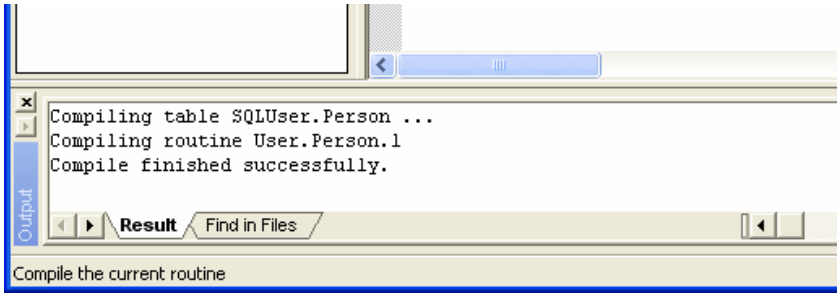
1.5.3.1 Compiling

A new class must be compiled before it can be used in Caché. To do this, click the *Compile* button.

Figure 1.20 The *Compile* button

After the compilation has completed, you see a message in the *Result* window that looks similar to the one shown in Figure 1.21.

Figure 1.21 Successful compilation of a class



The class definition is now complete, and the new User.Person class is ready for use.

1.5.4 The Caché Web Form Wizard

The *Caché Web Form Wizard* is a particularly easy way to create a Web front end for a Caché class definition with Caché Server Pages (CSP). We will use it to populate our person class with data.

First, we must create a CSP form. Right-click on *CSP Files* in the Caché Studio project hierarchy. Select *Create New CSP File* in the context menu that opens. Caché produces a new CSP form and displays it.

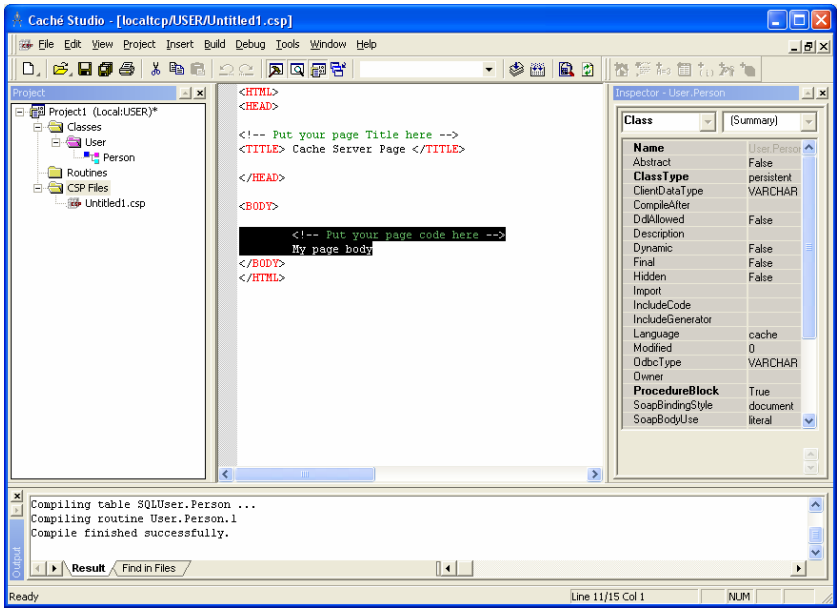


Figure 1.22 The newly-created CSP form

Mark the prepared default code between the <BODY> and </BODY> tags, then open the *Insert* menu and click the *Form Wizard*. A new window with the *Caché Web Form Wizard* opens.

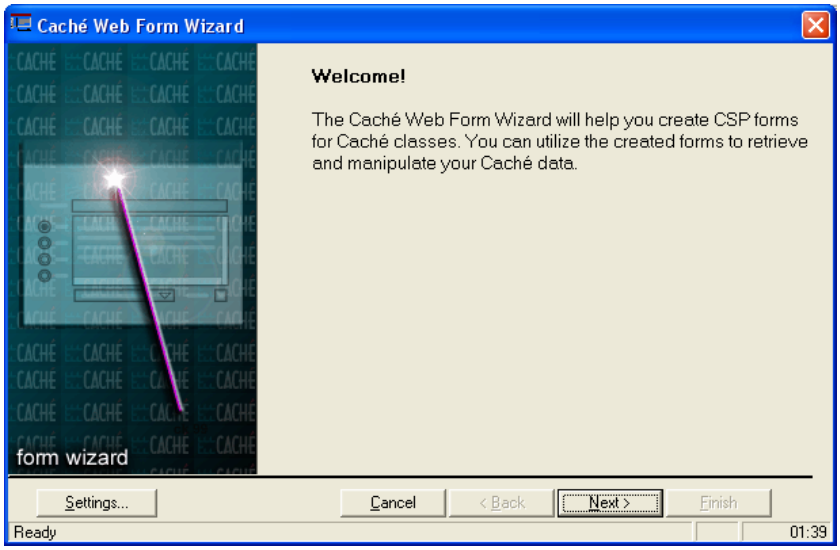


Figure 1.23 The Caché Web Form Wizard

Click *Next* to display a list of the available classes.

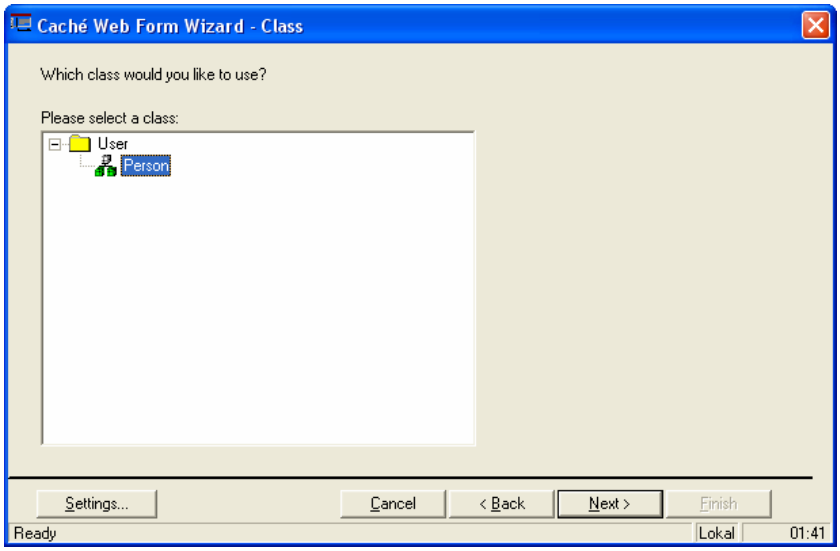


Figure 1.24 Selecting a class

The list contains a single class, `User.Person`. Select it and click *Next*. A list of the class's properties appears.

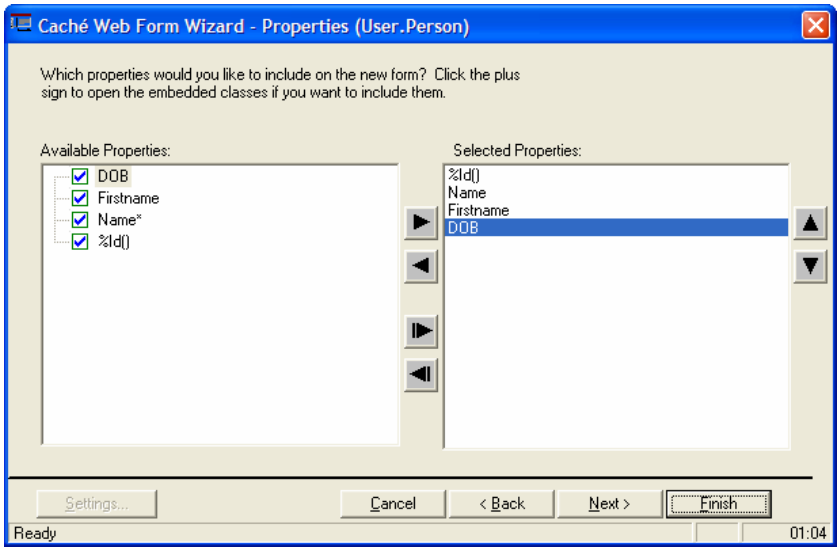


Figure 1.25 Selecting properties

Select each of the `%Id()`, `Name`, `Firstname`, and `DOB` properties to be included in the generated form, then click *Finish*. The newly generated code now appears between the <BODY> and </BODY> tags in the code window. To save the changes, go to the *File* menu and click *Save As*. Name the CSP form `person.csp`, and save the file in the `csp/user` directory.

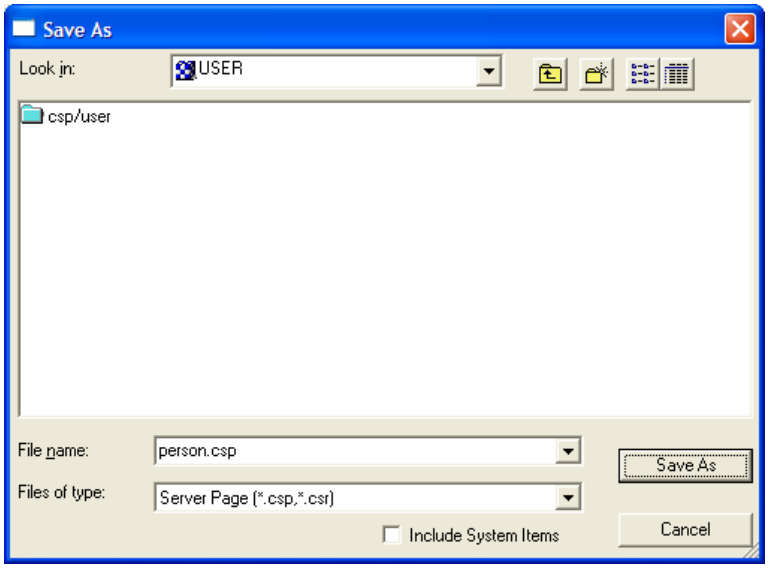


Figure 1.26 Saving a CSP form

To view the new Caché Server Page, click *Web Page* from the *View* menu. The Web browser opens, and the automatically generated form for our person class appears.

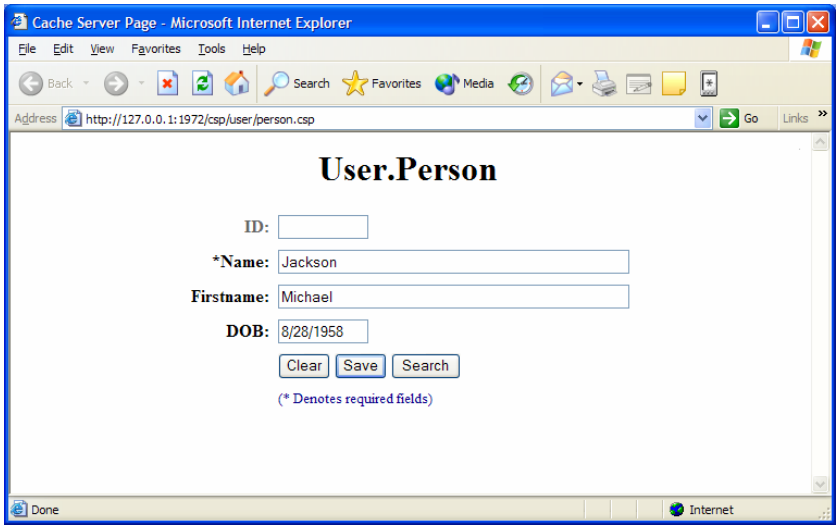


Figure 1.27 The automatically generated form in the Web browser

After you have used the wizard to create several persons, click the *Search* button to select a person to edit or delete.

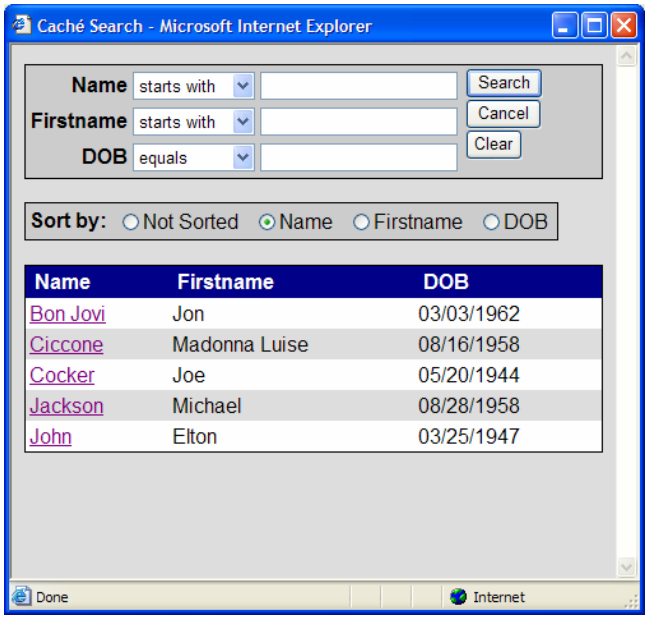


Figure 1.28 Using the generated search function

The form created by the *Caché Web Form Wizard* provides a general prototype that you can use for your own development projects. You can now edit and refine the code in Caché Studio according to your needs.

1.6 Outlook

Your first object-oriented Web application created with Caché

If you have worked through the example presented in this chapter, you have created your first object-oriented Web application with Caché. In the remainder of this book, we will discuss—in ever-deepening detail—different aspects of application development using Caché. The introduction presented in this chapter will help you develop your own examples in Caché and so provide a practical basis for the theoretical discussion in subsequent chapters. Note that you have already installed the software tools you will need.

However, before we progress deeper into development with Caché, the next chapter retraces a step to consider the general principles of object technology.

http://www.springer.com/978-3-540-00960-3

Object-Oriented Application Development Using the Caché Postrelational Database

Kirsten, W.; Ihringer, M.; Kühn, M.; Röhrig, B.

2003, XIV, 390 p., Hardcover
ISBN: 978-3-540-00960-3