

1C:ENTERPRISE 8.3

Hello, 1C

Rapid application development tutorial

Contents

Introduction
If you have everything you need?5
Infobase 6
Start "programming"12
Subsystems14
Catalogs16
Register41
Documents45
Reports
Improving the interface
Improving subsystems76
Adjusting the subsystem content80
Start page85
Command interface of Main section91
Managed forms94
Standard and ordinary attributes97
Object presentations101
Quick selection of values102
Adjusting reports104
Report variants109
Functional options
Cross-platform design
Linux
Web client
Mobile platform131
Where and how to study 1C:Enterprise149
1C:Enterprise 8 (training version)149
1C:AccountingSuite demo150
Business automation
1C:Enterprise 8
1C partners
Useful online resources
About 1C Company

Hello, 1C

Hello, 1C rapid application development tutorial demonstrates the most basic features of the cutting-edge 1C:Enterprise 8 platform. You will understand the application development process and see that it is easy to learn and develop applications powered by 1C:Enterprise 8.

Having downloaded 1C:Enterprise 8.3 training version for free, anyone can repeat the application development process on his or her computer.

The third edition includes the description of new features implemented in 1C:Enterprise 8.3, which you can find in the Cross-platform design chapter on page 125.

What do you think of this book? You are welcome to send us feedback as well as ask questions and get support on the Studying 1C:Enterprise platform forum.

Hello, 1C rapid application development tutorial, version 3.

Copyright (C) 2014 1C Company.

All rights reserved. No part of this document or the related files may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without the prior written permission of the publisher except for the use of brief quotations in a book review.

Limit of Liability and Disclaimer of Warranty: The publisher has used its best efforts in preparing this book, and the information provided herein is provided "as is." 1C Company makes no representation or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.

Trademarks: This book identifies product names and services known to be trademarks, registered trademarks, or service marks of their respective holders. They are used throughout this book in an editorial fashion only. In addition, terms suspected of being trademarks, registered trademarks, or service marks have been appropriately capitalized, although 1C Company cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark, registered trademark, or service mark. 1C Company is not associated with any other product mentioned in this book, except for 1C:Enterprise and applied solutions, developed by 1C Company or in association with 1C Company. 1C Company is not associated with any other vendor, mentioned in this book.

Russian text: V. Rybalka

Translation: T. Bugaevsky Correction: A. Novikov, S. Polikarpov

Introduction

The main business of 1C Company (see page 154) is development of business management and accounting software, as well as educational applications development and publishing, and software distribution. Over 1,000,000 companies and over 4,000,000 employees are using business software powered by 1C:Enterprise (see page 151). Market requirements for developers and qualified users capable operating 1C:Enterprise grows up constantly.

The main goal of this tutorial is to demonstrate the basic features of the 1C:Enterprise 8 platform and its software engineering technology. There will be no secrets except for one: within a few minutes, using almost only a mouse, you can create a fully functional application to keep personal finance records that is compatible with different DBMS (database management systems), Windows, Linux, web browsers, and even iOS and Android mobile devises.

Perhaps you are already familiar with one of high-level programming language, for example Java, C++, or Delphi. Many books and programming training courses start with a simple task such as creating a program that would display a simple text on a screen, for example, "Hello, world!"

In fact, what you are going to do in this tutorial is "Hello, world!" powered by 1C:Enterprise. Indeed, the functionality of the demo application is going to be much broader than the simple output of a phrase on a screen. However, considering all capabilities of the 1C:Enterprise 8 platform, what you are going to develop is precisely "Hello, world!" in the world of 1C:Enterprise.

If you have everything you need?

Before you begin, make sure that you have everything you need to get started. That is make sure that the 1C:Enterprise platform is installed.

Check that when you click **Start**, and then point to **All Programs**, there is **1C Enterprise 8 (training version)** inside.

If this application icon is not in the menu, the 1C:Enterprise platform must be installed.

If you do not have the installer, you can download the 1C:Enterprise 8 (training version) for free on 1C:Developer Network. After extracting from the archive, run **setup.exe**.

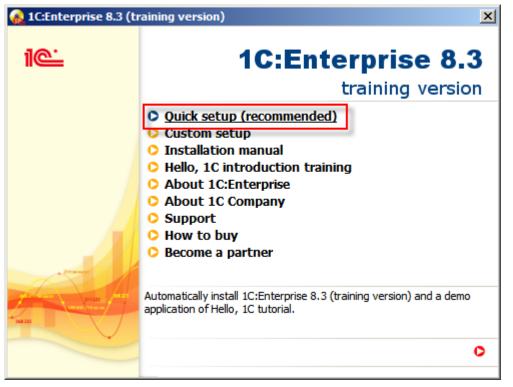


Figure 1-1. Installing 1C:Enterprise 8 (training version)

The installation procedure is simple. Agree with all default options, and continue clicking **Next** until the installation is complete.

Infobase

1C:Enterprise is not a universal IDE. You cannot create *any type* of program using 1C:Enterprise. It is designed for automation of business and individuals. For this reason, many concepts are already embedded into the heart of 1C:Enterprise, its technological platform.

Anywhere where there is 1C:Enterprise, there will be a technological platform. This provides uniformity so that the development and modification technology, as well as structure of 1C:Enterprise applications are always the same.

The major technological concept used in 1C:Enterprise is **Infobase**. Roughly, each infobase is a single 1C:Enterprise application. The unique characteristic is that each infobase contains not only the data the user works with, but also the program, which is named applied solution, that is executed by the platform. For example: 1C:AccountingSuite or 1C:Small Business both are applied solutions.

Thus, when you need 1C:AccountingSuite application, create an infobase with 1C:AccountingSuite applied solution and keep your finances accounting data in this infobase. You can create multiple infobases for keeping information for different companies with the same 1C:AccountingSuite applied solution, but different data. If you will need 1C:Small Business, you will have to create an infobase with another applied solution, 1C:Small Business. In that infobase you will keep goods production accounting data rather than finances accounting in 1C:AccountingSuite infobases.

Thus, if you have an infobase, you have everything you need for work: data and an applied that knows how to manage it.

The creation of any 1C:Enterprise applied solution starts with the creation of an infobase, where that applied solution and data, managed by it, will be stored.

Getting started. Start 1C:Enterprise: click **Start**, and then point to **All Programs**. Point to **1C Enterprise 8 (training version)**, and then click **1C Enterprise (training version)**.

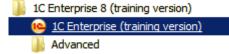


Figure 1-2. Starting 1C:Enterprise (training version)

The first thing 1C:Enterprise will do is open a list of available infobases. If you installed 1C:Enterprise (training version) that includes this tutorial, you can find the **Hello, 1C (demo)** infobase there. In this tutorial, you will create the same infobase step-by-step. Click **Yes** to add a new infobase to the list.

1C:Enterprise startup (training version)	×
Infobases	
IC:Enterprise	
Designer	
1C:Enterprise ×	_
Yes No Add Infobase list is empty. Add Add Infobase to the list? Add Remove Remove	
Settings	
Exit	

Figure 1-3. Adding a new infobase to the empty list

Notice: If you already have infobases in the list, there will be no suggestion to add a first infobase. In this case click **Add...** to add an infobase.

1C:Enterprise startup (training version)	×
Infobases	
	IC:Enterprise
	Designer 🛛
	Add
	Change
	Remove
	Settings
	Exit

Figure 1-4. Adding a new infobase

Now the wizard will ask what you would like to add to the list of infobases: a completely new infobase, which does not exist yet, or an existing infobase, for example, the one that already present on a local network server.

The default option is to create a new infobase, and this is what currently required. Thus click **Next**.

Add Infobase/group X
Adding an infobase to the list Creating an infobase Creating an infobase based on a supplied configuration, supplied demo infobase or creating an empty infobase without configuration
Adding an existing infobase to the list Adding an earlier created infobase located on this computer, in a local network or on 1C:Enterprise server to the list
< Back Next > Cancel

Figure 1-5. Creating a new infobase

There are two ways to create a new infobase: an empty infobase that will contain neither data nor applied solution, or use a template that can contain an applied solution and maybe even demo data. To create an infobase with preset application you can use first option on the next page of the infobase creation wizard.

Since you have not installed any template yet, the wizard will set an option to create a new empty infobase as default and you can continue by clicking **Next**.

Add Infobase/group	×
O Creating an infobase based on a template	
Select supplied configuration to start working or a demo sample to familiarize yourself:	
	^
Creating an infobase without configuration to develop a new configuration or to restore an earlier dumped infobase	~ n
< Back Next > Cancel	

Figure 1-6. Creating an infobase without an applied solution

The wizard will now ask for a name that you would like to give to the new infobase. The infobase name does not affect anything so any name can be used, but for referring matter name it **Hello, 1C**. This name will be displayed in the infobase list at startup. Click **Next**.

Add Infobase/group						
Please specify infobase name: Hello, 1C						
Select infobase location type:						
 On this computer or on a local network computer 						
On 1C:Enterprise server						
On web server						
< Back Next > Cancel						

Figure 1-7. Specifying the infobase name

Add Infobase/group
Please specify infobase parameters:
Infobase directory:
C:\hello_1c
Language (Country):
English (United States)
< Back Next > Cancel

Figure 1-8. Specifying the infobase location

On the next page you need not to change anything, click **Ready**.

Add Infobase/group		×					
Please specify startup options:							
Authentication version (user de	Authentication version (user defined):						
 Select automatically 							
O Prompt for name and password							
Connection speed:	Normal						
Additional startup options:							
Default run mode:							
 Select automatically 							
◯ Thin client							
◯ Web client							
◯ Thick client							
1C:Enterprise version:	8.3						
	< Back Ready Cancel						

Figure 1-9. The infobase is ready

As a result you will see the new infobase named **Hello, 1C** in the list of infobases. Click **Designer** to start developing the **Hello, 1C** application.

Designer mode is the developer mode of running 1C:Enterprise platform that is used to create or modify applications that already exist in infobases.

1C:Enterprise mode is a user mode for working with the data stored in infobases.

Hello, 1C

In this tutorial you are a developer, therefore click **Designer**.

1C:Enterprise startup (training version)	×
Infobases	
Hello, 1C	🦲 1C:Enterprise
	🔛 Designer
	Add
	Change
	Remove
	Settings
File="C:\hello_1c";	Exit

Figure 1-10. Running the Designer mode

Start "programming"

It is no accident that word "programming" is used within quotation marks in the title. Programming itself in its common use, writing code, is an essential part of 1C:Enterprise, but not the number one.

Since 1C:Enterprise, as it was said in Introduction chapter on page 4, is a problem-oriented platform, it hides from the developer many boring routine actions. All 1C:Enterprise applications are built using the same design methods, each application is put together from ready-to-use building blocks. The number of building blocks is limited, the platform has implemented functions of all of them and how they interact between each other.

Therefore, a developer simply needs to add necessary building blocks to the infobase and it will work right away. The platform itself will ensure the proper functioning of building blocks.

Of course, the number of default functions is quite limited. In practice, everything is much more interesting and full of surprises. For this, there is a built-in script language, named **1C:Enterprise script**, and a built-in query language, named **1C:Enterprise query language**. With them, you can adjust the behavior of building blocks, define algorithms of interaction between building blocks, implement your own data processing algorithms, etc.

As it was said in Introduction chapter on page 4, this tutorial is about writing "Hello, world!" using 1C:Enterprise. Therefore, you are going to use less of script and use close to no query language. The use of script and query language in 1C:Enterprise is a complicated subject that requires a separate book.

The current task is to create a simple application from building blocks. In other words, to demonstrate basics of the development process. Adding bells and whistles, improvement, and modification can be done later, if you are interested and have time to do it. However, it is still important that even in such a skeleton form, the application will be fully functional with minimum efforts.

Notice: To make sure, you can create an application with similar functions using any other universal IDE, and compare the necessary knowledge and time that you spent.

Now, get back to building blocks. Since purposes of 1C:Enterprise applications are predefined, building blocks are not abstract, but problemoriented and related to practical entities. For example there are building blocks of catalogs and documents classes, which business and individuals use in daily work.

In 1C:Enterprise, these building blocks are called configuration objects. All configuration objects are grouped into a tree. Thus, by looking at the **Configuration** object tree, you can observe the architecture of any application. You can quickly locate the object and learn its properties.

Now the configuration is opened after you clicked **Designer** in the end of previous chapter. To see the **Configuration** object tree, you need to click **Open configuration** \square .

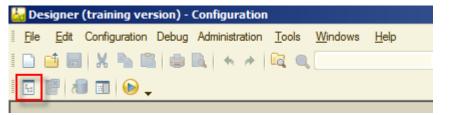


Figure 2-1. Opening the Configuration object tree

For now this tree is empty, it contains only top-level nodes, which can be found in any 1C:Enterprise infobase.

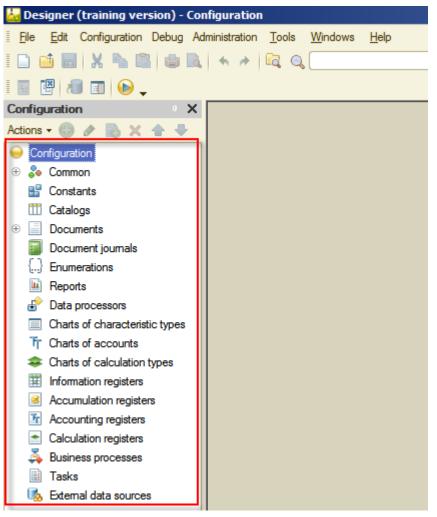


Figure 2-2. the Configuration object tree

The things that you will do next are adding configuration objects that are the basis of the **Hello, 1C** application. Along the way you do it, there will be some "bells and whistles" added, but without of going into much detail.

What configuration objects shall you add? It depends on the purpose of the automation, which is reflected in project requirements.

In this tutorial, according to the project requirements you will create a simple personal CRM (Customer Relationship Management) application. On the one hand, this application will store information about all your friends and acquaintances, in other words, maintain a contacts database. On the other hand, it will track different kinds of events, both past and future. At the same time, it will be able to track financial activities: receipts and expenditures related to both your friends and events in your live. In

addition, there will be added some features to make use of the application simple and easy.

That is it, nothing too complicated.

Subsystems

The first step is to define subsystems. The function of a subsystem is to group configuration objects by their business purpose. Using subsystems in the future, for example, will allow you easily create the application interface. An application interface is how an application provides a user with an access to its functionality.

First, add several subsystems. Later, when adding new configuration objects, you will include them to these subsystems.

If observe entire the project requirements that were described in Start "programming" chapter on page 4, you can see that there are three separated business purpose groups:

- Processes related to people.
- Processes related to events.
- Processes related to financial activities.

That is why you will create three subsystems: **Contacts**, **Events**, and **Finances**.

There is a common way to add configuration objects. Right click the branch of the configuration tree where the necessary objects will be located, and from the context menu choose **Add** \bigoplus (Ins) command.

The **Subsystems** branch is located in the **Common** group of the **Configuration** tree. Add a subsystem.

🕌 Designer (training version) - Configuration
<u>File Edit</u> Configuration Debug Administration <u>T</u> ools <u>W</u> indows <u>H</u> elp
I 🗅 📫 📰 X 🗞 🛍 🖷 🔍 🖘 🏕 🕰 🔍
I 🐷 🖼 🖅 🐨 -
Configuration X
Actions - 🕀 🖉 🔝 🗶 🎓 🗣
Configuration
😑 🐎 Common
Common m All subsystems
Session par 🕀 Add Ins
Roles
- Common attributes
🐉 Exchange plans
Filter criteria
Event subscription

Figure 3-1. Adding a subsystem

Hello, 1C

On the right side a new configuration object editor will open. This window contains properties of the new subsystem. Name the first subsystem **Contacts**. 1C:Enterprise will automatically add a value to **Synonym** property once you press the **Enter** or the **Tab** keys being positioned in the **Name** property. For now, there is no need to change anything else in this window, so click **Close**.

🕌 Designer (training version) -	Соп	figuration					
Eile Edit Configuration Debug	Ad	ministration <u>T</u> ools <u>W</u> indo	ws <u>H</u> elp				
I 🗅 🧉 🗟 🗶 🗞 🗳 🍅	G,	-		- X 🛱 🛱	🗅 🖹 🎾 D	(i) _	
i 📴 🔡 🔊 🗊 🕑 🖕							
Configuration *	×	Subsystem Contact	s				_ 🗆 ×
Actions 🗸 🔂 🖉 🕼 🗶 🔂 🍣	-		-				
Configuration		▶ Main	Name:	Contacts			
😑 🍣 Common		Functional options	Synonym:	Contacts			5 H
⊝ වීද Subsystems		Content	Synonym.	Contacts			
🛃 Contacts		Other	Comment:				
Common modules			Include in com	mand interface:			
Session parameters			Common	nd interface	_		
Y Roles							
Common attributes			Explanation:				
🕌 Exchange plans							
Filter criteria							
Event subscription			D: 1				
C Scheduled jobs			Picture:			×	
Functional options							
Defined types							
Setting storages							
Common forms							
Common commands							
Command groups							
Common templates Common pictures							
ZDTO packages	e						
Web-services			Actions 👻	< Back	Next >	Close	Help
web-services							

Figure 3-2. Creating the Contacts subsystem

In the same way, create other two subsystems: **Events** and **Finances**. As a result, you will have the following tree:

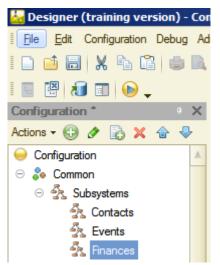


Figure 3-3. The Configuration object tree with newly added subsystems

You added service building blocks. Next, you will add business building blocks.

Give a name to the applied solution. right now, it has a generic name **Configuration**. In addition, at the same time, learn one more way to edit object properties, which is the **Properties** window. To open this window, double-click the top row of the **Configuration** object tree or right-click on the top, and then click **Properties** (Alt+Enter).

Now, in the same way as you did for subsystems, type in the **Name** property the name of applied solution: **Hello1C**. Then edit the automatically generated value of the **Synonym** property to **Hello, 1C**.

🕌 Designer (training version) - Cor	nfiguration	
File Edit Configuration Debug Ad	lministration <u>T</u> ools <u>W</u> indows <u>H</u> elp	
I 🗅 🥶 🔜 X 🍡 🛍 🖷 🔍	🛧 🏕 🚾 🔍	- X
i 🗉 📳 🔝 🕞 🖕		
Configuration * X	Properties	• x
Actions 🗸 🜐 🖉 📄 🗶 🛧 🕂	21 🖼 🔟 🗙 🗸	
\varTheta Hello 1C 🔺	▼Genera <mark>l:</mark>	
😑 🗞 Common	Name Hello1C	
⊖ Subsystems	Synonyn Hello, 1C	
States	Comment	
🖧 Events		
දීදු Finances	Default run mo Managed application	-

Figure 3-4. Applied solution properties

You can verify on your own that there is a shortcut to nearly any property of configuration object in the **Properties** window. In this tutorial, you will use **Properties** on numerous occasions.

Catalogs

By the project requirements, the applied solution must be capable of storing a list of people that the user contacts with and a list of events that are related to the user finances. In addition to storing data, those lists must be capable of classifying it. For the list of people, it must support the following classification: friend, acquainted, family, and so on. For list of events also must let to classify events by status.

Thus, for meeting these requirements, you will create three catalogs. For friends and other people create the **People** catalog. For contact information of various type (phone, address, email, and so on) create the **Contact types** catalog. For storing types of people relation to user (family, friend, acquaintance, and so on) create the **People relation types** catalog.

Another two catalogs will be used to store data regarding events: the **Events** catalog, where the actual data on occurred and planned events will be stored, and the **Event categories** catalog, intended to store categories of events (for example, study, sport, vacation, and so on).

Catalogs are located in the **Configuration** tree branch named **Catalogs**. Add a new catalog, for this right-click the **Catalogs** branch, and then click **Add** (Ins).

😹 Designer (training version) - Con	figuratio
Eile Edit Configuration Debug Ad	ministration
🗅 🥶 🔚 X 🗞 🛍 🖷 🖻	* *
i 🗉 🔛 🔊 🖬 🔞 🖕	
Configuration * X	
Actions 🛛 🕑 🥒 🔝 🗙 🛧 🖶	
\varTheta Hello1C	
🕀 🐉 Common	
🔡 Constants	
Catalog 🕂 🕀 Add Ins	
+ Documente	
Document journals	

Figure 4-1. Adding a catalog

When you add a catalog, a new configuration object editor is opened. You already saw this window when created subsystems. Catalog is a complex object to configure, therefore, this editor is intended to ease and speed up the configuration procedure.

Although the same catalog properties can be specified in Properties on the right, it is more convenient to use this dialog window. Following all the tabs of this editor ensures that all the necessary properties are filled in and nothing is missed.

Catalog Catalog1		_ 🗆 ;
Main	Name: Catalog	4
Subsystems	Name.	
Functional options	Synonym:]
Hierarchy	Comment:	
Owners		
Data	Object presentation:	
Numbering		
Forms	Extended object presenta	ation:
Input fields		
Commands	List presentation:	
Templates		
Generation	Extended list presentation	
Rights		
Data exchange	Explanation:	
Other		
Actions	< Back N	lext > Close Help

Figure 4-2. The Catalog configuration object editor

Catalog Contact Ty	pes	_ 🗆 ×
Main	Name:	Contact Types
Subsystems	0	
Functional options	Synonym:	Contact types
Hierarchy	Comment:	
Owners		
Data	Object presentat	ion:
Numbering		
Forms	Extended object	presentation:
Input fields		
Commands	List presentation	:
Templates		
Generation	Extended list pre	sentation:
Rights		
Data exchange	Explanation:	
Other		
Actions	▼ < Back	Next > Close Help

Type **ContactTypes** as the catalog name.

Figure 4-3. Adding the name and clicking Subsystems tab

There is nothing else to add on the **Main** tab, so click the **Subsystems** tab. On this tab, specify that this catalog belongs to the **Contacts** subsystem.

Hello, 1C

Catalog Contact Ty	pes _ 🗆 X
Main Subsystems Functional options Hierarchy Owners Data Numbering Forms Input fields Commands Templates Generation Rights Data exchange Other	Subsystems that include the catalog:
Actions	< Back Next > Close Help

Figure 4-4. Specifying the subsystem

The storage for contact types is configured.

To start the application in the 1C:Enterprise mode from the Designer mode, click **Start Debugging** (F5). In general, this is not the 1C:Enterprise mode, but the Debug mode, but in this tutorial this is not important. Now look at how the **Contact types** catalog looks like.

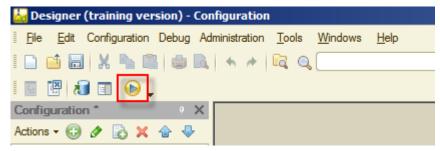


Figure 4-5. Start debugging

The platform will prompt you to update the database configuration, confirm this process.

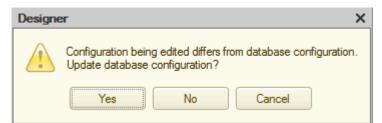


Figure 4-6. Updating the database configuration

The platform then will analyze the changes that had been made in configuration objects and inform you that a new object is added, named **Catalog.ContactTypes**.

Design Changes	<u> </u>
Changee in configuration design	Accept
 New object: Catalog.ContactTypes 	Cancel
P	

Figure 4-7. Restructuring data

Accept changes by clicking **Accept**. This type of extensive system check might seem odd to you, but for large infobases that contain hundreds of configuration objects, it is necessary.

After a short period, the platform will start in the 1C:Enterprise mode. You will see a blank main window of your application that will contain four sections, generated by 1C:Enterprise 8. These sections are quick menu, and previously added subsystems: **Contacts**, **Events**, and **Finances**.



Figure 4-8. Subsystems in the 1C:Enterprise mode at first run

The catalog that you have just created is included in the **Contacts** subsystem. Thus, click **Contacts**, and see the **Contact types** link on the screen.

After clicking this link, you will see a list of contact types, which is currently empty.

To reduce the amount of manual data entry, this tutorial includes an import utility and the sample data needed to populate the catalog. This utility is the data processor that will fill in the **Contact types** catalog data without entering the data manually.

The name of the data processor file is **ImportXMLData83.epf**. You can find it in the current distribution, for this, run **setup.exe**, click **Custom setup** and then click **Hello, 1C additional files**. In opened folder, you can find the data processor. Copy it and all other files in that folder to your computer.

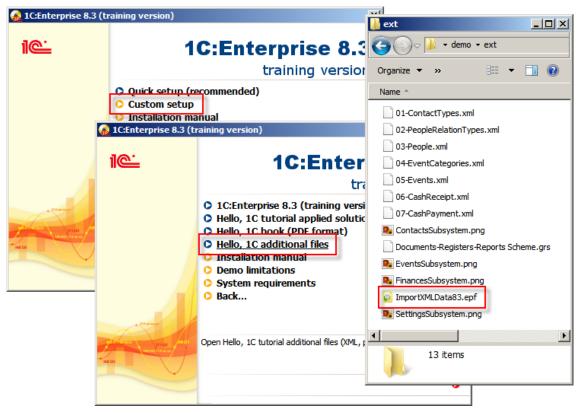


Figure 4-9. The data processor file

Important! This tutorial is designed and thoroughly tested using 1C:Enterprise version 8.3.5. It includes a few synchronous method calls. In 1C:Enterprise versions 8.3.6 or later synchronous method calls are disabled by default. If you are using 1C:Enterprise version 8.3.6 or later, double-click the root configuration node to open the property palette. Scroll down the property palette to find the **Synchronous call usage mode for extensions and add-ins property** and change its value to **Use**.

To run this data processor, click **Main menu** \bigcirc that is at the top left corner of the main application window.

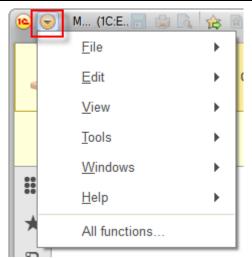


Figure 4-10. Main menu

In **Main menu** click **File** and then click **Open...** (Ctrl+O). In the opened dialog select **ImportXMLData83.epf**, and click **Open**.

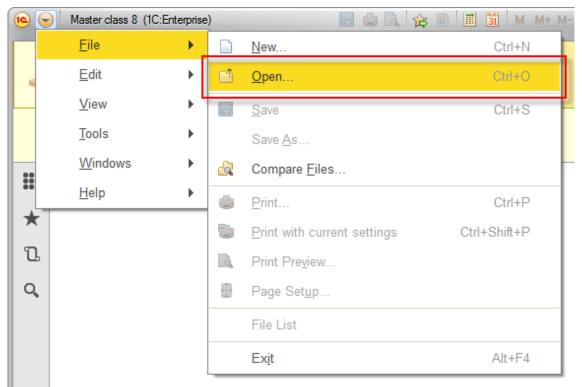


Figure 4-11. Opening the file

The data processor will let you to import the data from XML files.

• •) M (1 🔚 🗐 🗟 🏤 🖹 🗐 🗃 M M+ M- 🥡 🚽 🗕 🗆 🗙
4	Quick menu Ocontacts Events ()
	← → Import XML data 8.3.2.0 ×
\star	p
IJ	More 🔻
Q,	Parameters
	File name:
	Import
	▶ Import
1	Current calls: 0 Accumulated calls: 17

Figure 4-12. Importing data from XML file

In the opened data processor, click **Select** ..., and then find and select the **01-ContactTypes.xml** file.

	> Impo	ort XML data	8.3.2.0 ×
			More -
Paramete	ers		
File name:			
Import			
► Impo	ort		
🙆 Select the (data file		
00	↓ + Local Disk (C;) + 1c_disc	✓ demo ✓ ext	🔹 🐼 Search
Organize 🔻	New folder		
01-Contac 02-People	ctTypes.xml RelationTypes.xml		
🔮 03-People			
	Categories.xml		
05-Events 06-CashR			
07-CashR			

Figure 4-13. Selecting a file to be imported

After selecting the file, click **Import**. Shortly, the data processor will notify you that the data is imported.

← → Import XML data 8	.3.2.0 ×	
	More 🔻	
Parameters		
File name: C:\demo\ext\01-ContactTypes.xml		
Import Import		
Messages:	×	
 Import started at: 3/5/2014 12:30:48 PM Imported objects: 6 Import finished at: 3/5/2014 12:30:48 PM 		
Master class 8		×
Importing data:		
3/5/2014 12:30:48 PM: Import completed.		

Figure 4-14. Importing the data

Close the data processor and open the list form of the **Contact types** catalog. If this form is already opened, press the F5 key to update it.

	M (1C:Enter. 🔚 😂 🗟 🎓 😭 🗐 🛐 M M+ M- 🕡 🚽 🗕 🗆 🗙
	Quick menu Contacts Events
Conta	act types
Ⅲ ★	← → ☆ Contact types ×
ū	Create Find Cancel search More -
Q,	Description ↓ Code
	- Address 00000002
	- Email 00000004
	- ICQ 00000005
	- Phone 00000001
	- Skype 00000006
📑 Q	urrent calls: 0 Accumulated calls: 8

Figure 4-15. The list of items in the Contact types catalog

See that the data is correctly imported.

Notice that you did not create the list form for this catalog. A list form of a catalog is a list where all items of this catalog can be found. 1C:Enterprise generated this form for you automatically based on how you defined this catalog in the **Configuration** object tree. At the same time, all catalog commands are available, user can create new item, delete, search, and execute other actions.

Excellent. Close the main application window and return to the Designer mode. Here you will create remaining catalogs, and during that, you will learn about their individual properties.

Now create the **People relation types** catalog similar to the **Contact types** catalog. The only difference is that on the **Data** tab you will have to increase the length of the **Description** attribute to up to 150 characters.

Catalog PeopleRela	Code length:
Subsystems	
Functional options	Description length: Code type Default presentation
Hierarchy	Number As code
Owners	String As description
Data	
Numbering	🕒 🖉 🗙 🛧 🕂 🛄
Forms	In Attributes
Input fields	
Commands	
Templates	
Generation	
Rights	胡 ち 夕 × 合 チ 目
Data exchange	Tabular sections
Other	
	Standard attributes Characteristics
	Common attributes
Actions	<back next=""> Close Help</back>

Figure 4-16. The People relation types catalog

Start the application in the 1C:Enterprise mode again and import the prepared People relation types data from **02-PeopleRelationTypes.xml**.

Because of changes that had been made previously, the **People relation types** catalog will be displayed as follows:

• •	M (1C:Enter. 📰 📄 💽 🏤 😭 📰 🛐 M M+ M- 🥡 🚽 🗕 🗆 🗙				
4	Quick menu Contacts Events				
Con	tact types People relation types				
	\leftarrow \rightarrow \overleftrightarrow People relation types \times				
+ ₪	Croate Find Cancel search Mare -				
Q,	Description 4 Code				
	- Acquaintance 00000005				
	- Buddy 00000004				
	- Family 00000001				
	- Friend 00000002				
	- Girlfriend 00000003				
Current calls: 8 Accumulated calls: 16					

Figure 4-17. List of items in the People relation types catalog

Now, create a primary catalog, named **People**, where all people that a user knows will be stored. Same as previously created catalogs, include this catalog to **Contacts** subsystem. Then, click the **Data** tab, set length of the **Description** attribute to 150 characters. This attribute will be used to keep names of people.

Up to this point, it was nothing special, you have just created another catalog that will keep the list of names of friends and acquaintances and store those names in the **Description** attribute. However, there is very little benefit comparing to simply storing a list of names in a spreadsheet document. That is why you will use more features of 1C:Enterprise 8 and add more attributes to store additional information about people.

To store additional information, you can add any number of catalog attributes. In this tutorial, start with the following attributes: **Gender**, **RelationType**, and **Comment**.

- **Gender** attribute will store information about gender of people.
- **Relation type** attribute will store types of relation between users and people.
- **Comment** attribute will store any additional information about people that might be necessary to keep.

To add **Gender** as a new attribute, click the **Data** tab, then click **Add (Ins)** for the **People** catalog.

In **Properties**, enter **Name**, and **Synonym** once again will be generated automatically. There is one more thing to do. You need to specify the way to store data in the **Gender** attribute. As you can see in **Properties**, the default type for an attribute is **String**.

🛄 Catalog People		_ 🗆 ×	Properties: Gender X
Main	Code length:	9 🗘	21 🖼 🖌 🗸 🗸
Subsystems	Description length:	150 \$	▼General:
Functional options	Code type		Name Gender Synonym Gender
Hierarchy	Number As code		Comment
Owners	String As description		
🕨 Data			Type String 🗨
Numbering			Length 10 🗘
Forms			Allowed Lengtł Variable
Input fields	Gender		Open-ended ▼ Use:
Commands			

Figure 4-18. Adding the Gender attribute

You can leave everything as it is, but in that case, every time when adding a new person to the catalog, the user will have to enter gender manually in the **Gender** attribute. First, typing information manually for a large number of records is a very time consuming job. Second, there is a high risk of human error such as typo, wrong information input, and inconsistent values as there are many variations of writing genders. These errors, regardless of their nature, will make complicated any future automatic analysis of this data.

The first thing that may come to mind is to create the another catalog where specify both genders, and then use those catalog items. It is a reasonable idea, but creating a new catalog for just two objects is a redundant task. Which way is better then? There is another, new for you configuration object.

Enumeration is an object that is created and filled with data at the development of an applied solution stage. An enumeration can store a limited, known beforehand, and same type of data. **Gender** is precisely the type of data that is good to be stored as an enumeration, rather than as a catalog.

To create a new enumeration, leave **People** catalog as is for now and find **Enumerations** section in the configuration tree, then repeat the already familiar process of adding a new configuration object.

Rapid application development tutorial

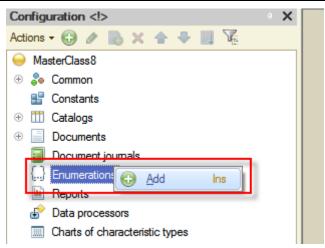


Figure 4-19. Adding an enumeration

In the opened window, as the **Name** of the new enumeration enter **Genders**. Then include the new object to the **Contacts** subsystem.

Enumeration Genders				
Main	Name: Genders			
Subsystems				
Functional options	Synonym: Genders			
Data	Comment:			
Forms				
Input fields	List presentation:			
Commands	Extended list presentation:			

Figure 4-20. The Genders enumeration

Click the **Data** tab, and then add two values using **Add** ⁽¹⁾(Ins), those values are **Male** and **Female**.

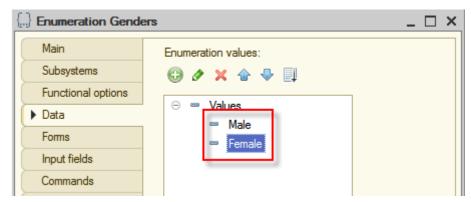


Figure 4-21. Values of the Genders enumeration

After the values are added, close the enumeration editor by clicking **Close**. You will see the previously opened **People** catalog editor. Now you can change the type of the **Gender** attribute from **String** to the newly created **Genders** enumeration. To do that, select the **Gender** attribute and in the **Properties** window in the **Type** property click **Select** ... button. If you accidently closed the **Properties** window, you can always open it by rightclicking on the configuration object and selecting **Properties** or by doubleclicking the attribute.

Properties: Gender X
ti 🛃 🖉 × →
▼General:
Name Gender
Synonym Gender
Comment
Type String 💌
Length 10 🗘
Allowed Leng Variable
Open-ended
▼Use:

Figure 4-22. Changing the type of the Gender attribute

In the opened window, find and select the **Genders** enumeration, then click **OK**.

Edit data type
Composite data type
□123 Number
□ ^a _b _c String
Date
Boolean
🗌 💾 ValueStorage
⊕ CatalogRef
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
Genders Genders Call ChartOfCharacteristic TypesRef
ChartOfAccountsRef
ChartOfCalculationTypesRef
BusinessProcessRef
□ r BusinessProcessRoutePointRef
TaskRef
ExchangePlanRef
□ 🖓 AnyRef
OK Cancel



The accuracy of the selection for the type can always be verified in the **Properties** window.

Properties: G	Properties: Gender X				
21 📑 🚡	× ×				
General:					
Name	Gender				
Synonym	Gender				
Comment					
Type EnumRef.Genders					

Figure 4-24. Verifying the attribute type

After making sure that everything is done correctly, return to the **People** catalog editor, and continue adding new attributes. Next attribute to be added is **RelationType**. Similar to **Gender**, add **RelationType** as an attribute and then select **PeopleRelationTypes** as its type, which is placed in the **CatalogRef** group in **Edit data type** window.

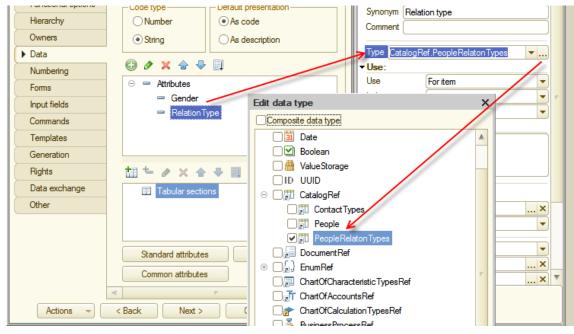


Figure 4-25. Creating the RelationType attribute

The last attribute to be added is **Comment**, which will keep notes and thoughts about a person. When adding this attribute, accept the default data type, **String**. However, the length value needs to be changed. To avoid any guessing while determining the appropriate length of comments, simply set the length of the **Comment** attribute string to **Open-ended**. For this select **Open-ended** check box in **Properties**.

Properties	Properties: Comment				
ai 📑 🛛	í × ×				
→General:	:				
Name	Comment				
Synonym	Comment				
Comment					
Туре	String	▼			
Length	0				
Allowed L	ength Variable	-			
Open-end	ed 🗹				
Use	For item	-			

Figure 4-26. Setting length of Comment attribute string to Open-ended

Keeping in mind the task requirements, you should create another set of attributes where contact information of people will be stored. You might conclude that since you have created **Contact types** catalog, then should add the **Contact type** attribute of the **CatalogRef.ContactTypes** type, that will keep the information about the type of the contact. Then add at least one more attribute that will store the value of that contact, and name it **Contact value**. In other words, to store a person's address, you will have to select the **Address** as the type of the contact, and then input the actual address in **Contact value** attribute.

At this point, consider one more thing. Once you will need to keep more than one contact, if you will store them in the same way as described above, you will have to add as many pairs of **Contact type/Contact value** attributes as you need to keep a required variety of contacts. But it breaks down the efficiency of the storage since every person will have own pairs of data. Furthermore, every time a new type of contact appears, you will be required to add a new pair of attributes. This procedure have to be repeated each time a new piece of information occur.

Fortunately, there is no need to do that. It is clear that every person has an own set of contact information. 1C:Enterprise 8 solves this problem by allowing you to create tables for each catalog item.

Add the **Contacts** tabular section for **People** catalog. Click **Add tabular section the**, and then name the table **Contacts**.

Rapid application development tutorial

III Catalog People		_ 🗆 ×	Properties: Contacts
Main Subsystems Functional options Hierarchy Owners Data Numbering Forms Input fields Commands Templates Generation	Code length: Description length: Code type Number String Comment Code type Default presentation As code As description Comment Comment	9 🛟	A X Image: Contacts Vame Contacts Synonym Contacts Comment Image: Contacts Vata: Standard attributes Open Vase: Use Use For item Image: Contacts Tooltip Image: Contacts
Rights	📶 🏪 🖉 🗙 🐨 🗣 💷		Fill check No check
Data exchange			
Other	Contacts Standard attributes Characte	eristics	

Figure 4-27. Adding the Contacts tabular section

Return to the **People** catalog editor, click on the **Contacts** tabular section, and add a new attribute to that tabular section by clicking **Add attribute** ⁺. Name a new attribute **Type** and select the **ContactTypes**, which is located in the **CatalogRef** group as its type.

🚻 Catalog People			_ 🗆 ×	Properties: Type
Main	Code length:		9 🗘	2‡ 📑 🗃 × ✓
Subsystems	Description length:		150 \$	▼General:
Functional options	Code type	- Default presentation -		Name Type
Hierarchy	Number	 As code 		Synonym Type Comment
Owners	 String 	As description		
▶ Data				Type CatalogRef.ContactTypes
Numbering	🖯 🖉 🗶 🔂 🗣			▼Use:
Forms	\odot 🛥 Attributes			Index Do not index
Input fields	😑 Gender			Full Text Search Use
Commands	Relation Ty			Presentation:
Templates	Comment			Tooltip
Generation				
Rights	to to a x 🎄	- ₽		
Data exchange		_,		Fill check No check
Other	Tabular sectio Contacts	ns		Choice Folders a Items
Other				Choice paramete
	— Туре			· · ·
				Choice paramete
	Standard attribution	Chamatari	rtica	Selection Form

Figure 4-28. The Type attribute of the Contacts tabular section

In the same manner, add the **Value** attribute, which is going to be a **String** with length of **1,024** characters.

Hello, 1C

Catalog People	_ 🗆 >	Properties: Value
Main	Code length: 9 \$	\$‡ 🖼 🖉 × ✓
Subsystems	Description length:	▼General:
Functional options	Code type	Name Value Synonym Value
Hierarchy	Number As code	Comment
Owners	String As description	
▶ Data		Type String
Numbering	O ≠ × ↑ ₹ □	Length 1,024 🗘
Forms	⊖	Allowed Length Vanable
Input fields	- Gender	Open-ended
Commands	RelationType Comment	Index Do not index
Templates	- Comment	Full Text Search Use
Generation		✓ Presentation:
Rights	🚹 💺 🌶 🗙 🚭 🖶 💷	Password mode
Data exchange	☐ ☐ Tabular sections	Tooltip
Other	□ □ Contacts □	
	Standard attributes Characteristics	Mask Multiline mode Extended edit
	Common attributes	Fill check No check

Figure 4-29. The Value attribute of the Contacts tabular section

At this point the **People** catalog is considered finished. Take a look what you have now. To start the application in the 1C:Enterprise mode click **Start Debugging** (F5).

Confirm the system request to update the configuration, and then accept the changes to the configuration.

Now go to the **Contacts** section that you have seen already, and look at what you have created.

•	🤨 🕞 M (1C:Enter. 🔚 🖶 🔍 🚖 🗟 🏢 🛐 M M+ M- 🥡 🚽 🗕 🗆 🗙				
4	Quick menu Contacts Events				
Con	tact types People People relation types				
₩ ★	← → ☆ People ×				
ם	Create Find Cancel search More				
Q,	Description ↓ Code Gender Relati				

Figure 4-30. The People catalog in the 1C:Enterprise mode

For now, this catalog is empty. Open the **ImportXMLData83.epf** data processor and import data from the **03-People.xml** file. Press the **F5** key to update the list, if it is empty.

Rapid application development tutorial

و چ	Master class 8 (1C:Enterprise)			🚖 🖻 🔳 📓 M	M+ M- 🥡 🚽 🗆 🗙
4	Quick menu Contacts Contacts Finances				
Contac	t types People Peo	ple relation types			
Ⅲ ★	← → ☆ Pe	ople			×
ם	Create Find	Cancel se	earch		More -
Q,	Description ↓	Code	Gender	Relation type	Comment
	🗕 Georgia	00000002	Female	Girlfriend	sophomore ;)
	= Grandmother	00000001	Female	Family	the best grandm
	 John Sheldon 	00000003	Male	Friend	neighbor
	🛥 Sarah Wells	00000004	Female	Acquaintance	MKTG 101
📮 Cu	ment calls: 1 Accumulated calls:	17			

Figure 4-31. The People catalog, filled with data

Now the list contains people. To verify, which data is contained in individual records, double-click any record.

0000000	D1 (People) (1C:Enterprise)	🖬 🗐 M M+ M- 🗆 🗙			
000000	00000001 (People)				
Save	and close Save	More 🔻			
Code:	00000001				
Description	n: Grandmother				
Gender:	Female	•			
Relation ty	pe: Family	다. · · · ·			
Comment:	the best grandmother in th	ne world			
Add]	More 💌			
#	Туре	Value			
1	Email	granny@1c.com			
2	Skype	granny			
3	Address	785 5th Avenue, New York			
4	Web site	http://1c-dn.com/forum/			
5	Phone	(212) 737-92-57			

Figure 4-32. The item of the People catalog

As you can see, main attributes, related to a person, are located on the upper part of the window, and the contact information is conveniently arranged in the tabular section on the bottom.

Note that in addition to generation of forms to display lists of catalog items, the platform also automatically generated a form of catalog items.

Close the main application window and return to the Designer mode to add two more catalogs, which are **Event categories** and **Events**.

The **Event categories** catalog is created in the same way as the **People relation types** catalog. The **Description** attribute length will be 150 characters as well. The only difference is the association with the subsystem. **Event categories** will be included to the **Events** subsystem.

Catalog EventCate	gories _ 🗆 X
Main	Subsystems that include the catalog:
Subsystems	
Functional options	Contacts
Hierarchy	Events Finances
Owners	
Data	
Numbering	
Forms	
Input fields	
Commands	
Templates	
Generation	
Rights	
Data exchange	
Other	
Actions	 <

Figure 4-33. Making Event categories a part of the Events subsystem

In the 1C:Enterprise mode, import the data for this catalog from the **04-EventCategories.xml** file.

•	M (1C:Enter 📄 🍙 🗟 🙀 🔳	🛅 M M+ M- 🥡 🖵 🗆 🗙
4	J Quick menu 🧼 Contacts	Events
Event	categories	
Ⅲ ★	🗲 🔸 🏠 Event cat	tegories ×
ם	Create Find Car	ncel search More -
O,	Description 4	Code
	– Meeting	00000001
	= Rest	00000004
	= Sport	00000003
	= Study	00000002
j⊈≣ C	urrent calls: 0 Accumulated calls: 31	

Figure 4-34. The Event categories catalog, filled with data

The last thing you need to do in this chapter is to add the only remaining catalog, named **Events**.

The **Events** catalog will also be included to **Events** subsystem. Set the length of the **Description** attribute to 150 characters. Then, add following attributes to this catalog:

- Begin date with Type equals to Date and Date Content Date and time.
- End date with Type equals to Date and Date Content Date and time.
- Category with Type equals to CatalogRef.EventCategories.
- **Details** with **Type** equals to **String** and checked **Open-ended** box.

Add a new tabular section, named **Participants** and add an attribute **Participant** of **CatalogRef.People** type to this tabular section.

As the result of these changes, the **Data** tab will look as follows:

Catalog Events	×
Main Subsystems Functional options Hierarchy Owners Data	Code length: Description length: Code type ONumber String As code As description
Numbering Forms	⊕
Input fields Commands	 BeginDate EndDate Category
Templates Generation	
Rights Data exchange Other	 Im t→ 𝗨 × Im Im
	Participant
	Standard attributes Characteristics Common attributes
Action	s 🔻 Kack Next > Close Help

Figure 4-35. Properties of the Events catalog on the Data tab

Click **Start Debugging** (F5), and confirm all changes. In the 1C:Enterprise mode import data into the **Events** catalog from the **05**-**Events.xml** file.

Rapid application development tutorial

• •	Master class 8 (1C:Enterprise)	Events	Finances	iii M M+ M- (i)	- 🗆 ×
Event	categories Events				
	← → ☆ Events				×
* L	Create Find Cancel	search		More	•
Q,	Description ↓	Code	Begin date	End date	*
	 Cafe with Georgia 	00000006	10/7/2014 4:00:00 PM	10/8/2014 7:00:00 AM	
	- Club	00000003	10/2/2014 9:00:00 PM	10/3/2014 8:00:00 AM	
	– College	000000004	9/1/2014 12:00:00 AM	2/1/2010 11:00:00 PM	
	 College financial assistance 	00000002	10/5/2014 7:30:00 PM	10/5/2014 7:45:00 PM	
	 Meeting relatives 	00000001	10/1/2014 7:30:00 PM	10/1/2014 11:00:00 PM	
				>	
률 Cu	urrent calls: 0 Accumulated calls: 86				



See the result in one of **Event** items.

College (Events) (1C:Enterprise)				
College (Events)				
Save and close Save More	•			
Code: 00000004	Â			
Description: College				
Begin date: 9/ 1/2014 12:00:00 AM				
End date: 2/ 1/2015 11:00:00 PM				
Category: Study -				
Details: first semester				
Add More -				
# Participant				
1 Sarah Wells				
2 Georgia				



Now you have created everything that is required to keep track of events in life of users.

Excellent! The **People** catalog can store the information about all of user's relatives, friends, and acquaintances, including a variety of contact information of these people. The **Events** catalog stores the information about past and planned events, also participants can be specified here. At the same time, three auxiliary catalogs, **ContactTypes**, **PeopleRelationTypes**, and **EventCategories** help users to customize data, related to primary catalogs.

Moving further.

Register

Now add a building block that may not be obvious from the first sight. It will be the **Financial transactions** accumulation register.

Questions, "What is a register? What is its purpose?" immediately arise.

Below you will find a simple explanation.

According to project requirements, the application should not only be capable of storing a list of friends and tracking events. It also should be capable of keeping records of financial transactions. At a minimum, it should keep records of cash flow and provide simple financial reports.

Registration of various events related to money income and outcome will be implemented by using configuration objects, named **Documents**. Later, you will add a couple of documents to the applied solution.

Together with registering money incomes and outcomes, it is required to create some reports, demonstratively displaying what happens to finances. These reports can be created using data that is contained in user input documents. These documents reflect actual receipts and expenditures of money. However, imagine that in a month (or a year), for example, you decide to supplement your financial records with new documents or functionally expand existing ones.

Making configuration changes, as you have already seen, is a simple task. However, what will happen with reports in this case? It might turn out that financial reports that you created previously will not work properly, because they use only income and outgoing records from the old documents as data source. This means that you might have to rebuild all related to this data reports. Live applied solutions usually contain a large number of reports.

To avoid this work, and a large number of errors that may become a result of this alteration, 1C:Enterprise uses the following development methodology:

To store data regarding activity of accounting subjects, finances for example, configuration objects named **Registers** are used. **Documents** that reflect business activities, record data in these registers. Reports, then, use these registers as data sources.

As a result, you have a separated structure. On one hand, once there is a complete set of registers, it is easy to create required reports that will demonstratively display information contained in these registers. On the other hand, when a new type of document appears in the applied solution, you only need to add the correct algorithm to place data in registers. This

methodology guarantees that previously created reports will not require any alterations and will display correct data.

Docu	iment 1	Documen	t2	Document N
Period	↓ Event		Person	Amount
+ 10/1/2013 9:05	5:09 Meeti	ng relatives	Grandmother	100.00
- 10/2/2013 4:48	3: 1 8		Georgia	50.00
- 10/2/2013 4:48	8:18 Club			12.00
- 10/3/2013 12:0	00:00 Club		Sarah Wells	15.00
+ 10/5/2013 4:02	2:25 Colleg	ge financial assistance		150.00
- 10/7/2013 4:00	0:49 Colleg	je	Georgia	30.00
- 10/7/2013 4:00	0:49 Colleg	je		10.00
Rej	port 1	Report 2	2	Report N

Figure 5-1. The Documents-Registers-Reports schema

So, for meeting project requirements of this tutorial it is enough to create only one accumulation register, name it **Financial transactions**. Where the application will store data on how much and how often a user receives and spends money.

Accumulation registers are located in the **Accumulation registers** branch of the **Configuration** object tree. Add a new accumulation register.

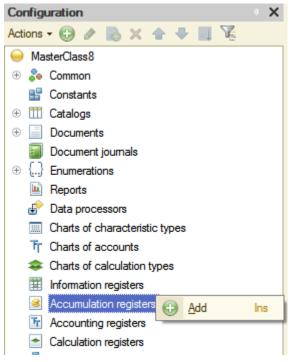


Figure 5-2. Adding an accumulation register

Name this register **Financial transactions** and include it to **Finances** subsystem. Then click the **Data** tab.



Figure 5-3. The Financial transactions accumulation register

Now add dimensions, resources, and attributes to this register.

S Accumulation register Financial Transactions		_ 🗆 ×
Main	⊖ ≠ × + + II	
Subsystems		
Functional options	L→ Dimensions	
▶ Data	- Attributes	
Recorders		
Forms		

Figure 5-4. The ccumulation register Data tab

Resources are the data that you are going to obtain from the register. For **Financial transactions** you need to know the amount of transactions. Hence, only one resource of **Number** type named **Amount** is required.

Dimensions refer to slices of information that is required to be obtained from the register. It is unlikely that you will be interested only in the overall balance. For example, to create a finance flowchart you may be interested in which events or people bring you most of the money, or which consume the most. To track this, you will need to have two dimensions **Event** and **Person**.

Attributes keep additional information that accompanies each record in the register. For **Financial transactions** you will not use attributes. However, nothing prevents you from adding them to the register in future and then keeping any auxiliary information there.

Add mentioned above dimension and resources. To add them, right-click the **Dimensions** and the **Resources** groups of the accumulation register, and in the context menu click **Add** ⁽²⁾ (Ins).

Now, add a dimension.

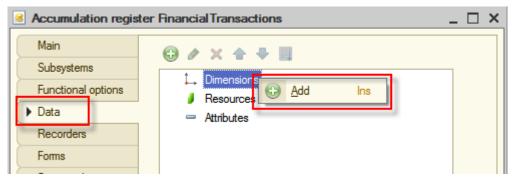


Figure 5-5. Adding a dimension

In the **Properties** window, define that the **Name** of the new dimension is **Event**, and its **Type** is **CatalogRef.Events**.

Propertie	s: Event	×
₽‡ 📑 🧎	lí × ✓	
→ General	<u>:</u>	
Name	Event	
Synonym	Event	
Comment		
Туре Са	talogRef.Events	

Figure 5-6. The Event dimension properties

Add a second dimension, named **Person**. Type is **CatalogRef.People**.

Properties: Person		• :	×			
₽Į 📑	x M	~				
- Genera	1:					
Name	Person	1			D	
Synonyn	Person					
Comment	1	1				
Type Catalog Ref.People						

Figure 5-7. The Person dimension properties

Now add the **Amount** resource. Accept default values of **Type** and **Length**, and adjust only **Precision**, increase it from **0** to **2**.

Properties: Amount			
21 📑 🍯	× *		
▼General:		A	
Name A	mount		
Synonym A	mount		
Comment			
Туре	Number 🗸		
Length	10 🗘		
Precision			
Non-negative			

Figure 5-8. The Amount resource properties

As a result, the **Financial transactions** accumulation register will look as follows:

Accumulation regis	ter Financial Transactions	_ 🗆 ×
Main	<u> 🖉 X 🕁 🖶</u> 🗐	
Subsystems	○ ↓ Dimensions	
Functional options	⊖ L Dimensions	
► Data		
Recorders		
Forms	Amount	
Commands	- Attributes	
Templates		
Rights		

Figure 5-9. The Financial transactions register

Documents

Now, proceed to adding documents that record money income and outgoings. The first document will be **Cash receipt**.

Documents are located in the **Documents** branch of the configuration object tree. Add a new document.

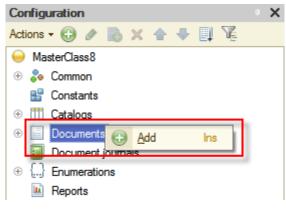


Figure 6-1. Adding a new document

Name it **CashReceipt**, and include it to the **Finances** subsystem, and then click the **Data** tab.

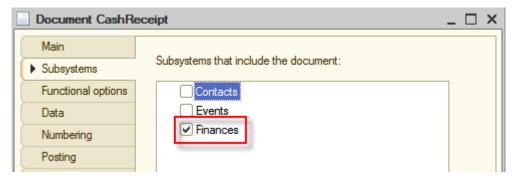


Figure 6-2. Creating the Cash receipt document

Each document, by default, has two attributes, **Number** and **Date** that indicate the sequential number of this document and the date when it was

created. However, in addition to these attributes, it is usually required to have other information regarding specific business activity in the document. For example, in this master class users would like to understand what was the source of money: who or which event. In addition to simplify data input, make possible the same document to register several similar events. For example, it is convenient to group all receipts for one day in one document; this will allow a user to supervise documents that this user enter.

As you might have guessed already, a **tabular section** will help us to accomplish this.

Create the **Receipts** tabular section.

Document CashRee	ceipt	_ 🗆 ×
Main Subsystems Functional options Data Numbering Posting Sequences Journals Forms	 Attributes 	
Input fields Commands	111 🖕 🖉 🗙 🗣 🗐	
Templates	 □ Tabular sections □ □ Receipts 	
Generation Rights		

Figure 6-3. Creating the Receipts tabular section

In order to be able to track events and people, you need to create two attributes in the newly created tabular section, **Event** and **Person**. To track the amount of transactions, create the **Amount** attribute.

- Event with Type equals to CatalogRef.Events.
- **Person** with **Type** equals to **CatalogRef.People**.
- Amount with Type equals to Number, Length equals to 10, Precision equals to 2, and select the Non-negative check box.

Because of these actions, **Data** tab of the document will look as follows:

Document CashRee	ceipt	_ 🗆 ×
Main Subsystems Functional options Data Numbering Posting Sequences Journals Forms	 ★ ★ ★ ↓ Attributes 	
Input fields Commands	ta 🔁 🖉 🗙 🍙 🐥 💷	
Templates Generation	Tabular sections Tractions	
Rights Data exchange Other	- Event - Person - Amount	

Figure 6-4. Adding attributes to the tabular section

Properties: Event			×
21 📑 🚡	× •		
→General:			
Name	Event		
Synonym	Event		
Comment			
Type CatalogRef.Events			
▼Use:			

Figure 6-5. Event attribute

Properties: Person				×		
zi 📑 👔	$\times \ \checkmark$					
■General:						
Name	Person					
Synonym	Person					
Comment	Comment					
Type CatalogRef.People						

Figure 6-6. Person attribute

Rapid application development tutorial

Properties: Amount				×
21 📑 🍯	× ×			
→General:				
Name	Amount			
Synonym	Amount			
Comment				
	-			
Туре	Number		-	
Length	10 \$			
Precision	2 ‡			
Non-negative				
▼Use:	S			

Figure 6-7. Amount attribute

To finish configuring this document, click the **Posting** tab.

As it is said above, documents write data to registers, and reports get this data from registers, and display in a convenient form to users.

Document register records are records that documents make in registers; in this master class records will be made in **Financial transactions** register.

Expand **Accumulation registers** node and check **Financial transactions** accumulation register. By doing this you define that **Cash** receipt document will write records in the **Financial** transactions register. Then click **Register records wizard** and the platform will assist you in creating the data-writing algorithm for registers.

Document CashRee	ceipt		_	×
Main	Posting:	Enable	•	
Subsystems	Real-time posting:	Enable	- T	
Functional options	Register records deletion:	Delete automatically on clei	,	
Data			5	
Numbering	Accumulation regis Financial Tr			
Posting				
Sequences				
Journals				
Forms				
Input fields				
Commands				
Templates				
Generation	Register records wize	ard 🗙		
Rights	FinancialTransactions			
Data exchange				
04				

Figure 6-8. Defining document register records

The **Register Records Wizard** window will open. Leave **Register record type** unchanged, equal to **Receipt**, since this document is a receipt of money. Then select **Receipts** in the **Tabular Section** field and click **Fill Expressions**.

In the table at the bottom of the window, see that the platform automatically found the correspondence between accumulation register attributes and document attributes. The value of the **Event** tabular section attribute of the document will be placed in **Event** register dimension. The value of **Person** tabular section attribute of the document will be placed in the **Person** register dimension. The value of the **Amount** tabular section attribute will be placed in the **Amount** register resource.

Register Records Wizard			□ ×
🖯 🗟 🗙 🛧 🕂			
Registers		Document attributes	
🕂 Accumulation Register. Fin	nancialTransactions	Date	
Register Record Type: Receipt Expense 		 Number CurRowReceipts.LineNu CurRowReceipts.Event CurRowReceipts.Person CurRowReceipts.Amount 	
Tabular Section: Receipt	s 🗸		< <u>B</u> ack <u>N</u> ext >
Field	Expression		Fill Expressions
🔔 Event	CurRowReceipts.Event		Clear Expressions
L. Person CurRowReceipts.Person			
Amount CurRowReceipts.Amount		[ОК
			Cancel
			Help

After filling out and verifying the correspondence table, click **OK**.

Figure 6-9. Register Records Wizard

The wizard will generate the procedure for writing document register records, in other words, for posting the document, and then display it on the screen.

📄 Document CashReceipt: Object module 📃 🖸	×
<pre>Document CashReceipt: Object module ; Procedure Posting(Cancel, Mode)</pre>	×
	•

Figure 6-10. Procedure for posting the document

As you can see, the procedure is quite simple. For each row of the tabular section in the document, the new record is created in the register, and this record contains the data from the row.

You could write this algorithm yourself, but to minimize the amount of work it is better to utilize **Register Records Wizard**. The wizard generated this code automatically, which is mentioned in comments at the begin and the end of the procedure.

Now start the application in the 1C:Enterprise mode by clicking **Start Debugging** (F5), and using the **ImportXMLData83.epf** data processor import demo data into the **Cash receipt** document from the **O6-CashReceipt.xml** file.

Click the **Finances** tab and open the list of **Cash receipt** documents.

•	Master class 8 (1C:Enterprise)	🖨 🖻 🏠 📓 🔳 M	M+ M- 🥡 🚽 🗕 🗆 🗙
4	🕽 Quick menu 🧼 C	ontacts 🧼 Events	Finances
Cash	receipt		
Ⅲ ★	← → ☆ Cash	receipt	×
D	Create Find	Cancel search	More -
Q,	Date ↓	Number	*
	10/1/2013 9:05:09	00000001	
	10/5/2013 4:02:25	00000002	

Figure 6-11. Cash receipt documents in the 1C:Enterprise mode

For now, documents are just imported, and they do not affect reports yet. In order for the data to be displayed in reports, documents have to be posted. When documents are posted, the data is being added to the **Financial transactions** accumulation register.

Open the first document. To post the document, click **Post and close**.

← → ☆ Cash receipt 00000001 from 10/ ×				
Post a	and close Save	Post	More •	
Number:	00000001		í.	
Date:	10/ 1/2013 9:05:09 PM			
Add]		More 🔻	
#	Event	Person	Ar	
1	Meeting relatives	Grandmother		

Figure 6-12. Posting the Cash receipt document

Successful document posting will display a notification on bottom right of the screen.

A	Changed:	×
G	Cash receipt 000000001 from 10/1/2013 9:05:09 PM	

Figure 6-13. Successful posting of the Cash receipt document

The fact that the document posting was successful is displayed as a check mark on the icon of this document in the list of documents.

Date	e ↓	Number
	10/1/2013 9:05:09	000000001
	10/5/2013 4:02:25	00000002

Figure 6-14. Posted document in the list of documents

To verify the way the document was posted, meaning the required information is recorded in the **Financial transactions** accumulation register, open **Main menu** \bigcirc of the application, click **All functions...**, then expand accumulation registers node, then click **Financial transactions**, and then click **Open**.

Notice: The **All functions...** menu item is available by default if an application is started in the debug mode from the Designer mode.

All function	All functions - Master class 8 (1C:Enterprise)				
All func	All functions				
Q Open		?			
8	Constants				
÷ 🎞	Catalogs				
÷ 🗐	Documents				
	Document Journals				
	Reports				
e 1	Data processors				
	Charts of Characteristic Types				
ቸ	Charts of Accounts				
\$	Charts of calculation types				
	Information Registers				
⊖ ≤	Accumulation Registers Financial transactions Accounting registers				
•	Calculation registers				

Figure 6-15. All functions

The list form of the **Financial transactions** accumulation register will open. For now, the register contains only one record that belongs to the first posted document. By double-clicking this record, you can always open the original document.

Notice the ⁺ icon, which means that the values of resources of this record are added to total values of resources.

← → ☆ Financial transactions						×
Find	Cancel search				More	•
Period	Ļ	Recorder	L	Event		-
+ 10/1/	2013 9:05:09	Cash receipt 000000001	1	Meeting relatives	s	
1000 B						

Figure 6-16. Accumulation register record

Switch to **Cash receipt** document list and post the second document. After the posting, return to the list of records of **Financial transactions** register. See that a new entry, belonging to the second document, appeared. You can update the form with F5 key.

← → ☆ Financial transactions				×	
Find	Cancel search			More	•
Period	Ļ	Recorder	L	Event	*
+ 10/1/2013	9:05:09	Cash receipt 000000001	1	Meeting relatives	
+ 10/5/2013	4:02:25	Cash receipt 000000002	1	College financial assi	

Figure 6-17. Accumulation register records

Finished with **Cash receipts**. However, users also need to track outgoings. Go back to the Designer mode. It seems obvious, that it is necessary to create the **Cash payment** document. You could easily create it manually in the same way as you did already with the existing **Cash receipt** document because compositions of two documents are identical. Another option, however, is to use the 1C:Enterprise 8 feature allowing you to create new configuration objects based on existing ones by simply copying them.

To do this, right-click the existing **Cash receipt** document in the list of metadata objects and click **Add by cloning** \square (F9).

Documents						
234 Document numerators						
📑 Sequences	🚽 Sequences					
⊖ 📄 CashReceipt						
- Attribute:	Open Object Module					
🕀 🔟 Tabular i 📴	Open manager module					
E Forms	Open Default Object Form					
🜔 Comman						
🕮 Template	Open Default List Form					
Document journa	Open Default Choice Form					
⊕ {} Enumerations	Open	•				
Reports	140					
🔄 Data processors	Wizards					
Charts of charac	Add					
ी Charts of accour	Change	F2				
🗢 Charts of calcula	N	F9				
📰 Information regist	<i>.</i>					
Accumulation rec	<u>D</u> elete	Del				

Figure 6-18. Adding by cloning an existing document

Once you click this button, a new **CashReceipt1** document appears. It is going to be an exact copy of **CashReceipt**.

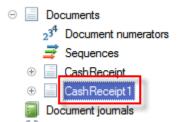


Figure 6-19. A copy of the Cash receipt document.

The only thing left to do is to rename the document and edit some of its properties. Do it.

First, open this document in the document editor editor and change the name of the document to **CashPayment**. Moreover, include the new document to the **Finances** subsystem.

Document CashPayment _ X					
Main		Name:	CashPayment		
Subsystems		Numo.			
Functional opti	ions	Synonym:	Cash payment		
Data		Comment:			

Figure 6-20. The Cash payment document

On the **Data** tab, using **Properties**, change the name of the tabular section to **Expenses**. Keep the remaining attributes unchanged as they meet project requirements.

Document CashPay	ment _ 🗆 X
Main Subsystems Functional options Data Numbering Posting Sequences Journals Forms	Attributes
Input fields Commands Templates Generation Rights Data exchange	tin t- ⊘ × ↔ ♥ II ○ II Tabular sections ○ II Expenses = Event = Person = Amount

Figure. 6-21. Renaming the tabular section

Click the **Posting** tab. Here you will change the document posting. For now the posting algorithm is copied from the **Cash receipt** document. That document keeps income but in the newly created document is designed to keep outgoings.

Open **Register Records Wizard**. The platform will prompt you to confirm that the existing script will be replaced. Click **Yes**.

Document CashPay	ment _ 🗆 X
Main Subsystems Functional options Data Numbering Posting Sequences Journals Forms Input fields Commands	Posting: Enable Real-time posting: Enable Register records deletion: Delete automatically on clei
Templates Generation Rights Data exchange Other	Register records wizard X Financial Transactions Designer Very State X Opening register register records wizard will cause the Posting procedure to be replaced. Do you want to continue? Yes No
Actions	Close Help

Figure 6-22. Opening Register records wizard

Here repeat all the steps that you performed for **Cash receipts** document. You only need to change the **Register Record Type** to **Expense** and select the tabular section **Expenses**. Then click **Fill Expressions**, and to update the module click **OK**.

Register Records Wizard			□ ×	
🕀 🗟 🗙 🛧 🖶				
Registers		Document attributes		
– Accumulation Register. Fina	ncialTransactions	Date		
		Number		
		CurRowExpenses.Line	eNumber	
		CurRowExpenses.Eve		
		CurRowExpenses.Per		
		CurRowExpenses.Am	ount	
Register Record Type:	Receipt • Expense			
Tabular Section: Expenses			< <u>B</u> ack <u>N</u> ext >	
Field	Expression		Fill Expressions	
🛴 Event	CurRowExpenses.Event		Clear Expressions	
🛴 Person	CurRowExpenses.Person			
/ Amount	CurRowExpenses.Amount		ОК	
			Cancel	
			Help	

Figure 6-23. Register Records Wizard

You can see that there are only two differences between two documents, the name of the tabular section and the type of the accumulation register record.

```
Document CashPayment: Object module
                                                                                       _ 🗆 🗙

    Procedure Posting(Cancel, Mode)

      //{{ REGISTER REGISTERRECORDS WIZARD
      // This fragment was built by the wizard.
      // Warning! All manually made changes will be lost next time you use the wizard.
      // register FinancialTransactions Expense
      RegisterRecords.FinancialTransactions.Write = True;
For Each CurRowExpenses Ir Expenses Do
          Record = RegisterRecords.FinancialTransactions.Add()
          Record.RecordType = AccumulationRecordType.Expense;
          Record.Period = Date;
          Record.Event = CurRowExpenses.Event;
          Record.Person = CurRowExpenses.Person;
          Record.Amount = CurRowExpenses.Amount;
      EndDo;
      //}} REGISTER REGISTERRECORDS WIZARD
  EndProcedure
                                                                                            - Br
```

Figure 6-24. Posting procedure of the Cash payment document

Start the application in the 1C:Enterprise mode and, using the data processor, import the **Cash payment** data from the **O7-CashPayment.xml** file. Then, switch to the **Finances** section and open the list of **Cash payment** documents.

Hello, 1C

•	Master cl (1C:Enterprise) 🔚 🍙 🗟 🚖 😭 📰 🛅 M M+ M- 🥡 🖵	- 🗆 ×
4	Quick menu 🧼 Contacts 🧼 Events	ances
Cash	cash receipt	
Ⅲ ★	← → ☆ Cash payment	×
D	Create Find Cancel search More	•
Q,	Date 4 Number	-
	10/2/2013 4:48:18 PM 000000001	
	10/3/2013 12:00:00 PM 00000002	
	□ 10/7/2013 4:00:49 PM 00000003	

Figure 6-25. The Cash payment document list

Post these three documents and then verify that the data is recorded to the **Financial transactions** accumulation register.

Find.	Cancel search				More
Period	d ↓	Recorder	Line num	Event	Person
+	10/1/2013 9:05:09	Cash receipt 00000	1	Meeting relatives	Grandmother
-	10/2/2013 4:48:18	Cash payment 000	1		Georgia
- 1	10/2/2013 4:48:18	Cash payment 000	2	Club	
- 1	10/3/2013 12:00:00	Cash payment 000	1	Club	Sarah Wells
+	10/5/2013 4:02:25	Cash receipt 00000	1	College financial as	
- 1	10/7/2013 4:00:49	Cash payment 000	1	College	Georgia
- 1	10/7/2013 4:00:49	Cash payment 000	2	College	

Figure 6-26. Records in the accumulation register

It is not difficult to understand that records in the register, belonging to **Cash payment** documents are marked with – icon. In addition, it is clear that a few lines of a document are recorded in the register in the form of individual entries. Note the **Line number** column.

At this point, the accounting part of the application is complete. Users are now able to fill a list of acquaintances and enter cash receipts and payments.

However, simple accounting is not enough. Users would also like to receive the information from the application in a convenient form. For this purpose, there are reports.

Reports

The next step in the development of the applied solution will be creation of convenient reports on activities. Reports as well will be created without any programming, using only visual design tools.

Reports are located in the **Reports** branch of the **Configuration** object tree. Let us add a new report.

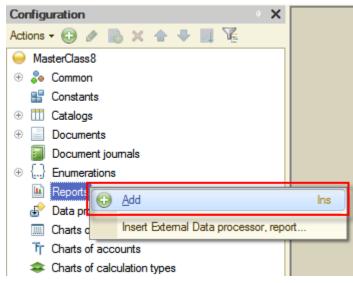


Figure 7-1. Creating a new report

Name the new report **Expenses** and then click **Open data composition schema** of this report.

Report Expenses		_ 🗆 ×
Main	Name:	Expenses
Subsystems		
Functional options	Synonym:	Expenses
Data	Comment:	
Forms		
Commands	Main data comp	position schema:
Templates		×Q
Rights		Open data composition schema
Other		

Figure 7-2. Opening the data composition schema of the Expenses report

Since this is a new report, it does not contain any data composition schema, the platform will open **Template Wizard** and suggest you to create a new template. That template will contain a data composition schema. Agree by clicking **Finish**.

Template Wizard		×
Name:	MainDataCompositionSchema	
Synonym:	Main data composition schema	
Comment:)
- Select template t	ype:	
O Spreadsheet	document	
O Text documer	nt	
O Binary data		
Active docum	ent	
OHTML docum	ent	
Geographical	schema	
Graphical sch		
 Data composi 	tion schema	
	tion appearance template	
Load from file:		
		_
	Finish Cancel Help	

Figure 7-3. Template Wizard

After that, the platform will open the **Data composition schema** editor. For now, this composition schema is empty.

Page Report Expenses: MainDataCompositionSchema					
Data Sets Data	set links Calculated fields Resources Parameters Templates Nested schemas	Settings			
<u>ta</u> - ×		0 ×			
Data Sets	Data Source Name				
		0			
📑 🗐		T			

Figure 7-4. The empty data composition schema

Now you will define sources of data that the report will use to retrieve data, and define the structure of the report.

Add a **Data set** of type **Query**. In other words, the data for this report will be retrieved from the database, managed by 1C:Enterprise.

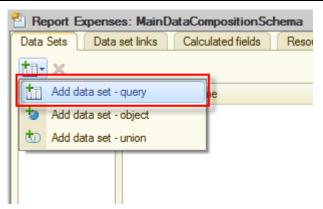


Figure 7-5. Creating a new query

You can enter a query manually or use **Query builder**. Let us use second way, for that click **Query Builder...**



Figure 7-6. Opening Query builder

In the **Query builder** window on the left there is a list of tables that can be used for retrieving data. Expand the **AccumulationRegisters** branch and double-click to select **FinancialTransactions.BalanceAndTurnovers** table.

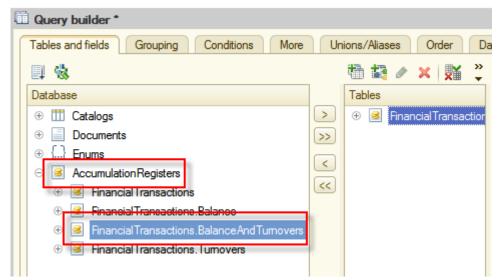


Figure 7-7. Selecting the FinancialTransactions.BalanceAndTurnovers table

In the **Tables** panel that is in the middle, select the **FinancialTransactionsBalanceAndTurnovers** table and click **Add all fields** . All fields of this table will be selected as query fields.

Hello, 1C

Query builder * Tables and fields Grouping	Condit	ions More Unions/Aliases Order Data	a compo	osition Features Query batch
🗐 🍓		ta 🙀 🖉 🗶 🐹 🎭 Tables	1	🔁 🥒 🗙 Fields
 ⊕ III Catalogs ⊕ Documents ⊕ Enums ⊕ AccumulationRegisters ⊕ FinancialTransactions ⊕ FinancialTransactions ⊕ FinancialTransactions ⊕ FinancialTransactions ⊕ FinancialTransactions ⊕ FinancialTransactions 		Financial Transactions Balance And Turnovers	∧<<	Financial Transactions Balance And Tu Financial Transactions Balance And Tu

Figure 7-8. Selecting data for query

The query is configured, click **OK**.

The wizard will generate the query text and automatically populate fields of the data composition schema.

Page 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	es: N	Main DataComp	posit	ionSc	hema							_	. 🗆 🗙
Data Sets Data	a set li	nks Calcula	ited fi	ields	Resources	Parame	ters	Te	mplat	es Nesteo	d schemas	S	ettings
tii- ×	Field	ls:									0 6	a 1	h x
⊖ Data Sets	-	Field	Path	h		Field	l restri	ction		Role	Presentati	o	Hierarc
🔟 Dat				Autotit	le	Fi	C	G	0		Ordering		Data se
						Attri	bute F	Restrict	tion		Expression	ns	Parame
						Fi	C	G	0				
	-	AmountClos	Amo	ountClo	singBalance					ClosingBal,			
			\Box	Amour	nt Ending balance	~	~	~	4	Amount			
	-	AmountExp	Amo	ountExp	pense								
			\Box	Amour	nt Expense	×	V	~	4				
	-	AmountOpe	Amo	ountOp	eningBalance					OpeningBal,			
				Amour	nt Starting balance	 ✓ 	\checkmark	V	~	Amount			
	-	AmountRec	Amo	ountRe	ceipt								
			_		nt Receipt	V	\checkmark	V	~				
	-	AmountTur		ountTu									
			-		nt Turnover	V	~	~	~				
	-	Event	Eve							Dimension			
		-	_	Event					U	-			
	-	Person	Pers			<u> </u>			U	Dimension			
			U	Persor	1		U	\cup	U				
	-				e								•
	Que	ry:									🔀 Que	ery B	uilder
		SELECT											
					ransactions								
					ransactions							_	
					ransactions ransactions								
					ransactions							-	
		Fina	anci	ialT	ransactions	Balan	ceAr	ndTu	rno	vers.Amou	ntReceip	pt,	
			anci	ialT	ransactions	Balan	ceAr	ndTu	rno	vers.Amou	ntExpens	3e	
		FROM		1	onRegister.	Finar	ai a'	Tree		ationa Pa	lancolor	ч т	
		ACCL	antu.	Taci	unkegister.	rinan	GT4	LT T d	1134	ccrons.ba	TanceAnd	u	
		0											•
📫 🔚	Auto	ofill (•										



Click the **Resources** tab. Here, using double-clicks select following fields:

- AmountClosingBalance
- AmountOpeningBalance
- AmountTurnover
- AmountReceipt
- AmountExpense

🕙 Report Expenses: MainDataC	omposi	tionSchema		_ 🗆 ×
Data Sets Data set links Cal	culated t	fields Resources Para	meters Templates N	Vested schemas Settings
Available fields	1	Field	Expression	Calculate by
- AmountClosingBalance		AmountClosingBalance	Sum(AmountClosingBala	
- AmountExpense	\gg	AmountOpeningBalance	Sum(AmountOpeningBal	
a AmountOpeningBalance		- AmountTumover	Sum(AmountTurnover)	
AmountReceipt		AmountReceipt	Sum(AmountReceipt)	
- AmountTurnover	<<	AmountExpense	Sum(AmountExpense)	
- Event				
- Person				

Figure 7-10. Data composition schema resources

Click the **Settings** tab. Here, in order to create the structure of the report, you will use another wizard. To do this, click **Open settings wizard** \boxtimes .

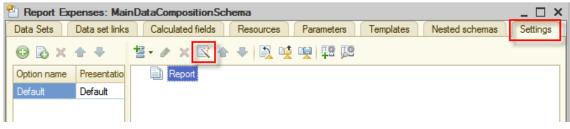


Figure 7-11. Opening Data Composition Settings Wizard

Data Composition Settings Wizard allows you to quickly create several simple variants of the report. For this tutorial purposes, **List** is a good choice. Therefore, do not change anything, and just click **Next**.

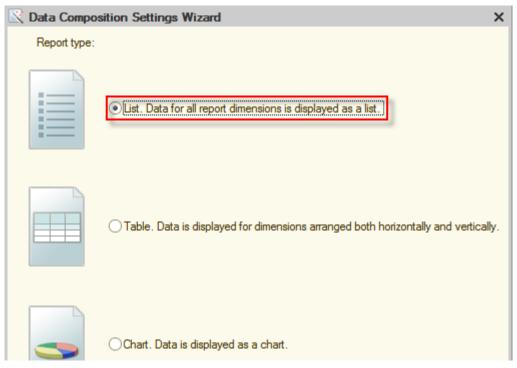


Figure 7-12. Creating a report in the form of list

On the next step, select fields that will be displayed in the report. Doubleclick the following fields in **Available fields** list:

- Person
- Event
- AmountTurnover
- AmountReceipt
- AmountExpense

Then click **Next**.

🔀 Data Composition Settings Wizard			×
Field Goods Warehouse Sum	report.		
Available fields		Selected fields	a
🕀 📼 Event		- Person	•
🕀 📼 Person	>>	👄 Event	
① AmountClosingBalance	<	Amount Tumover	
🕀 🥑 Amount Expense		Amount Receipt	
🕀 🥖 AmountReceipt	<<	🥖 AmountExpense	
① AmountOpeningBalance			
🕀 🤳 AmountTurnover			
🕀 📁 SystemFields			
🕀 🗀 DataParameters			

Figure 7-13. Selecting report fields

Now we need to select fields that will group the data in the report. Users would like to see reports on each acquaintance, and on each event associated with the acquaintance. To achieve this, the list of available fields double-click on both **Person** and **Event** fields to select them as grouping fields.

After that, click **OK**. You are now done with defining the structure of the report.

C Data Composition Settings Wizard				×
Group by Goods Warehouse 	. Group	ing fields are displaye	ed in the report.	
Available fields		Grouping fields	Grouping type	
🛞 📼 Event		– Person	Elements	-
🕀 📼 Person	>>	- Event	Elements	
① AmountClosingBalance				
🕀 🦸 AmountExpense				
🕀 🦸 AmountReceipt	<<			
④ J AmountOpeningBalance				
🟵 🤰 AmountTumover				

Figure 7-14. Selecting group fields of the report

The platform will now show the structure of the report.

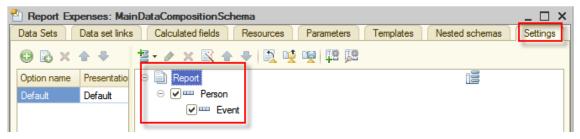


Figure 7-15. Completed structure of the report

It is almost done. Now, give a user an ability to arbitrarily set the report period. This will be useful when the register will contain large amount of records accumulated after a long period of using this application. This is easy to do as under the structure of the report, there are two parameters available, named **Begin of period** and **End of period**.

皆 Report Exp	enses: Mai	inDataCompositionSchema
Data Sets	Data set links	s Calculated fields Resources Par
0 🔂 🗙	* *	🗄 • 🖉 🗙 🖹 🛧 🗣 📓 💆 🖳
Option name	Presentatio	⊖ 📑 Report
Default	Default	
		🕑 🚥 Event
		Settings: Report Report
		Parameters Selected f Filter
		Display unavailable parameters (setting va
		Parameter
		Begin of period
		End of period
< c		

Figure 7-16. Report parameters

Right-click each of parameters and then select **Custom settings item** property.

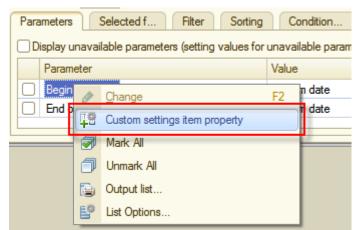


Figure 7-17. Opening Custom settings item property

In the opened window, select the **Include in custom settings** check box and click **OK**.

Item's custom settings	×
Include in custom settings Presentation	
Edit mode	Quick access
	OK Cancel Help

Figure 7-18. Include in custom settings

After repeating this procedure for both **Begin of period** and **End of period** parameters, continue configuring the report. To do this, close the data composition schema editor and return to **Expenses** report editor. Click the **Subsystems** tab and include this report in the **Finances** subsystem.

Report Expenses		_ 🗆 ×
Main Subsystems 	Subsystems that include the report:	
Functional options	Contacts	
Data	Events	
Forms	✓ Finances	
Commande		

Figure 7-19 Including Expenses report in the Finances subsystem

Start the application in the 1C:Enterprise mode and take a look at the report. Notice that this report is located in the **Finances** section. Also, notice that the platform automatically placed the report in the **Reports** menu group.

👝 🕞 Master cl (1C:Enterprise) 📄	🖶 🗟 🏤 🗟 🔳 🗃 м м	+ M- 🕡 🚽 🗕 🗆 🗙
Quick menu 🧼	Contacts 🥥 Events	Finances
Cash payment Cash receipt	Reports -	
	Expenses	

Figure 7-20. Opening Expenses report

After the report is opened, click **Create**.

(Select variant			More	•
egi	n of period: 🗌 🛛 / / 🗄 : : AN 👻 🗰 En	d of period:		:: AM -	₽
	Person	Amount	Amount	Amount	-
	Event	Turnover	Receipt	Expense	
-		128.00	150.00	22.00	-
Γ	Club	-12.00		12.00	
	College	-10.00		10.00	
	College financial assistance	150.00	150.00		
-	Georgia	-80.00		80.00	
		-50.00		50.00	
	College	-30.00		30.00	
-	Grandmother	100.00	100.00		
	Meeting relatives	100.00	100.00		
5	Sarah Wells	-15.00		15.00	
	Club	-15.00		15.00	
	Total	133.00	250.00	117.00	

Figure 7-21. The Expenses report

Having confirmed that the report works properly and displays records of the **Cash receipt** and the **Cash expense** documents according to dimensions that were defined for the register, return to the Designer mode.

Create one more simple report that can show the current balance of available funds. Name it **CashBalance** and include it in the **Finances** subsystem. Similarly, to the **Expenses** report, click **Open data composition schema** for the **Cash balance** report on the **Main** tab and create the data composition schema. Add the data set of the **Query** type on the **Data sets** tab.

ł	🔄 Re	port CashBalance: Ma	in DataCompo
ſ	Data	Sets Data set links	Calculated fie
	† 11-	×	
I	tı	Add data set - query	
ľ	10	Add data set - object	
	¢D	Add data set - union	

Figure 7-22. Adding the query for the Cash balance report

In the opened window, use **Query builder** as previously. However, for the new report, select the **FinancialTransactions.Balance** table.

Tables and fields Grouping Conditions More	Union	s/Aliases Order Data comp
II 🖏		🛅 🙀 🖉 🗙 🎉 糞
Database		Tables
⊕ III Catalogs	>	🕀 🧕 Financial Transaction >
⊕ 📃 Documents	>>	>>
⊕ {} Enums		<
Accumulation Registers		
Financial Transactions	<<	<<
⊕ Sinancial Transactions.Balance		
🕀 📧 Financial Transactions.BalanceAnd Turnovers		
🕀 🥑 Financial Transactions. Turnovers		

Figure 7-23. Selecting tables for the Cash balance report

Next, expand **FinancialTransactionsBalance** in the middle panel, click the **AmountBalance** field, and then double-click or click **Add selected** to select it.

Ę	🗒 Query builder				
	Tables and fields Grouping	Conditions More Unions/Aliases O	rder Data composition Features Query batch		
	💷 🍓	🏪 🙀 🖉 🗙 👹 🐃	🕒 🖉 🗙		
	Database	Tables	Fields		
	🕀 🎹 Catalogs	Financial Transactions Balance	FinancialTransactionsBalance.AmountBalance		
	Documents	→ t L Event	>>		
	🕀 {} Enums	C Person			
	🕀 🥃 Accumulation Registers	Amount Balance			
			«		

Figure 7-24. Selecting the AmountBalance field

Click **OK** to close **Query builder**. After this, again you will see the **Data Sets** tab of the data composition schema editor. No changes need to be made here. Click the **Resources** tab, then double-click the **AmountBalance** resource.

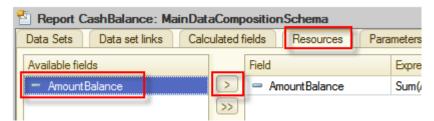


Figure 7-25. Selecting the AmountBalance resource

Open the **Settings** tab. Here, use the **Data Composition Settings Wizard** \bowtie once again, confirming the default type, List.

🖹 Data Compos	sition Settings Wizard X	
Report type:		
	List. Data for all report dimensions is displayed as a list.	
	Table. Data is displayed for dimensions arranged both horizontally and vertically.	
-	◯Chart. Data is displayed as a chart.	

Figure 7-26. Data Composition Settings Wizard

On the next page, double-click the **Period** standard field in the **DataParameters** group. This field will allow seeing the date, which the report is created for. Then add the **AmountBalance** field.

🖹 Data Composition Settings Wizard X			
Field Goods Warehouse Sum			
Available fields	Selected fields	<u>ه</u>	
🟵 🥖 AmountBalance	DataParameters.Period	•	
🕀 🗀 SystemFields	>>> 🚺 Amount Balance		
○ DataParameters □ Period	< <<		

Figure 7-27. Selecting report fields

After selecting fields, click **OK** because there is nothing else that this report is required to display. The next thing to do is to include the **Period** parameter to **Custom settings** as you did when configuring the **Expenses** report.

Parameters Sele	ected f Filter Sorting	Condition U	ser Fields Other sett	
Display unavailabl	le parameters (setting values for (unavailable parameters	is not recommended)	0
Parameter		Value	Date	1
Period	hange	F2 date		đ
「草着 ロ	ustom settings item property			ሞ
1 M	Nark All	-		
<u></u>	Jnmark All			
P 0)utput list			
E ^o Li	ist Options			

Figure 7-28. Including Period to Custom settings

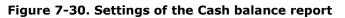
Set the default value of the **Period** parameter to **Beginning of this day** so that the report will automatically display the information for the current day.

Parameters Selected f Filter Sorting	Condition	User Fields	Other set	t
Display unavailable parameters (setting values fo	unavailable paramet	ers is not recom	mended)	٨
Parameter	Value	Date		1
Period	Custom date	. ×		A
	Custom date			
	Beginning of this da	ву		平 帶
	Beginning of this w	eek	e .	
	Beginning of this te	n days		
	Beginning of this m	onth		
	Beginning of this qu	uarter		
	Beginning of this ha	alf year		
	Beginning of this ye	ear		
	Beginning of the ye	esterday		
	Beginning of last w	eek	v	

Figure 7-30. Selecting Beginning of this day as the default value of the Period parameter

Part CashBalance:	MainDataCompositionSchema		_ 🗆 ×	
Data Sets Data set links	Calculated fields Resources Parame	ters Templates N	ested schemas Settings	
\varTheta 🗟 × 🛧 🗸	철· 🖉 🗙 🖹 🍝 🕂 🔯 🖳 🖳	1 5 8		
Option name Presentatio	⊖ 📄 Report		12	
Default Default	Contraction of the second s			
Settings: Report Parameters Selected f Filter Sorting Condition User Fields Other sett				
	Display unavailable parameters (setting values for unavailable parameters is not recommended)			
	Parameter	Value	Date 🦪	
< 0 ►	Period	Beginning of this day	3/25/2014 12:00:0	

After that changes, the report settings will look as follows:



Now start the application in the 1C:Enterprise mode and check the report.

Rapid application development tutorial

📧 😠 Master clas (1C:Enterprise) 🚪	🖶 🗟 🚖 🖻 🔳 🗃 M	M+ M- (i) - 🗆 ×
Quick menu 🧼	Contacts 🥥 Events	Finances
Cash payment Cash receipt	Reports -	
	Cash balance	
	Expenses	Ţ.

Figure 7-31. Opening the Cash balance report in the 1C:Enterprise mode

Information presented in the report is extremely simple but useful. Now the user can always check how much money is available now.

+ +	🖙 🖓 Cash	balance	>
Creat	e Select varia	ant	More 💌
Period:	Beginning of this	s day	· ∎
Data parameters: P		Period: 3/25	/2014 12:00:00 AM
		Balance	
3/2	/2014 12:00:00 AM	133.00	
To	al	133.00	
			· · ·

Figure 7-32. The data of the Cash balance report

Excellent! Now, create another report. After this report will be finished the CRM application will be complete.

Return to the Designer mode, create a new report, and name it **DailyChart**, then include it in the **Finances** subsystem, and finally click **Open data composition schema**. Add the **Data set** of the **Query** type, and then open **Query builder**.

In the left panel select the **FinancialTransactions.BalanceAndTurnovers** table. Up to this point, the process has no difference from what you did during configuring the **Expenses** report.

Seeing that the report is named **Daily chart** you might have guessed that a new report will display not numeric data but a chart. And now the trick. To present the data from the register in the suitable for this report format, select **FinancialTransactionsBalanceAndTurnovers** table in the middle panel of **Query builder** and click **Virtual table parameters**

🗓 Query builder *	
Tables and fields Grouping (Conditions More Unions/Aliases Order Data
💷 🎕	🛅 🛃 🥒 🗙 🔀 🌄
Database	Tables
⊕ IIII Catalogs	GenerationsBalanceAndTumovers
① Documents	
⊕ {} Enums	
Accumulation Registers	
③ Financial Transactions	< <u><</u>
🕀 🥃 Financial Transactions.B	3
⊕ 🥃 Financial Transactions.B	
🕀 🧾 FinancialTransactions.T	d

Figure 7-33. Opening Virtual table parameters

In the opened window, in the the **Periodicity** field, select **Day** and then click **OK** to save and close the window.

Virtual Table Settings	×
BeginOfPeriod EndOfPeriod	· · · · · · · · · · · · · · · · · · ·
Periodicity	Day
ComplementMethod	▼ X
Condition	
	OK Cancel Help

Figure 7-34. Adjusting Virtual table parameters

Then, in the middle panel, select the **Period** and the **AmountClosingBalance** fields from the **FinancialTransactionsBalanceAndTurnovers** table.

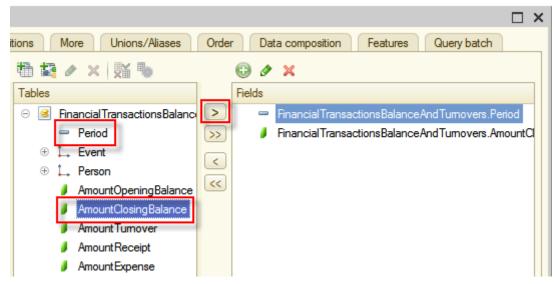
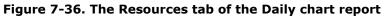


Figure 7-35. The Period and the AmountClosingBalance fields

Click **OK** to close **Query builder**. Then, click the **Resources** tab and select the **AmountClosingBalance** field as a resource.

Part DailyChart: MainDataCompositionSchema					
Data Sets Data set links Calco	ulated fields Resources I	Parameters Templates N			
Available fields	Field	Expression			
 AmountClosingBalance 	AmountClosingBalance	e Sum(AmountClosingBala			
🗢 Period	>>				



Click the **Settings** tab and open **Data Composition Settings Wizard**. This time, click **Chart. Data is displayed as a chart**, and then click **Next**.

🖹 Data Compos	X Data Composition Settings Wizard X			
Report type:				
	◯ List. Data for all report dimensions is displayed as a list.			
	Table. Data is displayed for dimensions arranged both horizontally and vertical	ly.		
-	Chart. Data is displayed as a chart.			

Figure 7-37. The Chart report type

After adding **Period** and **AmountClosingBalance** to selected fields, click **Next**.

🔀 Data Composition Settings Wizard		×
Field Image: Goods Image: Warehouse Select fields to display in the report. Image: Sum		
Available fields	Selected fields	
 Period AmountClosingBalance SystemFields DataParameters 	Period AmountClosingBalance <	
< Back Next > OK Cancel Help		

Figure 7-38. Adding chart fields

On the next page, add the **Period** field to the list in the **Points** panel on the right side. Click **Next**.

🖹 Data Composition Settings Wizard				×
Charts Select fields to group charts, in the report.	series and c	chart points by. G	àrouping fields are d	lisplayed
Available fields	Ser	ies:		
- Period	Gr	ouping fields	Grouping type	
AmountClosingBalance				-
	>>			
	<			
	e Poi	nts:		
	Gr	ouping fields	Grouping type	
	> Gr	ouping fields	Addition type	
	>>		Elements	
		Period	No addition	
	<< L			
		arts:		
		ouping fields	Grouping type	•
	>			+
	<			
	<< L			
< Back	Next >	ОК	Cancel	Help

Figure 7-39. Configuring Points of the chart

On the next page, add **Period** to order fields on the right panel. Click **Next**.

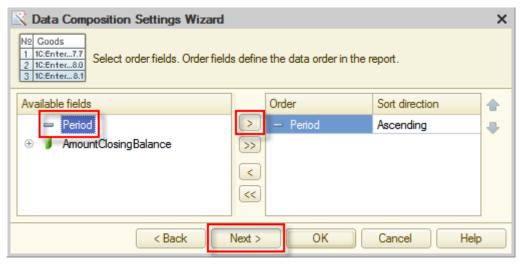


Figure 7-40. Data order fields of the chart

On the next page, in the **Chart type** field select **Line**. At this point, the design of chart complete, and you can click **OK**.

🖹 Data Composit	ion Settings Wizard	×
Sele	ct chart type.	
Chart type:	Column 3D	
	N Line	
22	Step	e
18	Stacked graph	
16	- Marea	
12	Stacked Area	
10	Normalized area graph	
Ĝ 🔶	L. Column	
4	Stacked Column	
2	Normalized bar chart	
<point 1=""></point>	Column 3D	
	< Back Next > OK Cancel Help	

Figure 7-41. Chart types

In the same way as you did for previous reports, include the **Begin of period** and the **End of period** parameters in **Custom settings** using **Custom settings item property**.

Para	ameters	Sele	ected	f	Filter		Sorting		Condition	
	Display unavailable parameters (setting values for unavailable parame									
	Paramet	ter						Valu	ie	
	Begin o	f period						Curd	tom data	
Ō	End of		Ø	<u>C</u> ha	nge				F2	
			轚	Cust	tom settin	gs	item prop	erty	1	
			a	Mad					<i>a</i> .	

Figure 7-42. Adding properties to custom settings

Now you can see what you have created. Start the application in the 1C:Enterprise mode, open the **Daily chart** report, and then click **Create**.

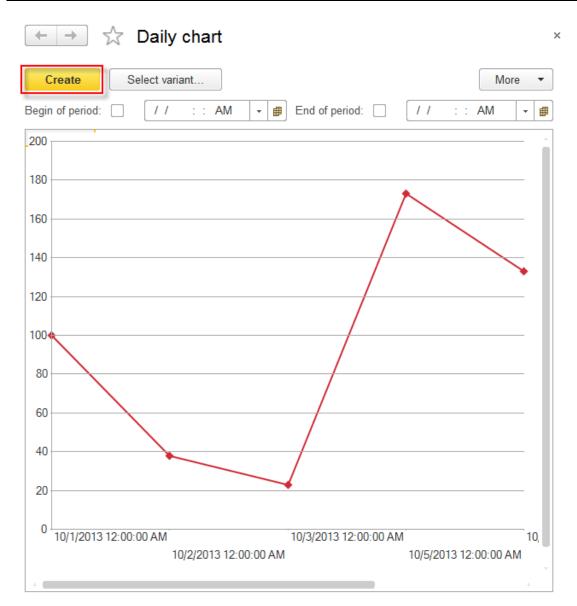


Figure 7-43. The Daily chart report

From this chart users can clearly see the changes in balance of money, made by **Cash receipts** and **Cash expenses**.

Improving the interface

So far, you have created a CRM application that is able to perform all actions, mentioned in project requirements. Except for one objective. At this point, the application is simple but not user-friendly.

Look at the main application window.

10 🕤 M	aster (1C:Enterpri	ise) 🔚			31 M M+	M- (i)	+ _ 🗆 ×
2	Quick menu	\bigcirc	Contacts	\bigcirc	Events	\bigcirc	Finances
*							
IJ							
0,							
📑 Curre	nt calls: 0 Accumul	ated calls: {	5				

Figure 8-1. Start page

Start page is empty. All sections look the same and contain only the minimum that was automatically generated by the platform. In next chapters, you are going to improve this workspace.

Improving subsystems

As the first step, you will set icons for subsystems. This is not a difficult task. In the Designer mode open the subsystem editor and select the desired image in the **Picture** field for each subsystem. Start with the **Contacts** subsystem.

Hello, 1C

🗞 Subsystem Contact	5	_ 🗆 ×
Main	Name:	Contacts
Functional options		
Content	Synonym:	Contacts
Other	Comment:	
	Include in comm Command	
	Explanation:	
	Picture:	×

Figure 8-2. Selecting an icon for the subsystem

There are no pictures in the applied solution yet, so the list is blank. To have them available, import images from files. In the opened window, click **Add**. You can find images in the current distribution, for this, run **setup.exe**, click **Custom setup** and then click **Hello, 1C additional files**. In opened folder, you can find images. Copy them from that folder to your computer. And then add to the applied solution.

Select picture	×
From library	
From configuration Standard	
Add Edit Delete	
Clear Save OK Cancel	5

Figure 8-3. Adding an image

Click Select from file.

Common pictu	re CommonPicture1	×
Name	Common Picture 1	
Synonym		
Comment		
Subsystems		
		Select from file
		Clear
		Set transparent background
		Remove transparency
		Edit
		Save to file
Image type:		

Figure 8-4. Selecting a file

For the **Contacts** subsystem select **ContactsSubsystem.png**, and then click **Open**.

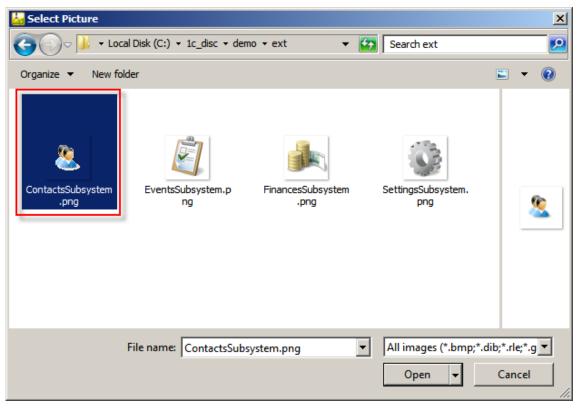


Figure 8-5. Selecting an image

Give the name to the picture, for example, **ContactsSubsystem**.

Common pictu	ire ContactsSubsyste	em X
Name	ContactsSubsystem	
Synonym	Contacts subsystem	
Comment		
Subsystems		
		Select from file Clear
<u>®</u>		Set transparent background Remove transparency Edit
		Save to file
Image type:	png	-

Figure 8-6. Naming the common picture

Now you can close the window with the picture. In the **Select picture** window select the picture that you have just imported, and then click **OK**.

Select picture	×
From library	
From configuration Standard	8 48 x 48
Add Edit Delete	
Clear Save	OK Cancel

Figure 8-7. Selecting a picture

Now the **Contacts** subsystem has a unique appearance. Repeating the above steps, select images for the remaining subsystems:

- EventsSubsystem.png for the Events subsystem;
- FinancesSubsystem.png for the Finances subsystem.

As a result, pictures for subsystems will look as follows:

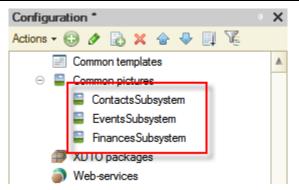


Figure 8-8. Common pictures for subsystems

Start the appliation in the 1C:Enterprise mode and look what you have now.

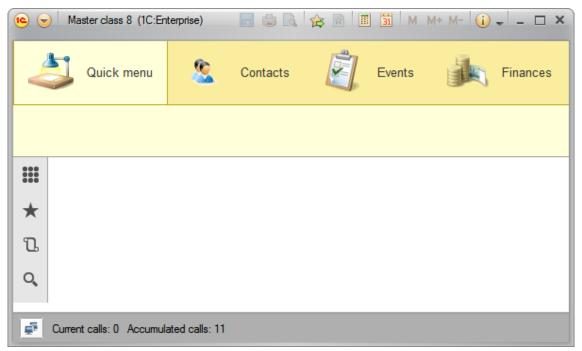


Figure 8-9. Subsystem icons in the 1C:Enterprise mode

Adjusting the subsystem content

Sections panel with newly added icons looks more user-friendly. Now, look at what is inside of sections. Their appearance is still quite blind. In **Contacts** there are only three links to catalogs, and in **Events** only two. The **Finances** section looks a bit more pleasant. Here also included reports besides documents.

Hello, 1C

📧 🕤 M (1C:Enterprise, training version)		r 📰	M M+ M- 🥡	- 🗆 ×
Quick menu 🧕	Contacts	Ever	its	Finances
Cash payment Cash receipt	Reports 👻			

Figure 8-10. Content of sections

Add some diversity to the content of sections. Open the list of available commands for subsystems, and activate those commands that may be useful to users. Start with the **Contacts** subsystem. Open the subsystem editor, and click **Command interface**.

🟂 Subsystem Contact	S	_ 🗆 ×
Main Functional options Content Other	Name: Synonym: Comment:	Contacts Contacts
	-	Imand interface:

Figure 8-11. Opening Command interface

Place the often used **People** catalog in the **Navigation panel.Important** command interface group either by dragging it there with a mouse or clicking **Move a command** $\stackrel{\leftarrow}{\leftarrow}$. Add the ability to create new records for all three catalogs in the **Actions panel.Create** command interface group by selecting **Visibility** check boxes. Then change the order of the **People: create** command by clicking **Move up** $\stackrel{\leftarrow}{\bullet}$.

As the result, you will have following command interface settings:

Command interface		:
🚰 🍙 🗣 🛛 🚱 👘 Filter by roles	s: <not set=""></not>	•
Command	Visibility	Visibility by roles
Navigation panel.lmportant		,
E People	\checkmark	
 Navigation panel.Normal (manual order) 		
Contact types	\checkmark	
People relation types	✓	
 Navigation panel.See also 		
Actions panel.Create (manual order)		
m People: create		
Contact types: create		
People relation types: create		
Actions panel.Reports	Terrando de la constante de la	
🐡 Actions panel.Tools		

Figure 8-12. Adjusting Command interface of the Contacts subsystem Make the similar changes to the **Events** subsystem.

😡 Command interface		×
🧬 🍲 🌷 🚱 😺 🐌 Filter by roles: <	<not set=""></not>	▼
Command	Visibility	Visibility by roles
 Navigation panel.Important (manual order) Events Navigation panel.Normal Event categories Navigation panel.See also Actions panel.Create Event categories: create Events: create Actions panel.Reports 	 ✓ ✓ ✓ ✓ 	
Actions panel. Tools		

Figure 8-13. Adjusting Command interface of the Events subsystem

Also, check **Finances** subsystem. Here set **Financial transactions** command as an **Important** command.

Figure 8-14. Adjusting Command interface of the Finances subsystem

Filter by roles:

Visibility

~

Image: A start
 Image: A start</l

-

•

×

¥

Visibility by roles

Start the application in the 1C:Enterprise mode and take a look at the interface now. Well, links in the **Contacts** section look more orderly. The **People** catalog can be easily found now, which is convenient, considering that users will work with this catalog more often than with others. In addition, there is a new command group for creating new catalog items in this section.

10 🕤	M (1C:Enterprise, training vers	ion) 📕 🏮 🖪	🚖 🖻 🔳	31 M M+ M- 🥡	×	
4	Quick menu	Contacts	<u></u>	Events	Finances	
Peopl	e Contact types Peo	ople relation types				
	Contact types					
+ Ⴂ	Create Find	Cancel se		eople relation types	Aoro T	
Q,	Description ↓	Code	Gender	Relation type	Con	
	🗕 Georgia	00000002	Female	GirlPerson	sopl	
	- Grandmother	00000001	Female	Family	the	
	 John Sheldon 	00000003	Male	Person	neig	

Figure 8-15. The Contacts section

Look at the interface of the **Events** section, which now looks more complete.

Hello, 1C

Command interface

Command

Navigation panel.Important (manual order)

Navigation panel.Normal (manual order)

Financial transactions

Cash payment

Navigation panel.See also

Cash payment: create
 Cash receipt: create
 Actions panel.Reports (manual order)

Cash receipt

Actions panel.Create

🛄 Expenses

Daily chart

Actions panel. Tools

😐 Cash balance

🌮 🛧 👆 🚱

اچ ھ	M (1C:Enterprise, training version) 📄 🍙 🔍 🎓 🖬 🔳 M M+ I	M- 🕡 🚽 🗕 🗆 🗙
4	Quick menu S Contacts	Finances
Event		
	Event categories	×
★ Ⴂ	Create Find Cancel search	More 🔻
Q,	Description 4	Code
	– Cafe with Georgia	0000000
	- Club	0000000
	- College	0000000

Rapid application development tutorial

Figure 8-16. The Events section

Now, look at the **Finances** section. This time all objects that the user might need are easily accessible. For example, you no longer need to open **All functions** in **Main menu** each time when you want to view records of the **Financial transactions** register.

•	M (1C:Enterprise, trai	ning version)		31 M M+ M- 🥡 🗸	- 🗆 ×		
4	Quick menu	<u>&</u> c	Contacts 🗾 E	Events	inances		
Finan	cial transactions	Cash paym	ent Cash receipt		orts 🔻		
Ⅲ ★	Cash receipt						
l D	Find	Cancel search		More	•		
O,	Period	Ļ	Event	Person	*		
	- 10/3/2013	12:00:00	Club	Sarah Wells			
	+ 10/5/2013	4:02:25	College financial as				
	- 10/7/2013	4:00:49	College	Georgia			

Figure 8-17. The Finances section

Start page

You have managed the lack of navigation links in subsystems. But **Start page**, which is the first page that the user sees when open an application in the 1C:Enterprise mode, is still empty.

1C) М.	(1C:Enterprise, tra	aining version)		🚖 🖹 🔳	31 M I	M+ M- 🕕	×
4	5	Quick menu	8	Contacts	Ê,	Events		Finances
*								
IJ								
Q,								

Figure 8-18. The empty Start page

You definitely want to make things right. To do this, return to the Designer mode, open **Properties** of the **Configuration** root node, and click on **Open** link of **Start page working area** field.

Rapid application development tutorial

Configuration X	Properties	0	×
Actions 🗸 💮 🖉 🗟 🗙 🛧 🖶	21 📑 🍯 🗙 🗸		
lello1C	✓General:		
🕀 🐉 Common	Name Hello1C		
🔡 Constants	Synonym Hello, 1C		1
⊕ IIII Catalogs	Comment		1
Documents			
Document journals	Default run mode	Managed application	e
⊕ {} Enumerations	Use purposes	Personal computer	
Eeports	Script variant	English 💌	
Data processors	Default roles	X	
Charts of characteristic types	Managed application mo	Open	
Tr Charts of accounts	Session Module	<u>Open</u>	
Charts of calculation types	External Connection Mo	Open	
Information registers	Additional full-text search	· X	
Accumulation registers			
Accounting registers	Common settings storag		
Calculation registers	Report settings storage	×	Į
Business processes	Report variant settings s	×	
Tasks	Form data settings stora	×	
🍢 External data sources	Dynamic list user setting	×	
	Command interface	Open	
	Start page working area		
	Main section command	-	
	Main section picture	<u>Open</u>	T

Figure 8-19. Opening Start page working area

There are a few things that you can change here for now, which is why simply select **Two columns, different width (2:1)** as the value of the **Starting page template** field.

Starting page work a	area						_ 🗆 ×
Starting page template:	Two	columns, d	lifferent	width (2:1)	1	-	
Left column:	-		Ri	ght column:			
😌 🗟 🗙 🌚 😓				🖻 🖪 🗙	☆ 🕹		
Form	Height	/isibility	F	om		Height	Visibility
			>				
			<				

Figure 8-20. Starting page template

The feature of **Start page** is that you can place there only those forms that are explicitly created in the **Configuration** object tree. Thus, the first thing you need to do is to create forms using **Form Wizard**. Those forms are the list form of the **Financial transactions** accumulation register, the list form of the **People** catalog, and the report form of the **Cash balance** report.

It is easy to do. In the **Configuration** object tree, select the configuration object, for which you are going to add a form. Expand this branch and right-click **Forms** node inside of it, then click **Add** 😳 (Ins).

Start with the register.

Θ	3	Ac	cum	ulation registers		
	Θ	8	Fina	ancialTransactions	1	
		\oplus	1.	Dimensions		
		\oplus	1	Resources		
			-	Attributes		
				Form Add	lee	1
			O	Commana	Ins	
			<u>نې</u>	Templates		

Figure 8-21. Adding the list form for the Financial transactions accumulation register

The **Accumulation Register Form Wizard** window will open. In this case, the default option **Accumulation register list form** is the required one to create the desired list form, so simply click **Finish**.

Select form type Accumulation Accumulation Generic form	register list form	
Set form as de Name: Synonym: Comment:	List Fom	
< Back	Next > Finish Cancel Help	

Figure 8-22. Adjusting the list form of the accumulation register

Find the **People** catalog in the **Configuration** object tree, and add a list form for this catalog as well.

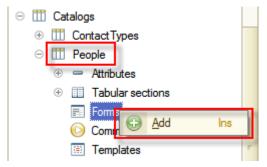


Figure 8-23. Adding a list form for the People catalog

	Wizard
Select form ty	/ре:
OCatalog ite	em form
Catalog fol	lder form
Catalog list	t form
Catalog ch	noice form
Catalog fol	lder choice form
Generic fo	m
Default ite	s default m/folder form ListForm
Set form as Default iter Name: Synonym:	m/folder form
Default ite Name:	m/folder form
Default iter Name: Synonym:	m/folder form
Default iter Name: Synonym:	m/folder form
Default iter Name: Synonym:	m/folder form
Default iter Name: Synonym: Comment:	m/folder form

Click Catalog list form in the Select form type panel, and click Finish.

Figure 8-24. Adjusting a list form of a catalog

Now create the report form for the **Cash balance** report.

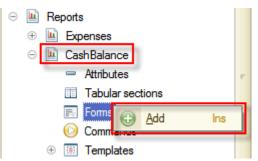


Figure 8-25. Adding a report form for the Cash balance report Click **Report form** in the **Select form type** panel and click **Finish**.

Report form wiza	rd	×
Select form type Report form Report setting Report variar Generic form	gs form	
Set form as de		
Name:	ReportForm	
Synonym:	Report form	
Comment:		
	lack Next > Finish Cancel Help	

Figure 8-26. Adjusting a report form of a report

Return to the **Starting page work area** editor. This time you will be able to customize its appearance by adding to it newly created forms. Let it have an appearance with the **Financial transactions** register list form and the **Cash balance** report form on the left side, and the **People** catalog list form on the right side.

For implementing this, click **Add** ⁽²⁾ above the **Left column** list and select the list form of the **Financial transactions** accumulation register, and then click **OK**.

Starting page work area	I.			_ 🗆 ×	
Starting page template:	Two columns, differe	ent width (2:1)	-		
Left column:		Right column:			
🕀 🔂 🗙 🌚 🐣		🕀 🗟 🗙 🌚 🐣			
Form Heig	ht Visibility	Form	Height	Visibility	
	10 🗹 <u>Identical</u> >)			
Selec	t managed form			×	
•	111 Catalogs			Ж	
	Reports		Car	ncel	
Θ	Accumulation regis	sters			
	😑 🧾 FinancialTrans	actions			
⊖ 📰 Forms					
	List For	m			

Figure 8-27. Adding the list form of the Financial transactions accumulation register to Start page

In the same way, add the report form of the **Cash balance** report.

Starting page work	area			_ 🗆 ×
Starting page template:	Two columns, di	fferent width (2:1)	-	
Left column:		Right column:		
🕀 🔂 🗙 🌚 🐣		🔂 🗟 🗙 🌚 🗣		
Form	Height Visibility	Form	Height	Visibility
Accumulation Register	10 ⊘ Identical	>		
•	Select managed form	i i i i i i i i i i i i i i i i i i i		×
	⊕ III Catalogs			Ж
	□ I Reports		Car	ncel
	⊖ 🔟 CashBalan	ce		
	 □ Forms □ Forms □ Re ⊕ S Accumulation r 	egisters		

Figure 8-28. Adding the Cash balance report form to Start page

Repeat the same steps for **Right column** of **Start page**. Here, add the list form of the **People** catalog.

Starting page work	area	_ 🗆 ×
Starting page template:	Two columns, different width (2:1)	-
Left column: 🕞 🔂 🗙 🏠 😓	Right column:	
Form		Height Visibility
Accumulation Register	10 ⊘ <u>Identical</u> >	10 v <u>Identica</u>
Report.CashBalance	Select managed form	×
	 Catalogs People Forms List Form Reports Accumulation registers 	OK Cancel

Figure 8-29. Adding the list form to the Start page

Start the application in the 1C:Enterprise mode and look at **Start page**.

Hello, 1C

👝 😔 🛛 Master class 8 (1C:Enterprise, training version)	■ 🖨 🗟 🎓 🖉 📓 M M+M- 🛈 + – 🗆 :
Quick menu Scontacts 🗐 Events 🙀 Finances	
Image Image Image Financial transactions Image Find Cancel search Period Image Image Image Image Image Find Image Image Image Image	People More Create Find More Person Description Sarah Wells Georgia
+ 10/5/2013 4:02:25 Cash receipt 00000 1 College financial as - 10/7/2013 4:00:49 Cash payment 000 1 College - 10/7/2013 4:00:49 Cash payment 000 2 College	Georgia = Grandmother Georgia = John Sheldon Sarah Wells
Cash balance Create Select variant Period: Beginning of this day	More
Data parameters: Period: 3/28/2014 12:00:00 AM Parameters.Period Amount Balance 3/28/2014 12:00:00 AM 133.00 Total 133.00 Total 133.00 Total 133.00	

Figure 8-30. Start page in the 1C:Enterprise mode

Now **Start page** looks more useful. From here users can easily check financial transactions, manage the **People** catalog, and see how much cash they have available at a moment.

Command interface of Main section

At the current moment, only three objects are available on **Start page**. What if users would like to have more objects available? Is it necessary to create object forms for all of them and then manually place each one of them on the **Start page**? No, it is not necessary. If you place many forms on **Start page**, the interface will be overloaded and thus not user friendly. To make users able to access more objects from **Start page** without making the interface of it overloaded, you can adjust **Command interface** of **Main section** as you did with **Command interface** of subsystems.

In the Designer mode open **Properties** of root node in the **Configuration** object tree, and click **Open** link of **Main section command interface** property.

Rapid application development tutorial

Configuration X	Properties	0	×
Actions - 😳 🖉 🔝 🗙 🛧 🕂	91 E T × ~		
\varTheta Hello 1C	▼General:		
🕀 👶 Common	Name Hello1C		
🔡 Constants	Synonym Hello, 1C		
① III Catalogs	Comment		
Documents			/
Document journals	Default run mode	Managed application	e
⊕ {} Enumerations	Use purposes	Personal computer	
E Reports E E E	Script variant	English 💌	
Data processors	Default roles	X	
Charts of characteristic types	Managed application mc		
Tr Charts of accounts	Session Module	<u>Open</u>	
Charts of calculation types	External Connection Mor	<u>Open</u>	
Information registers	Additional full-text search	×	
① Accumulation registers			
Accounting registers	Common settings storage	×	
 Calculation registers 	Report settings storage	×	
	Report variant settings s	×	
Tasks	Form data settings storag	×	
🎼 External data sources	Dynamic list user setting:	×	
		<u>Open</u>	
	Start page working area		
	Main section command i		_
	Main section picture	<u>Open</u>	T

Figure 8-31. Opening Command interface of Main section

In the opened window select configuration objects and place them in **Command interface** of **Main section** using button. Afterwards arrange objects in **Command interface** by dragging them as it is shown below:

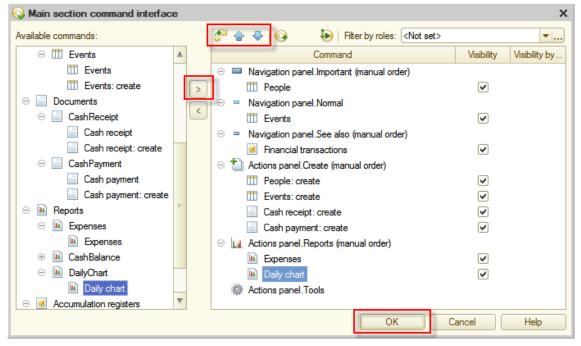


Figure 8-32. Adjusting Command interface of Main section

See how **Start page** looks now. Start the application in the 1C:Enterprise mode.

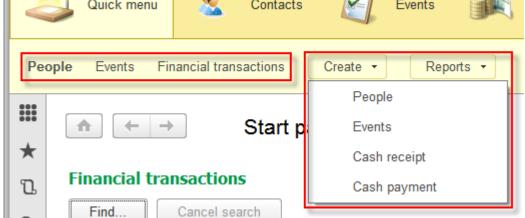


Figure 8-33. Adjusted Command interface of Start page in the 1C:Enterprise mode

Excellent! Now everything is at fingertips of users. By the way, using window separators, users can adjust the interface to increase more often used items, which is very convenient.

payment vov	1	College	Georgia			John She
payment 000	2	College		-	-	Sarah We
			More	•		

Figure 8-34. Separators

Cash balan	ce		
Create	Select varia	nt	More 🔻
Period: 🗹	Beginning of this	day	- ∎
Data pa	arameters: F	Period: 3/31/	2014 12:00
Parame	eters.Period	Amount Balance	
3/31/2014	4 12:00:00 AM	133.00	
Total		133.00	-
<			E.

Figure 8-35. The report form with adjusted size

Managed forms

If you look more precisely at register and catalog lists on **Start page**, you will see that the information is not very conveniently displayed. In the register list, there is odd information, but on contrary, in the catalog list it is not enough.

				People	
		More -		Create Find	More 🔻
Line num	Event	Person		Description 4	Code
1	Club	Sarah Wells		😑 Georgia	00000
1	College financial as			- Grandmother	00000
1	College	Georgia		 John Sheldon 	00000
2	College			 Sarah Wells 	00000
	1 1 1	1 Club 1 College financial as 1 College	Line num Event Person 1 1 Club Sarah Wells 1 College financial as 1 College Georgia	Line num Event Person 1 Club Sarah Wells 1 College financial as 1 College	More Create Find Line num Event Person Image: Description Image: Description 1 Club Sarah Wells Image: Description Image: Description 1 College financial as Image: Description Image: Description 1 College Georgia Image: Description 1 College Georgia Image: Description

Figure 8-36. Start page

To amend the situation, return to the Designer mode. Start with the register and open the recently created list form of the **Financial transactions** accumulation register.

Until now, all forms that you saw were generated automatically, even those that were created with one click to be placed on **Start page**.

There is no need to manually draw forms in 1C:Enterprise 8. The developer only needs to configure the form composition in a hierarchical tree inside the top panel of the form editor, and the appearance of the resulting form is displayed as a preview inside the bottom panel of the form editor.

The platform automatically calculates positions and sizes of items on the form.

E Accumulation register Financial Tra	ansactions: ListForm	
🕒 🖉 🗙 🚖 🖶 🚍 - 📪		75 📧 🖉 🗙
Form		Attribute
Command bar		
List SettingsComposerUserSettings		🕀 🎹 List
⊖ III List		
Command bar		
- Period		
- Recorder		
- LineNumber		
- Event		
- Person		
- Amount		
Elements Command interface		Attributes
Grind Cancel search	1	
Period	Recorder	Line num

Figure 8-37. Managed form editor

Now, you will configure this list form. Remove odd the **Recorder** and the **LineNumber** attributes. Select the desired attribute in the list, and click **Delete current item** ×(Del).

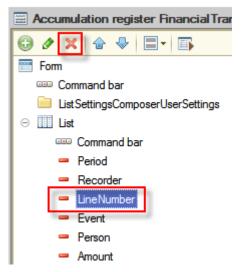


Figure 8-38. Deleting attributes of the managed form

After deletion of odd form items, the platform will redraw the form and display the preview as a user will see it in the 1C:Enterprise mode. The current form will look as follows:

Accumulation register Financial Tr	ansactions: ListForm	
😳 🖉 🗶 🍙 😓 🔲 🖬	· · · · · · · · · · · · · · · · · · ·	70 🐻 🖋 🗙
📰 Form	A	Attribute
Command bar		
ListSettingsComposerUserSettings		III List
⊖ III List		
Command bar		
- Period		
😑 Event		
- Person		
- Amount		
Elements Command interface		Attributes
Grind Cancel search	h	More 🔻
Period	Event	Person
*		

Figure 8-39. Configuring a managed form

In addition to deletion of odd form items, also adjust presentation of remaining form items.

In the 1C:Enterprise 8 platform, many properties that affect the data presentation in the interface are placed directly in properties of attributes of configuration objects. It is enough to describe attribute presentation settings in the object metadata itself. Then, the platform will automatically apply these properties when display attributes in all forms where they are placed.

Of course, if there is a need to change a presentation for a particular form, then those changes can be done in properties of the form item.

To begin, change the width to **10** for all form items in **List** table of list form of **Financial transactions** accumulation register.

Location:			
HorizontalAlign	Auto		Ŧ
HeaderHorizon	Left		Ŧ
FooterHorizonti	Auto		Ŧ
VerticalAlign	Auto		•
Width		10	¢
Height		0	\$

Figure 8-40. Adjusting the width of the managed form item

For the **Amount** form item select **Right** as a value of **HorizontalAlign** property so amounts could be observed easier.

Location:		
HorizontalAlign	Right	•
HeaderHorizon	Left	-
FooterHorizonti	Auto	•
VerticalAlign	Auto	•

Figure 8-41. Adjusting alignment of the form item

Then give a more user-friendly name to the **Period** form item let it be **Date**.

General:		
Name	Period	
Title	Date	

Figure 8-42. Adjusting a title to a form item

Now, start the application in 1C:Enterprise mode and look at the list form of the accumulation register.

Financial transactions

Find Cance	search		More	•
Date ↓	Event	Person	Amount	*
- 10/3/2013 12:	Club	Sarah Wells	15.00	
+ 10/5/2013 4:0	College financial assi		150.00	
- 10/7/2013 4:0	College	Georgia	30.00	
- 10/7/2013 4:0	College		10.00	

Figure 8-43. The list form of the Financial transactions accumulation register

You can verify on your own whether this form looks the same by opening it in the **Finances** section or by opening **Main menu** \bigcirc , then clicking **All**

functions... and opening this form in the form list. The form will look the same. Settings of the managed form are applied to all places of the applied solution where this form is used.

Standard and ordinary attributes

Continue to the **People** catalog. The **People** list on **Start page** currently contains less information. Besides, the **Description** word does not clearly present what is contained in the corresponding column. In addition, the list includes the **Code** column that seems not very useful to users.

As a start, you will change **Synonym** of the **Description** attribute to **Full name**. One way to do this is to open a list form and in **Properties** of the **Description** form item enter a synonym, as you did for the **Period** form item in the list form of the **Financial transactions** accumulation register.

However, this time, a different procedure is more suitable. There are several forms in the **People** catalog now, the list form and the catalog item form. The list form was created in the Designer mode, and the catalog item form is generated automatically by the platform. If you will change **Synonym** in the list form only, the catalog item form will not be changed. Therefore, it is better to change **Synonym** to **Full name** in one place, which is **Properties** of **Standard attribute**. To do this, find the **People** catalog in the **Configuration** object tree, right-click on it and then click **Standard attributes**.

⊝ 🎹 Catalogs	
① ① ContactType	es
🕀 🎹 People	
🕀 🎹 People 🌇	Open Object Module
🕀 🎹 Event 🐼	Open manager module
🕀 🎹 Event: 🔣	Open Predefined Data
Document	Open Default Object Form
Document	Open Default Group Form
⊕ {} Enumerati	
E E	Open Default List Form
Data proce	Open Default Choice Form
Charts of c	Open Default Group Choice Form
Tr Charts of a	0
🗢 Charts of c	Open •
Information	Wizards 🕨
🕀 🧾 Accumulat	Add
Calculation	Change F2
	<u>.</u>
👗 Business p 💦	Clone F9
📓 Tasks 🗙	Delete Del
🎼 External di	Move Up Ctrl+Shift+Up
-	Move Down Ctrl+Shift+Down
Ū.	Sort
-	Standard attributes
-	Characteristics

Figure 8-44. Opening Standard attributes

In the opened window find the **Description** attribute, and then in **Properties** specify **Full name** in the **Synonym** property.

🗧 C: Standard attributes 🔔 🔲 🗙	: Standard attributes - Description 🔹 X
Standard attributes:	21 🖼 🖉 🗙 🗸
◦ IsFolder	• General:
- Code	Synonym Full name
- Description	Comment
 Parent 	▼Use:
• Owner	Full Text Search Use
- DeletionMark	▼ Presentation:
🗢 Ref	Fill check Display error
- Predefined	Fill value
- PredefinedDataName	Fill from filling da

Figure 8-45. Standard attributes

After that, select **Comment** attribute, and in **Properties** select the **Multiline mode** and the **Extended edit** check boxes since in **Comment** is used to keep notes regarding the person. Enabling these options will make adding and editing of notes more convenient.

	✓ Presentation:
① ① Contact Types	Password mod
□	Tooltip
\odot 🛥 Attributes	
- Gender	
Relation Type	Mask
Comment	Multiline mode 🕑
① Tabular sections	Extended edit 🕑
⊕ 🗊 Forms	e

Figure 8-46. Enabling the Multiline mode and the Extended edit options

Next, open the list form of the **People** catalog and delete the **Code** form item. Then to add the **Comment** column to the list, expand the **List** form attribute in the right panel and drag the **Comment** attribute inside the **List** form table in the left panel.

🗮 Catalog People: ListForm						
🕒 🖉 🗶 🚖 😓 🗐 🖬	5 🐻 🖋 ×					
📰 Form	Attribute	Us				
Command bar		alv alv				
ListSettingsComposerUserSettings	🗆 🛄 List					
⊖ III List	Code					
Command bar	Comment					
 Description 	① ① Contacts					
Comment	Deletion Mark					
	Description					
	🕀 🚺 Gender					

Figure 4-47. Adding the Comment attribute to the managed form

Set **Width** properties of the **Description** and the **Comment** form items to **10**. You can select more than one form item by holding down **Ctrl** key and

clicking on each attribute. When more than one form item is selected, you can change most properties for all of them in **Properties**.

Now, start the application in the **1C:Enterprise** mode and review the changes to the list form of the **People** catalog. This time the **People** list on **Start page** looks more useful.

Deonlo

Deenle

reopie			
Create	Find	More	•
Full name	Ļ	Comment	÷
😑 Georgi	а	sophomore ;)	
😑 Grandr	nother	the best grandmother in the world	
😑 John S	heldon	neighbor	

Figure 8-48. The adjusted list form of the People catalog

The list form of the **People** catalog is also updated in the **Contacts** section where it is also placed. In addition, the item form of the catalog, which is generated by the platform automatically, also reflected all changes that you have recently made, such as **Synonym** of **Description** attribute and **Multiline mode** of **Comment** attribute.

reopie		
Create Find.	Cancel search	
Full name	Grandmother (People) (1C:Enterprise)	M M+ M- 🗆 🗙
😑 Georgia	Grandmother (People)	
	Save and close Save	More 👻
 Grandmother 	Code: 000000001	
 John Sheldon 	Full name: Grandmother	
	Gender: Female	•
 Sarah Wells 	Relation type: Family	· - □
	Comment:	
	the best grandmother in the world	More 🔻
	Add	More +
	# Туре	Value
	1 Email	granny@1c.com
	2 Skype	granny
	3 Address	785 5th Avenue, Nev

Figure 8-49. The adjusted item form of the People catalog

Make a couple more improvements to the application. Adjust the **Events** catalog to make it more user-friendly. To do so, set **Title** as **Synonym** for the **Description** form item in **Standard attributes**, and set **Attendee** as **Synonym** for the **Participant** attribute of the tabular section, and then select the **Multiline mode** and the **Extended edit** check boxes for the **Details** attribute.

🗧: Standard attribu 🗕 🗖 🗙	: Standard attributes - Description 🕚 🗙
Standard attributes:	51 🛃 🔎 × 🔸
IsFolder Code Description Parent	▼General: Synonym Title Comment ▼Use:
-	Eul Tort Complete
Figure 8-50. Setting Synonym	for the Description Standard attribute
 Tevents Attributes BeginDate EndDate Category Details Tabular sections Participants 	Properties: Participant Properties: Participant Properties: Participant Synonym Attendee Comment Type CatalogRef.People
Participant Forms	▼Use:
Figure 8-51. Setting Synonym for t	the Participant attribute of tabular section
 ○ III Events ○ Attributes □ BeginDate 	
 EndDate Category Details Tabular sections 	Mask Multiline mode

Figure 8-52. Enabling the Multiline mode and the Extended edit options

Extended edit

Participants

Participant

Now, you can review the changes. The **Events** catalog now looks much better.

$\frown \leftarrow \rightarrow \text{Events}$	😉 Cafe with Georgia (Ev (1C:Enterprise) 🔯 🗐 🛐 M M+ M- 🗖 🗙
Create Find Cancel sea	Cafe with Georgia (Events)
Title	Save and close Save More
 Cafe with Georgia 	Code: 000000006
	Title: Cafe with Georgia
🗢 Club	Begin date: 10/ 7/2013 4:00:00 PM
	End date: 10/ 8/2013 7:00:00 AM
- College	Category: Rest 🗸 🗗
 College financial assistance 	Details: ;))))
 Meeting relatives 	
	Add More 🕶
	# Attendee
	1 Georgia

Figure 8-53. The adjusted Events catalog

Object presentations

You might already noticed that names of items in **Navigation panels** of sections and in **Create** groups of **Actions panels** are the same, both opening of list form and creating a new item are plural. A beginner user will be confused with that.

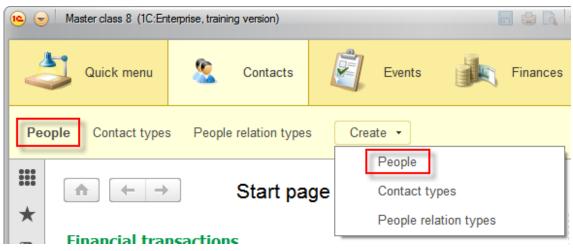


Figure 8-54. Same names for opening the list form and creating a new item

It is easy to fix. All that you need to do is to adjust **Object presentation** in **Properties** of each catalog. In other words to specify its singular name. Do this for each catalog as follows:

- For the Contact types catalog set Object presentation to Contact type
- For the People relation types catalog set Object presentation to People relation type
- For the People catalog set Object presentation to Person
- For the Event categories catalog set Object presentation to Event category
- For the Events catalog set Object presentation to Event

Catalog Contact Ty	pes		_ 🗆 ×
▶ Main	Name:	ContactTypes	
Subsystems			<u> </u>
Functional options	Synonym:	Contact types	
Hierarchy	Comment:		
Owners			
Data	Object presen	itation:	
Numbering		Contact type	
Forms	Extended obje	ect presentation:	
Input fields			

Figure 8-55. Adjusting Object presentation

Check how it looks like. Now names in the **Create** group are more grammatical.

📧 🕤 Master class 8 (1C:Enterprise, training version)						Į	a 🖨 🖪	
	Quick menu	<u>&</u>	Contacts		Ż	Events		Finances
People	Contact types	eople	e relation types	6		ite 🔻		
	♠ ← →		Start pa	ge	ļ	Person Contact ty	pe	
* ~ F	inancial trai	saction	IC .			People rel	ation type	

Figure 8-56. Difference between Synonyms and Object presentations

Quick selection of values

Now, try to change **Relation type** of the **People** catalog item or **Category** of **Event**, or **Type** in one of rows of contacts of a person. When you click a field that allows selecting values, to select a value you can either start typing first characters of a value description or click the **Show all** link and see the list of all items in a separate window. If you used the second method, to select a value, click the row with that value and then click **Select** or simply double-click the row.

Hello, 1C

Grandmother (F	^p erson) (1C:Enterprise)	Ŕ	📰 🛐 M M+ M- 🗆 🗙
Grandmoth	ner (Person)		
Save and	close Save		More 💌
Code:	00000001		
Full name:	Grandmother		
Gender:	Female		•
Relation type:	Family		→ 2
Comment:	Enter the search stri) P (1C:Enterprise) 😭	📰 🛐 M M+ M- 🗆 🗙
the best grand	Click <u>Show all</u> to sel Click <u>Cireate item</u>	People relation t	ypes
	Show all	Select Cre	More 🔻
		Description	↓ Code ^
Add		 Acquaintance 	00000005
# Ту	ре	😑 Buddy	00000004
1 Er	mail	- Family	00000001

Figure 8-57. Selecting values from the list of catalog items

In cases when the list of items in catalog is short, opening an additional window is not necessary and slows down operation with an application. You can change the application behavior in **Properties** of the catalog to one when its items will be selected from a drop-down list. To do this, select the desired catalog in **Configuration** metadata object tree, and then select the **Quick choice** check box in its **Properties**. Select this check box for the **Contacts types**, the **People relation types**, and the **Event categories** catalogs.

Properties: Contact Types		×
₽‡ 📑 🇃 × 🗸		
Full Text Searc Use	-	
Include in the 🗸		
Main Presental As description	-	
Editing Method In dialog	-	
Quick choice 🗹		
Choice Methoc Both ways	-	

Figure 8-58. Enabling the Quick choice option

Now see it in the 1C:Enterprise mode. A drop-down list is displayed to select values.

Grandmother (Per	son) (1C:Enterprise)	📰 🛐 M M+ M- 🗆 🗙	¢
Grandmothe	er (Person)		
Save and cl	ose Save	More 🔻	
Code: 0	00000001		
Full name:	Grandmother		
Gender:	emale		
Relation type:	amily		
Comment:	Acquaintance		
the best grand	Buddy		
	Family		
	Friend		
Add	Girlfriend	More -	
# Ty	+		

Figure 8-59. Drop-down list as Quick choice is enabled

Adjusting reports

Look at the **Daily chart** report. In general, it is fine, but a user may be confused about some technical information, the format of dates, and the legend. In addition, it would be much better to be able to see points of the plot directly on the chart, not only when hovering over them with the mouse.

	Select variant	More -
Begi	n of period:/ / :: AM # End of period:/ / :: AM	
200		
180		
	\sim	
160		
140		
120		
100	Amount Ending	
8	Amount Ending balance 10/1/2013 12:00:00 AM 100	
60		
00		
40		
20		
	10/1/2013 12:00:00 AM 10/3/2013 12:00:00 AM 10/7/2013 12:00:00 AM 10/2/2013 12:00:00 AM 10/5/2013 12:00:00 AM	

Figure 8-60. Daily chart report

It is easy to improve this chart. In the Designer mode, open the **MainDataCompositionSchema** template for this report.

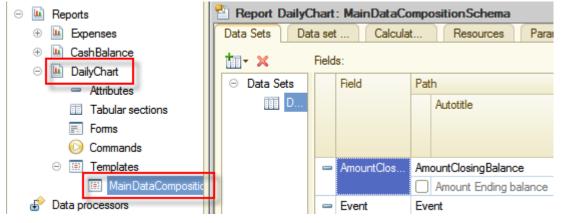


Figure 8-61. MainDataDompositionSchema of the Daily chart report

On the **Data sets** tab, adjust the legend by editing titles of **AmountClosingBalance** and **Period** fields. In addition, for **Period** field set a more user-friendly date format.

Next, set the title of the **AmountClosingBalance** field to **Balance**. To be able to do this, find this field in the list of fields and then select check box on the left to the title that you want to edit. Autotitle caption of this column will automatically change to **Title**. Then instead of **Amount Ending balance** enter **Balance**.

皆 Report Daily	Chart	: MainDataCo	mpos	sitionSchema						
Data Sets Da	ata se	t Calcula	t	Resources	Parame	t	Tem			
Fields:										
⊖ Data Sets		Field	Path			Field	restrict			
III D				Title		Fi	C (
						Attrib	ute Re			
						Fi	C (
	-	AmountClos	Amou	untClosingBalanc	e					
				Balance		~				
	-	Event	Even	ıt		~				

Figure 8-62. Setting the title of the AmountClosingBalance field

Do the same with the **Period** field, change its title to **Date**.

	Field		h	Field	restric	tion		Role	Presentatio
			Title	Fi	C	G	0		Ordering
				Attrib	ute R	estrict	ion		Expression
				Fi	C	G	0		
-	AmountClos	Am	ountClosingBalance						
		\checkmark	Balance	V	\checkmark	\checkmark	\checkmark		
-	Event	Eve	ent	~		4	4		
		\Box	Event	~		4	4		
-	Period	Per	iod					Period, 1	
			Date	~	\checkmark	\checkmark	\checkmark		
-	Person	Per	son	~		\checkmark	\checkmark		
			Person	4		4	\checkmark		

Figure 8-63. Setting the title of the Period field

Now for the **Period** field change the date format. Enter the appearance column of **Period** by clicking **Select**

Path	Field	l restri	ction		Role	Presentatio	Hierarchy Check:	Value Type	Appearance
Title	Fi	C	G	0		Ordering	Data set	Available	Edit options
	Attri	Attribute Restriction			Expressions	Parameter	values		
	Fi	C	G	0					
AmountClosingBalance									
Balance	~	1	4	4					
Event	~		4	1					
Event	~		~	~					
Period					Period, 1				
✓ Date	V	\checkmark	4	4					50
Person	V		V	V					
Person	\checkmark		\checkmark	\checkmark					

Figure 8-64. Editing the field appearance

Open the **Field format** editor, and find the **Format** property.

Field format		×
Parameter	Value	
Indents		
Autoindent		
Horizontal position	Left	
Vertical position	Тор	
Placement	Block	
Text orientation		
Format		
Mark negatives	False	
Unfilled mark	False	e
Minimum width		
Maximum width		
Minimum height		
Maximum height		*
	OK Cancel	Help

Figure 8-65. The Field format editor

Open Format String Wizard by clicking Select ..., and then click the Date tab and in the Date format field enter MM/dd/yyyy. After that, click OK.

Format String Wizard	×
Language (Country)	
Number Date Boolean	
Date format	MM/dd/yyyy
Local date format	
Empty date presentation	
- Example:	
4/ 2/2014 10:11:49 PN × 📦	04/02/2014

Figure 8-66. Format String Wizard

Next, click **OK** to close the **Field format** editor. After that value in the **Appearance** column on the **Data Set** tab for the **Period** field will look as follows:

Pat	Path		restri	ction		Role	Presentatio	Hierarchy Check:	Value Type	Appearance
	Title	Fi	C	G	0		Ordering	Data set	Available	Edit options
		Attribute Restriction			tion		Expressions	Parameter	values	
		Fi	C	G	0					
Am	ountClosingBalance									
	Balance	~	4	\checkmark	\checkmark					
Ev	ent	~		4	1					
	Event	~		4	4					
Per	iod					Period, 1				Format
Date		~	~	~	~					
-		-		-	-					

Figure 8-67. The adjusted Period field

The last thing is left to do is to adjust view of plot points. To do this, click the **Settings** tab. Then, on the bottom tabs click the **Other settings** tab.

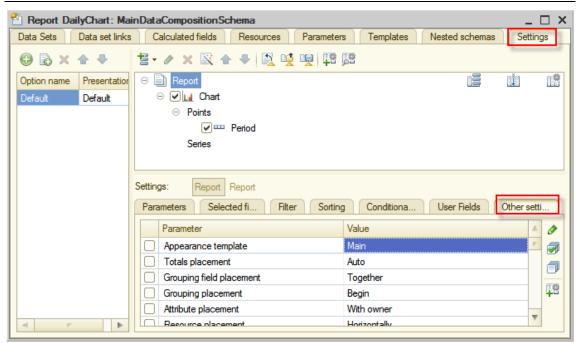


Figure 8-68. The Other settings tab

Find the **Label list** parameter under **Chart type**, and then from the dropdown list select **Value**.

Parameters Selected fi Filter	Sorting Conditiona User Fields	Other setti
Parameter	Value	A Ø
Chart type	💸 Line	
Base value		· (
Ignore base value	True	
Label list	Series + percent	👎
Separation mode	Series	A
Dieplay data table	Value	V

Figure 8-69. Selecting Value as a value of the Label list parameter of Chart type Now, check it in the 1C:Enterprise mode to see the result of changes.

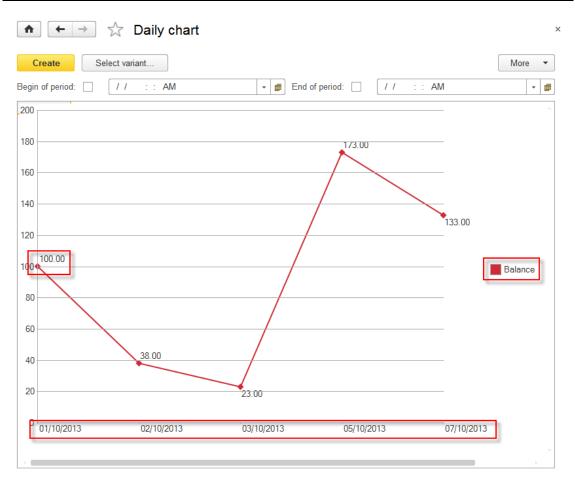


Figure 8-70. The adjusted chart

Excellent! Everything that was required is implemented. The chart shows meaningful legend and values are displayed next to plot points. This allows analyzing finances easier.

Report variants

You might ask the right question: "Do I have to open the report the Designer mode each time when I need to change its view settings?" After all, the user might not be able or allowed to use the Designer mode, for example, when connecting through HTTP protocol. The answer is no, you do not have to use the Designer mode each time you need to change something in the report settings. Most of report settings, including view settings, can be changed in the 1C:Enterprise mode without engaging a developer.

As an example, you will change the **Expenses** report. The default view of the report, meaning the settings that the developer made for it, are as follows:

(Create Select variant			More	-
egii	n of period: 🗌 🛛 / / 💠 : : AM 💌 🗊 E	nd of period:		:: AM -	
	Person	Amount	Amount	Amount	
	Event	Turnover	Receipt	Expense	
-		128.00	150.00	22.00	
	Club	-12.00		12.00	
	College	-10.00		10.00	
	College financial assistance	150.00	150.00		
)	Georgia	-80.00		80.00	
		-50.00		50.00	
	College	-30.00		30.00	
-	Grandmother	100.00	100.00		
	Meeting relatives	100.00	100.00		
)	Sarah Wells	-15.00		15.00	
	Club	-15.00		15.00	
	Total	133.00	250.00	117.00	

Figure 8-71. The Expenses report with default settings

Imagine that a chart would be easier for the analysis, so you need to change the report into a chart. Click **More** and then click **Change variant...**

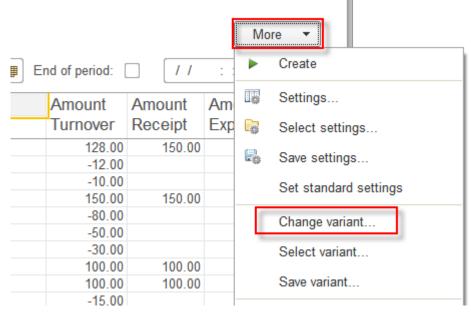


Figure 8-72. Changing a report variant

You will see the report settings window, the same as you used in the Designer mode when configured the data composition schema for this report. The default settings are displayed. The one that you have configured in the Designer mode for this report.

Click the **Person** node, and then click **Delete** (Del) to delete these settings. Confirm the deletion when the dialog appears.

```
Hello, 1C
```

Report "Expenses" variant "Default" - Hello, 1C (1C:Enterprise))		
Report "Expenses" variant "Default"			
Add Group Delete	Custor	n settings iten	n pro
Report structure		M	
⊖ 📄 Report			
⊖ 🗹 🚥 Person			
Event			
1 Para III Cust III Grou III Field	s 🕅 Filter	i Sorting	9

Figure 8-73. Deleting settings of the current variant

After that, right-click the **Report** node, and in the context menu click **New** chart.

Report "Expenses" variant "Default" - Hello, 1C (1C:Enterprise)							
Report "Expenses" variant "Default"							
Add 🔻	Gr	oup Delete		Cu	istom	settings	s ite
Report structure)					M	
Report	ta	New grouping			Ins		
		New table					
	t.	New chart					
-	Ð	New nested report					
î Para 🕼	Ŀ	Group				S	orti
Custom settir	Ľ	Ungroup					
	×	Delete			Del		
Parar	車 帶	Custom settings item p	roper	ties			
B	egin o	r penou			C	ບອເບາ ກ d	ate

Figure 8-74. Creating a chart

Next, right-click the **Points** node, and in the context menu click **New** grouping... In the opened window, select the **Event** field.

Report structure		M		1
⊖ 📄 Report	12			
Chart Chart Points Series Grouping - Helio, 1C (1C:Enter Grouping	prise)	×		
Field: Type: Elements		Select Field	- Hello, 1C (1C:E	Enterprise)
OK Custom settings it on proportion	Cance	Select Fi		
Parameter			Event	
Begin of period			Person	
End of period			Amount Ending	g balance

Figure 8-75. The Event grouping field

Then, right-click the **Series** node, and in the context menu click **New grouping...** In the opened window, select the **Person** field.

Report structure		M	
⊝ 🗹 📊 Chart			
⊖ Points			
Grouping - Hello, 1C (1C:Enterpri	se)	×	
Series Grouping			
Field:	\rightarrow	×	
Type: Elements		Gelect Field	- Hello, 1C (1
Custom settings	Cance	Select Fi	eld
Parameter		Available f	fields
Begin of period		+ • •	vent
End of period			Person Amount End

Figure 8-76. The Person grouping field

Now, click the **Report** node, then click the **Fields** tab.

Report "Expenses" variant "Default"

Add Group Delete	Custor	n settings iten	nroperties	More	•
Oloup	Ouston	r settings iten	i properties	WINC	
Report structure		M	1.	1	
Caral Cara Cara					
○ ☑ Lu Chart					
⊖ Points					
Event					
⊖ Series					
Verson					
					Þ
🛗 Para 🛗 Cust 🛗 Grou 📑 Field	s 🕅 Filter	iii Sorting	g 🚮 Cond.	🏥 Add	it

Figure 8-77. Report fields

On the **Fields** tab check that the **Amount Turnover** field is placed right after **Person** and **Event** fields. In addition, check boxes on the left from **Amount Receipt** and **Amount Expense** fields are cleared.

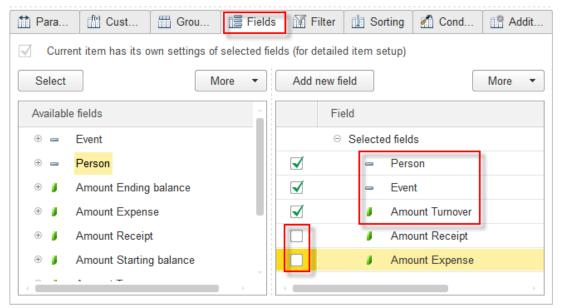


Figure 8-78. Selected fields of the report

Now click the **Additional settings** tab, and for the **Chart type** property select **Bar Graph**.

Rapid application development tutorial

🛗 Para	ft Cust	Grou	📑 Fields	Filter	1 Sorting	🚮 Cond	📑 Ad	dit
Curren	t item has its c	wn additional s	settings				_	
Custom s	settings item p	roperties					More	•
Parameter				/alue				*
Horizontal placement of overalls				Auto				
Vertical p	lacement of ov	eralls	A	Auto				
Field title	type		A	uto				
○ Chart type			C	olumn 3D			-	
Base value				Stock			÷	
Ignore base value				Open-High-Lo	w-Close			
<pre></pre>				Bar Graph				

Figure 8-79. Selecting the chart type

Click **Finish editing**, to save and close the report editor

Custom settings item properties	More
Parameter	Value
Horizontal placement of overalls	Auto
Vertical placement of overalls	Auto
Field title type	Auto
Chart type	Bar Graph -
Base value	
Ignore base value	Yes
	• • • • • • • • • • • • • • • • • • •

Figure 8-80. Saving the report settings

In the report window, click **Create** and see how dramatically the appearance of the report has changed. Now the user can clearly see what the most money is being spent for.

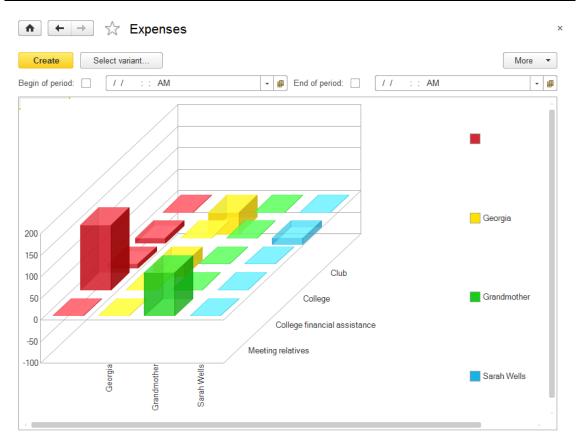


Figure 8-81. The new appearance of the Expenses report

You do not need to change the default settings of the report because the platform provides convenient save and load commands for report variants.

Click More and then click Save variant...

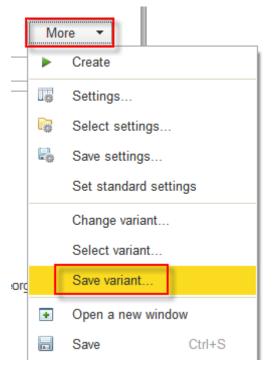


Figure 8-82. Saving the report variant

Input **3D Chart** for the name and click **Save**.

Save report variant - Hell (1C:Enterprise) 🗙					
Save report variant					
Previously saved variants:					
Cave the variant with the following name:					
3D Chart					
Save Cancel ?					

Figure 8-83. Specifying the report variant name

In the future, you will be able to use this variant as many times as it will be necessary. To load the report variant click **Select variant...**

Create	Select variant	More -
Begin of period:	Select report variant (1C:Enterprise)	/ /
	Select report variant	
	3D Chart	
	Default	
	Select Cancel ?	

Figure 8-84. Loading the custom report variant

Note that you can always restore the version that was created by the developer in the Designer mode because it is listed as **Default** in the list of report variants.

C	reate	Select variant		More -
Begin	of period:	Select report variant (1C:Enterprise) ×	1	/ 🔹 🗊
	Person	Select report variant	nt	Amount 🕤
	Event		ver	Receipt
Ξ.		3D Chart	28.00	150.0
	Club	Default	12.00	
	College	Doldan	10.00	
	College 1		50.00	150.0
	Georgia		B0.00	
		Select Cancel ?	50.00	
	College	Select Cancer ?	30.00	
	Crandmath	ar 1	00.00	100.0

Figure 8-85. Loading the default report variant

Functional options

Another useful part of 1C:Enterprise 8 is the functional options feature. Functional options allow you to group some of applied solution features and then with a single click manage whether the set of features will be used in the specific application or not.

Functional options are advantageous during the applied solution implementation. For example, the CRM that you create in this tutorial include three accounting sections: managing people, events, and finances. If you will decide to share your applied solution with your friends or to enhance its functionally and sell, you will see that not everything that the applied solution is capable of is useful for each user. For example, not everyone will need to keep all events that occur in his or her life. However, maintaining a list of acquaintances and keeping records of money income and outgoings is useful for most users.

Thus, keeping in mind that someone may find events feature not useful, you can segregate it to the functional option and let a user to decide whether to use it or not. If you do so, 1C:Enterprise 8 platform will maintain all references to objects of the functional option in all parts of the applied solution. Moreover, if the user does not need features of the functional option, the platform will automatically disable and hide those objects.

The platform will perform all these changes on its own, so that the developer does not need to write the script for that. The only thing that is required is to set the desired value of the functional option in the 1C:Enterprise mode!

Consider a simple example where the value of the functional option is stored in the constant of the Boolean type. If the value is True, then the functional option is enabled. If the value is **False**, then the functional option is disabled.

Because you are creating the first custom option of the application, it is a good time to create the **Settings** subsystem, and specify the **SettingsSubsystem.png** icon for it.

Subsystem Settings	3	×
Main	Name:	Settings
Functional options		
Content	Synonym:	Settings
Other	Comment:	
		nmand interface:
	Explanation:	
	Picture:	SettingsSubsystem

Figure 9-1. Adding the Settings subsystem

At once, place the new subsystem after the operational subsystems. To do this, click the root configuration node, and then in **Properties** click the **Open** link of the **Command interface** property.

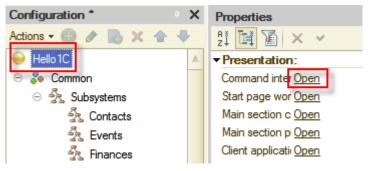


Figure 9-2. Opening the Command interface editor

Move the **Settings** subsystem to the bottom.

😡 Command interface			×
@ 😼 🚱	Filter by roles:	Not set>	◄
Command		Visibility	Visibility by roles
 Sections (manual order) 			
Contacts		 Image: A start of the start of	
Events		 Image: A start of the start of	
🕭 Finances		 Image: A start of the start of	
Settings			

Figure 9-3. Moving the Settings subsystem down

Create a constant.

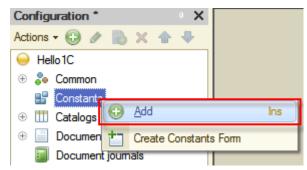


Figure 9-4. Creating a constant

Name it UseEvents, and set its Type to Boolean.

Configuration *	• ×	Properties: UseEvents
Actions 🗸 🕕 🖋 🔒 🗶 合	-	21 📴 🔟 🗙 🗸
\varTheta Hello 1C		▼General:
🕀 🐉 Common		Name UseEvents
😑 🔡 Constants		Synonym Use events
📲 UseEvents		Comment
🕀 🎹 Catalogs		
Documents		Type Boolean 💌
De sum est inume le		

Figure 9-5. Adjusting the Use events constant

After that, right-click the **UseEvents** node in the **Configuration** tree and then click **More** in the context menu.

• Common • Constants · Constants • UseEvents • Open Value Manager Module • Open Default Fom • Documents • Document jour • Documents • Document jour • Document jour • Data processor • Data processor • Charts of chara • Charts of calcu • Information regi • Accounting regi • Calculation regi • Calculation regi • Data processor • Charts of calcu • Information regi • Charts of calcu • Calculation regi • Calculation regi • Calculation regi • Tacke • Tacke	\varTheta Hello 1C			
Image: Second secon	🕀 🍣 Common			
• □ Catalogs ○ pen Value Manager Module • □ Documents ○ pen Default Form • □ Document jour • □ Create Constants Form • □ Reports △ Add • □ Data processor △ Add • □ Charts of chara △ Clone • □ Charts of accou △ Delete • □ Information regi △ Move Up • □ Accounting reg △ Move Down • □ Calculation regi △ Sort • □ Calculation regi □ more ferences to object	🖂 🔡 Constants			
(*) Image: Catalogs (*)	🔡 UseEvents		Open Value Manag	er Medule
Image: Second state Image: Second state Create Constants Form Image: Second state Add Ins Image: Second state Image: Second state F2 Image: Second state Image: Second state F2 </th <th>① Catalogs</th> <th>-0</th> <th>Open value Manag</th> <th></th>	① Catalogs	-0	Open value Manag	
⊕ Enumerations Create Constants Form ⊕ Reports Add Ins ⊕ Data processor Change F2 □ Charts of chara Clone F9 ↓ Charts of accou Delete Del ⊕ Information regi Move Up Ctrl+Shift+Up ⊕ Accounting regi Sort Sort ● Calculation regi Find references to object	① Documents		Open Default Form	
	Document jour	to.	Create Constants Fo	m
 Data processor Change Change Change Clone Clone P Charts of accou Charts of calcul Information regi Accumulation r Accounting reg Calculation regi Business proce Find references to object 	⊕ {} Enumerations	_		
Image: Charts of chara Image: Charts of chara Clone F9 Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul Image: Charts of calcul	① Image: Below the second s	Ð	Add	Ins
Image: Charts of account Image: Charts o	🔄 Data processor	1	<u>C</u> hange	F2
Information regi Image: Delete	Charts of chara	B	Clone	F9
 Charts of calcul Information regi Accumulation regi Accounting regi Calculation regi Business proce 	দি Charts of accou		Delete	Del
 Accumulation r Accounting reg Calculation regi	🗢 Charts of calcu		Delete	
Image: Solution of the second seco	Information regi	÷	Move <u>U</u> p	Ctrl+Shift+Up
Calculation regi Business proce Find references to object	🕀 🧧 Accumulation n	-₽-	Move Down	Ctrl+Shift+Down
Calculation regi Find references to object	Accounting reg		Sort	
	Calculation regi		-	
Taska Sind Defense and in Object	Business proce	p	Find references to a	bject
	Tasks	Q,	Find References in	Object
🗞 External data s 👘 More Alt+Shift+Enter	ሌ External data s	ð	More	Alt+Shift+Enter
Properties Alt+Enter		1	Properties	Alt+Enter

Figure 9-6. Opening the More properties of the UseEvents constant Then, include the new constant to the **Settings** subsystem.



Figure 9-7. Including Use events to the Settings subsystem

Then create the **Use events** functional option and adjust it to keep its value in the **Use events** constant. For that click **Functional options** node in the **Configuration** tree, then click **Add** (Ins) in the command bar of the **Configuration** tree. Then input **UseEvents** as the name of the functional option, and then select **UseEvents** constant in the **Data path** property.

Hello, 1C

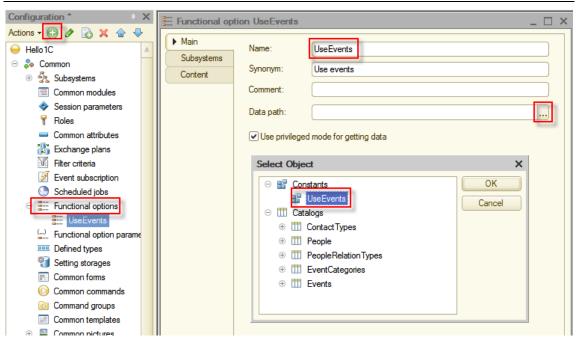


Figure 9-8. Creating the UseEvents functional option

Because of these actions, a command that allows to open the form and change the value of **Use events** functional option in the 1C:Enterprise mode will appear in the **Settings** section.

Now you need to include configuration objects and object properties to the functional option. Ask yourself: "What in your applied solution is related to events?" The answer is:

- the **Events** subsystem;
- the **Events** catalog;
- the used only for events **Event categories** catalog;
- the Event attributes of tabular sections of the Cash receipt and the Cash payment documents;
- the **Event** dimension of the **Financial transactions** accumulation register.

It is very easy to include the mentioned above objects and properties to the **Use events** functional option. To do this, double-click the **Use events** functional option to open it in the editor and then click **Content** tab. After that, in the tree on top set check boxes on the left of configuration objects and object properties. For verification see the panel on bottom that the platform fills with selected objects.

Emiliaria Functional opt	tion UseEvents _	□ ×			
Main					
Subsystems	○ SinancialTransactions				
Content	□ ↓ Dimensions				
	Event	c			
	C 1., Person	T			
	Objects included in the functional option:				
	🗢 🗞 Common				
	Events				
	☐ Catalogs				
	EventCategories				
	○ Documents				
	⊖ ☐ CashReceipt				
	○ □ Tabular sections				
	□ II Receipts □				
	😑 Event				
	⊖ 🧾 CashPayment				
	Tabular sections				
	⊖ 🗊 Expenses				
	Event				
	 Accumulation registers FinancialTransactions 				
	 Imancial transactions □ ↓ Dimensions 				
	Let Event				
Action	ns 👻 Close Help				

Figure 9-9. Including objects to the functional option

Now, start the application in the 1C:Enterprise mode, click the **Settings** section, click the **Tools** menu, and then click **Use events**.

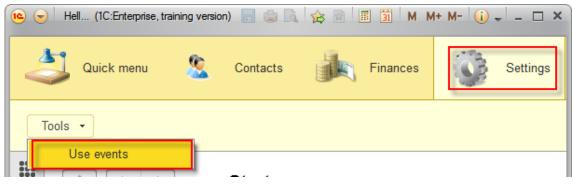


Figure 9-10. Opening the Use events constant form

Enable **Use events** to make sure that the created functional option did not affect anything in the application. For this, select the **Use events** check box, and then click **Save and close**.

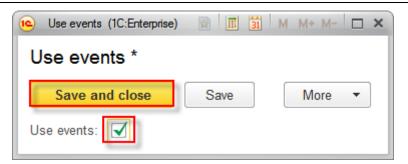


Figure 9-11. Enabling the Use events functional option

Close the main application window and start the application in the 1C:Enterprise mode once again to see that the application works same as it was before you created the functional option.

Now, in the **Settings** section disable **Use events** functional option.

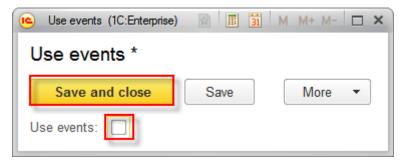


Figure 9-12. Disabling the Use events functional option

After saving changes, restart the application in the 1C:Enterprise mode.

Look at the application carefully.

First, that you might mention is that the **Events** section and various object references related to events are no longer displayed on **Start page**.

👝 😠 Hell (1C:Enterprise, training version)	×
Quick menu	Contacts Finances Settings
People Financial transactions	Create Reports Person
	Cash receipt Cash payment
า Financial transactions	

Figure 9-13. The Events section and Events catalog are disappeared

Second is that references to events have disappeared from all forms, including the information displayed in reports.

$\textcircled{\ } \checkmark \qquad \checkmark \qquad \checkmark \qquad \qquad$	Financial tr	ansactions	$\bigstar \longleftarrow \Rightarrow \swarrow \text{Expenses} \qquad \times$
Find Cancel s	earch	More •	Create Select variant More 🔻
Date ↓	Person	Amount	Begin of period: End of period: / - #
+ 10/1/2013 9:0	Grandmother	100.00	Person Amount Amount Amount
- 10/2/2013 4:4	Georgia	50.00	Turnover Receipt Expense
- 10/2/2013 4:4		12.00	128.00 150.00 22.00 Georgia -80.00 80.00
- 10/3/2013 12:	Sarah Wells	15.00	Grandmother 100.00 100.00
- 10/3/2013 12	Sarah vvens	15.00	Sarah Wells -15.00 15.00
+ 10/5/2013 4:0		150.00	Total 133.00 250.00 117.00
- 10/7/2013 4:0	Georgia	30.00	
- 10/7/2013 4:0		10.00	
-			

Figure 9-14. The Events column in the list form and Event details of the report are disappeared

You made what was required, the use of events in the applied solution is now optional and depends on settings of a specific application.

Cross-platform design

Now it is time to explain on what operating systems and devices you can run your application.

Linux

You can use the application that you have developed in this tutorial without any modification both on computers with Microsoft Windows and Linux operating systems.

1C:Enterprise 8.3 platform have introduced following analogs of already available for Microsoft Windows client applications:

- **Thin client** allows users to run applications in the 1C:Enterprise mode;
- **Thick client** allows users to run applications in the 1C:Enterprise and the Designer modes.

Client applications support both file and client/server modes. They are available both for x86 and x86-64 architectures.

This means that now not only the 1C:Enterprise mode users can work on Linux, but also developers and administrators.

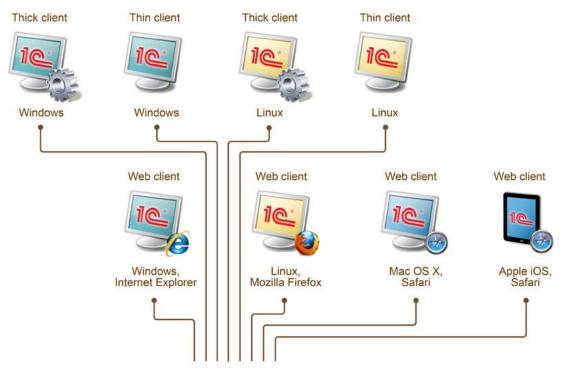


Figure 10-1. Cross-platform design

The full list of supported operating systems you can read in System requirements.

In Linux, the application will look and feel the same as it does on Microsoft Windows.

Rapid application development tutorial

	10 👻	Hello, 1C (1C.Enterprise) 🔚 🖨 🕼 M M+ M- 🕧 🚽 🗆 🗙
File Edit View Bookmarks Settings		
administrator@linux:~\$		👤 Quick menu 🙎 Contacts 🖉 Events 🚮 Finances 🔞 Settings
	Peo	ple Events Financial transactions Create - Reports -
	Ⅲ ★	★ → Start page
	Ð	Financial transactions People
File Edit Configuration Debug Administration	_	Find Cancel search More Create More
	<u> </u>	
		Date ↓ Event P Amoun Full name ↓ Com
Configuration		- 10/7/2013 4:00 College G Georgia sopho
Actions - 💿 🖉 💽 🗙 🛧 🗣 🔲 🎉		– 10/7/2013 4:00 College
Hello1C		Grandm the best
 ⊕ Sommon ⊕ Gonstants 		grandm
Constants Catalogs		in t
⊕ 📃 Documents		Cash balance - John neighbor
Document journals		Sheldon
⊕ {} Enumerations		Create Select variant More
⊕ Image Reports		Period: 🖌 Beginning of this day 🗣 🗐 🤤 Sarah MKTG Wells 101
Charts of characteristic types		
Tr Charts of accounts		Parameters.Period Amount
🗢 Charts of calculation types		Balance 4/14/2014 12:00:00 AM 133.00
Information registers		Total 133.00
Accumulation registers		
 Accounting registers Calculation registers 		
	-	
Press F1 for help	÷	Current calls: 2 Accumulated calls: 51
🔣 🝋 1 Hello, 1 C		🖽 🗙 📑 🕶 🗮 🔛 💥 us 🗤 👼 🔺 05:25 PM

Figure 10.2 1C:Enterprise on Linux

Web client

It is a common situation when you are not at your workplace, but need some data urgently. In this case, an access through the Internet using a web browser might save you.

It is no doubt that you can develop a web interface for any system. To do this you need to develop, debug and deploy it as well as maintain compatibility with different web browsers. 1C:Enterprise platform allows developers to save efforts on development of the web interface.

The platform allows you almost in a single click publish on the web server entire the application that you have developed in this tutorial. After that, users will be able to access the application using regular web browsers. At the time of writing this book there are four supported web browsers, they are: Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari.

This means that users do not need to install any of 1C:Enterprise applications or licenses, the only requirements are the connection to the Internet and one of supported web browsers. At the same time, you will need to have a server where 1C:Enterprise platform and a web server are installed. At the time of writing this book there are two supported web servers, they are: Microsoft IIS and Apache.

To install the web server, click **Start**, then click **Control panel**, and then click **Uninstall a program**. On the navigation panel click **Turn Windows features on or off**. In **Windows Features**, select **Internet Information**

Services. The default set of features is enough. Click **OK** to setup and close this window.

🛃 Window	s Features	<u>- 🗆 ×</u>
Turn Win	dows features on or off	(?)
	eature on, select its check box. To turn a feature eck box. A filled box means that only part of the f n.	
	Indexing Service	
	Internet Explorer 10	
🛛 🛨 🔽 🚺	Internet Information Services	
	Internet Information Services Hostable Web Co	re
🛛 🗉 🔽 🚺	Media Features	
🛛 🛨 🔽 🚺	Microsoft .NET Framework 3.5.1	
🛛 🗉 🗖 🚺	Microsoft Message Queue (MSMQ) Server	
🛛 🗄 🗹 🚺	Print and Document Services	
	Remote Differential Compression	
	RIP Listener	-
	ОК Са	ncel

Figure 11-1. Installing the Internet Information Services web server

To publish web client you need to have the **Web server extension modules** component of 1C:Enterprise platform installed. To install the component, click **Start**, then click **Control panel**, then click **Uninstall a program**, in the list of applications find **1C:Enterprise 8 (training version) (***<version number>***)** and select it. The version must match the version that you use for publishing the application. Then click **Change** on top of the list. In the installation wizard select the **Modify** option, and click **Next**.

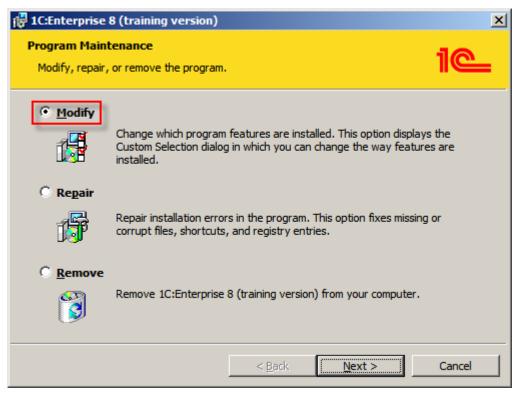


Figure 11-2. Modifying the 1C:Enterprise installation

On the next page select **This feature, and all subfeatures, will be installed on local hard drive** for **Web server extension modules**. After that, follow the wizard confirming default settings.

🙀 1C:Enterprise 8 (training version)	×
Custom Setup Select the program features you want installed.	1@
Click on an icon in the list below to change how a feature is installed.	
IC:Enterprise Feature Description Web server extension modules Web server extension	n modules
This feature will be installed on local hard drive.	
This feature, and all subfeatures, will be installed on local hard drive.	
× This feature will not be available.	- h

Figure 11-3. Installing the Web server extension modules component

Enabling an access to the application through the Internet is simple. Open the application in the Designer mode. In **Main menu** click **Administration** and then click **Publishing on web-server...**

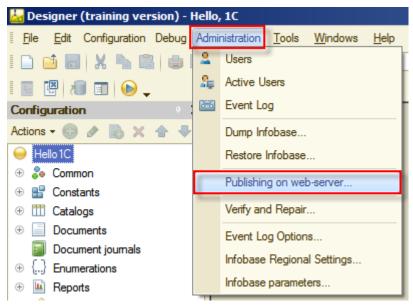


Figure 11-4. Opening the Publish to the web server wizard

In the **Publish to web server** wizard in **Name** field input **hello_1c** and in **Directory** field see that the last folder is changed to **hello_1c**. Then, click **Publish**.

Hello, 1C

Publish to web	server		×
Main Addi	tional		Publish
Name:	hello_1c		Remove
Web server:	Internet Information Services	•	
Directory:	C:\inetpub\wwwrodt\hello_1	c\	Save
Publish thin	client and web client		
Publish web	services by default		Load
Publish web	services		

Figure 11-5. Publishing the application on the web server

The platform will ask for confirmations during publishing. Agree with them.

Web client publication	×
It is recommended to restart w Restart now?	eb server when this publication parameters change is implemented.
	Yes No

Figure 11-6. An example of the publication confirmation dialog.

After the application is published, you can access it from any place around the world. To connect to the application you will need to know only its URL.

Notice: Details of configuring web servers are skipped in this book. In short, you will need to let **IIS_IUSRS** user on the server an access to 1C:Enterprise installation **bin** folder, to the folder that you have specified in **Directory** field of the **Publish to web server** wizard, and to the folder where your infobase is placed. Then, you need to enable **Anonymous Authentication** in Authentication settings of the **hello_1c** IIS application and for 64-bit Windows you need to set to **True** value of **Enable 32-Bit Applications** in advanced settings of the **DefaultAppPool** application pool.

In the tutorial you run it on your local computer, thus the URL will be local only. To log on type http://localhost/hello_1c in the address bar of your web browser and you will see the 1C:Enterprise interface, with which you are already familiar.

Hello, 1C (1C:Enterprise, training version)	Windows Inter	net Explorer		•	_ 0
Solution of the style of th		IC (1C:Enterprise, tr			☆ ☆
Hello, 1C (1C:Enterprise, training version		ic (IC:Enterprise, tr	al ^		
	Contacts	Even	ts Fin	11 Fa.	ettings
People Events Financial transaction	ons Create	e - Repo	orts 🔹		
Sta	rt page				
Financial transactions				People	
Find Cancel search			More •	Create	More •
Date ↓	Event	Person	Amount *	Full name ↓	Comment
- 10/3/2013 12:00:00 PM	Club	Sarah Well	15.00	- Georgia	sophomor
+ 10/5/2013 4:02:25 PM	College fin		150.00		
- 10/7/2013 4:00:49 PM	College	Georgia	30.00	- Grandmother	the best
- 10/7/2013 4:00:49 PM	College		10.00		grandmotł in the
				- John Sheldon	neighbor
Cash balance					
Create Select va	riant		More •	🛥 Sarah Wells	MKTG 101
Period: 🗹 Beginning of thi	s day		· …		
	od: 4/15/201 mount	4 12:00:00 AM	Å		
	alance		_		
4/15/2014 12:00:00 AM Total	133.00 133.00				
			r	L	*

Rapid application development tutorial

Figure 11-7. Web client in Microsoft Internet Explorer

As it is said above, you can use your favorite web browser to work with your application.

Hello, 1C

Hello,	1C (1C:Enterprise, training version)	+				
+)@	localhost/hello_1c/en_US/			☆ マ C Soogle	P 🖡 1	🌳 🗖 🕅
	Hello, 1C (1C:Enterprise, training v	ersion)				M M+ M- 🥡
4	Quick menu	Contacts	Event	ts Finan d	ces 👸 Set	tings
Peop	le Events Financial transactio	ns Create	• Repo	rts 🔹		
	★ → Sta	rt page				
r L	Financial transactions				People	
L L	Find Cancel search		More 🔻		Create	More •
•	Date ↓	Event	Person	Amount	Full name ↓	Comment
	- 10/3/2013 12:00:00 PM	Club	Sarah Well	15.00	🗕 Georgia	sophomore
	+ 10/5/2013 4:02:25 PM	College fina		150.00		;)
	- 10/7/2013 4:00:49 PM	College	Georgia	30.00	- Grandmother	the best
	- 10/7/2013 4:00:49 PM	College		10.00		grandmoth in the
					- John Sheldon	neighbor
	Cash balance					
	Create Select va	riant		More 🔻	🛥 Sarah Wells	MKTG
	Period: 🗹 Beginning of this	day		•		101
	Data parameters: Period: 4/15/2014 12:00:00 AM					
		mount alance				
	4/15/2014 12:00:00 AM	133.00				

Figure 11-8. Web client in Mozilla Firefox

Notice: The 1C:Enterprise (training version) has a set of limitations, thus you can not protect it with the password as users feature is disabled for the training version. In addition, to log on using the web client you need to quit the Designer mode first as there can be only one concurrent user in the training version. The actual at the time of writing this book limitations are listed in 1C:Enterprise 8 (training version) chapter on page 149.

You can see live demo on 1C:Developer Network in Applications section.

Mobile platform

One more interesting thing is that the application that you have developed with almost no efforts can be started on iOS or Android mobile devices. This feature is provided by the mobile platform that is a part of 1C:Enterprise 8.3 platform.

The look and feel of applications for Android and iOS devices can be a slight different.

Notice: Systematic instructions on how to create a mobile application for end users you can find in 1C mobile application rapid application

development tutorial. In this tutorial, you are going to start your application in the mobile platform for developers.

Details on features of applications based on 1C:Enterprise platform for mobile devices you can see in Chapter 25. Developing solutions for the mobile platform of 1C:Enterprise 8.3. Developer guide.

1C:Enterprise 8 mobile platform is a set of tools that let you to create applications that work on Android or iOS mobile devices. Those devices are mostly smartphones and tablet PCs.

The 1C:Enterprise mobile application that you install on the mobile device is a combination of the mobile platform and the infobase. The mobile infobase is similar to the file infobase. It consists of a database that stores data in a file and a mobile applied solution that is a program, which can be started on a mobile device and manage data in the database.

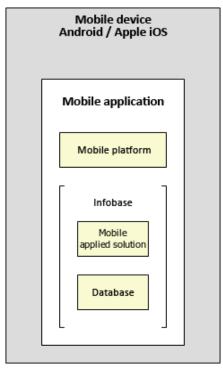


Figure 12-1. 1C:Enterprise 8 mobile platform

Thanks to the mobile platform you can right now, without using any thirdparty IDE, install, start, and see how works your application on the mobile device. As an example, you will do this for the Android device.

For this, you will need following:

- Make your applied solution compatible with the mobile platform. You will have to do a few changes for that.
- The mobile platform for developers installed on an Android mobile device.
- The web server running on the computer where you develop the application, and accessible by IP from your mobile device.

Start the application in the Designer mode and verify it for capability of being started on mobile devices. For this, in **Main menu** click **Configuration**, and then click **Check configuration**...

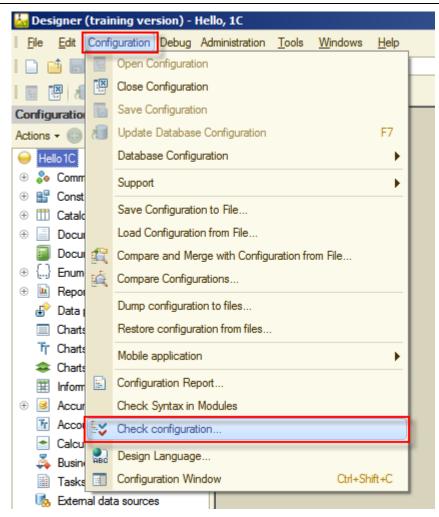


Figure 12-2. Open Configuration check

You can set the name for the current set of configuration verification options to make it easier to repeat this verification later. Let it be **Mobile**. Select following check boxes:

- Verifying configuration integrity and consistency;
- Find unresolved references;
- Mobile application client;
- Mobile application server;
- Search for unsupported functionality.

Click OK.

Configuration check X
Verification options:
Mobile
Verifying configuration integrity and consistency
Find unresolved references
Syntax check of modules (default run mode: Managed application)
Thin client
Web client
Server
External connection
External connection (client-server variant)
Mobile application - client
Mobile application - server
Thick client (managed application)
Thick client (managed application, client-server variant)
Transfer modules without source codes
Logical module check
Search for unused procedures and functions
Check whether assigned handlers are available
Search for empty handlers
Perform extended check
Search for methods that open modal windows
Mobile application checkup
Search for unsupported functionality
OK Cancel Help

Figure 12-3. The Configuration check window

The configuration will be verified for compatibility with the mobile platform. The result of this verification will be 11 errors.



Figure 12-4. The result of verification

Those errors should not make you worry. Here they are:

Hello, 1C

M	essages
	Configuration.Hello1C : Mobile application does not support multi-form desktop.
	Subsystem.Contacts : Metadata class not supported by mobile application platform.
	Subsystem.Events : Metadata class not supported by mobile application platform.
	Subsystem.Finances : Metadata class not supported by mobile application platform.
	Subsystem.Settings : Metadata class not supported by mobile application platform.
	FunctionalOption.UseEvents : Metadata class not supported by mobile application platform.
	Report.Expenses : Metadata class not supported by mobile application platform.
	Report.CashBalance : Metadata class not supported by mobile application platform.
	Report.CashBalance.Form.ReportForm : Metadata class not supported by mobile application platform.
	Report : Type not supported by mobile application platform.
	Report.DailyChart : Metadata class not supported by mobile application platform.

Figure 12-5. The list of errors

If you look carefully, you might mention that there are several unsupported by the mobile platform features used in your application. This happened because until this moment you did not take care of compatibility with mobile devices and used all features that you needed.

Now, as a simplest way, you will delete the use unsupported by the mobile platform features and configure **Mobile desktop** that will be compatible with the mobile platform.

Notice: To backup the current application, you can in the Designer mode in **Main menu** click **Administration**, and then click **Dump infobase...** In the opened dialog select a file name and click **Save**. To restore the application, in **Main menu** click **Administration**, and then click **Restore infobase...** In the opened dialog select the previously saved DT file, and click **Open**.

So, delete all subsystems, functional options, and reports.

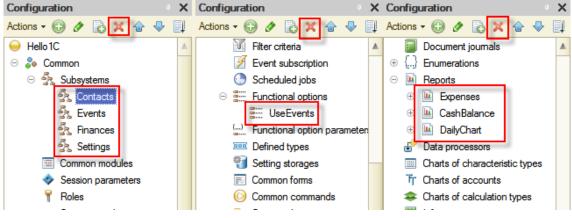


Figure 12-6. Deleting unsupported metadata objects

In **Properties** of the root of the **Configuration** tree, click **Select** ..., and in **Use purposes** list select **Mobile device** check box, and clear **Personal computer** check box. Then click **OK**.

Rapid application development tutorial

Configuration * X	Properties X
Actions - 💼 🖉 💽 🗙 🛧 🕂 🜉	21 🖬 🖉 🗙 🗸
\varTheta Hello 1C	▼General:
🕀 💑 Common	Name Hello 1C
🕀 🔡 Constants	Synonym Hello, 1C
🕀 🎹 Catalogs	Comment
Documents	
Document journals	Default run mo Managed application
⊕ {} Enumerations	Use purposes Personal computer
🛄 Reports	Use purposes X
🔄 Data processors	
Charts of characteristic types	Fersonal computer OK
Tr Charts of accounts	✓Nobile device Cancel
Charts of calculation types	Cancer

Figure 12-7. Adjusting Use purposes

Notice: Unsupported by the mobile platform application objects are disabled now.

The **Mobile desktop** of the mobile application is not configured yet. After it will be done, the application will be compatible for the mobile platform.

Create a common form.

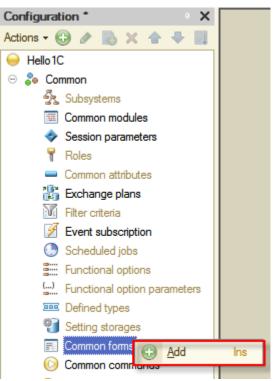


Figure 12-8. Creating a common form

Name it **MobileDesktop** and set **Synonym** to **Hello**, **1C**. The form type should stay **Generic form** as for default. After that, click **Finish**.

Common form	wizard		×		
Select form ty Generic for Constants Report for Report sel Report va	rm form m tings form				
O Dynamic li	Dynamic list settings form				
O Search for	m				
Name:	MobileDesktop				
Synonym:	Hello, 1C				

Figure 12-9. Adjusting the common form

It seems reasonable that users would like to see the most often used **People** and **Events** lists on the **Mobile desktop**. To provide this information to users, create two attributes of the **DynamicList** type. The first one will be **People**. To add the form attribute, in the top right panel of the form editor click **Add attribute o** (Ins). In **Properties** of the new attribute set **Name** value to **People**, and then click **Select** ... of the **Type** property, then select **DynamicList**.

📰 Configuration Hello 1C: Mo	bileDesktop	
🕀 🖉 🗙 🛧 🕂 ҧ		Properties: Attribute X
	010 0 0	51 [c] 20 × ∽
🗃 Form	Attribute	▼General:
Command bar		Name People
	⊕ 🎹 People	Title People
		nuo reopie
		Type DynamicList
		Arbitrary Main a Number of Arbitrary
		Number
		View
		Edit
		Functi
	B Attachistory	UUID
Eleme Comm	Attributes	Objet DynamicList
	1	Custo Value Table
		Dynar

Figure 12-10. Adding the People attribute of the common form

For the **MainTable** property, click **Select** ..., then in **Select table** select the **People** catalog, and then click **OK**.

Rapid application development tutorial

Properties: Attribute	• ×	Select table	×
₽Į 📑 🎦 × 🗸		Table	
▶ General:		🕀 🔡 Constants	
▶ <mark>Use:</mark>		□ □ Catalogs □	
✓Object:		Contact Types	
CustomQuery DvnamicDataRe		III People	
MainTable	×	PeopleRelation Types EventCategories	
AutoSaveUserS 🗸		Events	

Figure 12-11. Selecting the main table of the dynamic list

In the same way, add the **Events** attribute. With the only difference, the **MainTable** property will be the **Events** catalog.

Configuration Hello 1C: Mo	bileDesktop		
🖯 🖉 🗙 🚖 🖶 📑			Select table X
	0 0 0 0	21 🖻 🔟 × 🗸	Table
🗃 Form	Attribute	▼General:	
Command bar		Name Events	🕀 🔡 Constants
	⊕ III Events		
	People	Title Events	Contact Types
			T People
			PeopleRelationTypes
		Type DynamicList	EventCategories
		Mar and a Co	III Events
		Main attribute	Documents
		▼Use:	
		View <u>Open</u>	⊕ {} Enums
		Edit <u>Open</u>	⊕ 🧾 Accumulation Registers
		Functional optio	
Eleme 🗧 Comm	🗧 Attributes	▼Object:	
	1	CustomQuery	
		DynamicDataRe 🗸	
All actions 👻 📀		List setupOpen	
Ŭ	1	Main Table 111 Catalog.Events	OK Cancel
·			
		Auto Save UserS 🗸	

Figure 12-12. Adding the Events form attribute

Now, drag both attributes to the **Form** node on the top left panel of the form editor. You can select more than one form attribute by holding the **Ctrl** key and clicking form attributes, and then drag them all at a time to the destination.

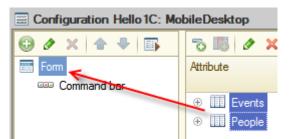


Figure 12-13. Placing form attributes on the form

On the opened confirmation dialog, click **Yes** to agree that columns of both tables will be added to the form automatically.

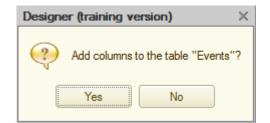


Figure 12-14. The confirmation of that columns will be added automatically

After that, you will have two tables on the form, displaying all attributes of the **Events** and the **People** catalogs.

Delete not required columns from the form. For that, in the **People** table holding the **Ctrl** key select **PeopleCode**, **PeopleGender**, and **PeopleRelationType** and then click **Delete current item** × (Del).

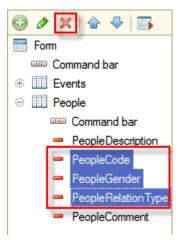


Figure 12-15. Deleting not required columns from the People table

From the **Events** table delete the **EventCode** and the **EventCategory** columns.

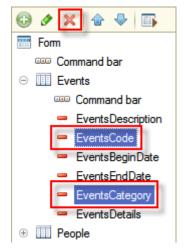


Figure 12-16. Deleting not required columns from the Events table

Open the **Starting page work area** editor (see Figure 8-19). For that, right-click the root node of the **Configuration** tree and then click **Open start page work area**. Delete the **Financial transactions** accumulation register list form from the **Left column** and the **People** catalog list form from the **Right column**. In **Starting page template**, select **One column**. After that, add the **MobileDesktop** common form to the list of forms.

Starting page work area		_ 🗆 ×
Starting page template: One column		-
🕀 🔂 🗙 🗁 🍣		
Form	Height	Visibility
	10	✓ <u>Identical</u>
Select managed form		×
🖂 😓 Common	ОК	
 □ Common forms □ Mobile Desktop □ Catalogs 	Cancel	
⊕		

Figure 12-17. The single MobileDesktop form on the Start page

Now, you are done with the configuration changes. To save them press the **F7** key. You can check the configuration again (see Figure 12-2), this time there will be no errors found.

The last step is to upload and start this application on the mobile device. You will use the tool that is similar to publication on the web server that was described in Web client chapter on page 126.

The web server will keep the application as a single XML file. The mobile platform for developers connects to this web server, downloads that XML file and installs on the mobile device. After that, the application can be started on the mobile device.

In the Designer mode in **Main menu** click **Configuration**, then click **Mobile application**, and then click **Publish...**

Hello, 1C

🛃 Designer	(traiı	ing version) - Hello, 1C		
<u>F</u> ile <u>E</u> dit	Confi	guration Debug Administration <u>T</u> ools <u>W</u> indows	<u>H</u> elp	
I 🗋 📫 📕	4	Open Configuration		📃 💌 🕿 🕿 🖄 🖉 🖉
		Close Configuration		
Configuratio		Save Configuration		
Actions -	1	Update Database Configuration	F7	
\varTheta Hello 1C		Database Configuration	•	
😑 🝰 Comm		Support	•	
옷 S 🗐 C		Save Configuration to File		
→ S		Load Configuration from File		
₽ R	£	Compare and Merge with Configuration from File		
- C	цó.	Compare Configurations		
🏭 E M R		Dump configuration to files		
E E		Restore configuration from files		
	<u></u>	Mobile application	•	Record to file
8 F			-	
() Fi	8	Configuration Report		Update published application
D Balance		Check Syntax in Modules	L	Publish

Figure 12-18. Opening Mobile application publishing

In the opened window select the **Create virtual directory on web server** and **Update mobile application when a database configuration is updated** check boxes, and input **hello_1c_mob** in **Name**. See that the last folder in **Directory** is also changed to **hello_1c_mob**. Then, click **Publish**.

Mobile application publishing X				
Create virtual directory on web server	Publish			
Name: hello_1c_mob	Disconnect			
Web server: Internet Information Services	Close			
Directory: C:\inetpub\wwwroot\hello_1c_mob}	Help			
Update mobile application when a database configuration is updated				
Use operating system authentication on web server				

Figure 12-19. Publishing the mobile application

Confirm all actions that the platform will ask confirmation for, and your application will be published.

To install and start the application on iOS devices in the developer mode you will need the developer account. Moreover, you will need a Macintosh computer with XCode developer environment.

For Android devices, you only need to install 1C:Enterprise mobile platform for developers using packages that are available in the distribution kit of 1C:Enterprise mobile platform.

For this, upload one of APK installation packages to the mobile device, for example using USB cable. These files are available in **Android** folder of the distribution kit. Install **1cem-arm.apk** if you have a device with ARM

processor (suits for most devices) or **1cem-x86.apk** if you have a device with x86 processor.

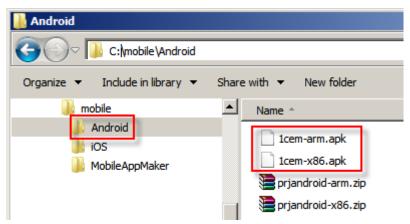


Figure 12-20. Mobile application for developer installation packages for Android Then, run the installation package.

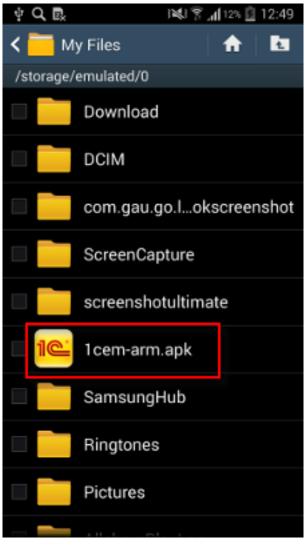


Figure 12-21. Running the installation package

It will require an access to several system features.

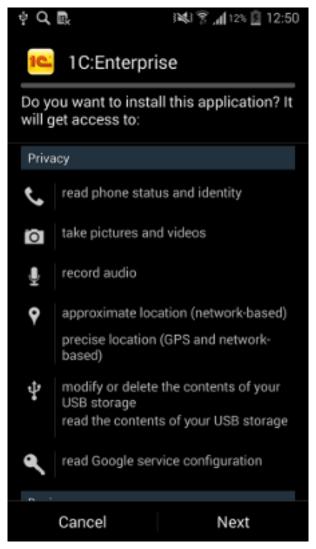


Figure 12-22. Starting the installation

After that, it will install in a few moments.

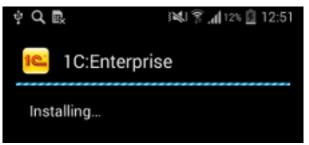


Figure 12-23. Installing the mobile application

When you will have the mobile platform for developers installed, tap **1C:Enterprise** icon to start it.

Rapid application development tutorial



Figure 12-24. Starting the mobile platform for developers

If you start the mobile platform for the first time, it will show you an empty list of applications. To add a new application, tap **Add application**.

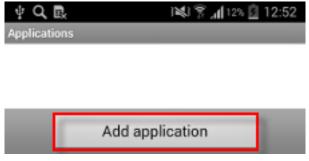


Figure 12-25. Adding a new mobile application

In **Address** input **http://<your computer IP>/hello_1c_mob**. Instead of *<your computer IP>* insert IP of the computer where you have the web server with published mobile application.

Notice: To find out the computer IP, see its network connection properties. The mobile device and the server should be in the same network. The easiest way of doing that is using a Wi-Fi router. Both, the computer with web server, and the mobile device should be connected to it.

Then, tap **Download**.

ψQ∞	- B.	12:5 🗋 12 🕅 الم	53
Address:			
http://10).70.1.39/hello_1c	_mob	
User nan	ne:	_	
Passwor	d:		
	Down	oad	

Figure 12-26. The mobile application URL

The mobile platform will download the application.

Downloading	
0%	0/100

Figure 12-27. Downloading the mobile application

Then, the mobile platform will suggest you to adjust some properties. Leave **Application name** as it is and select the **Restart from designer** check box. Then, tap **Done**.

4 Q 🗈	j 💐 🍞 📶 12% 📋	12:59
Application	name:	
Hello, 1C		
Restart fro	m designer:	~
	Done	

Figure 12-28. The mobile application startup parameters

The application will be installed.

Installing	
0%	0/100

Figure 12-29. Installing the mobile application

You will see the application in the list of applications. To start it, tap the application name.

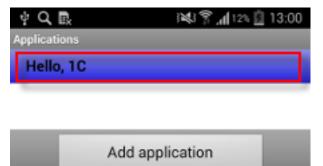


Figure 12-30. Starting the mobile application

The first screen that you see after that is the **Mobile desktop** form.

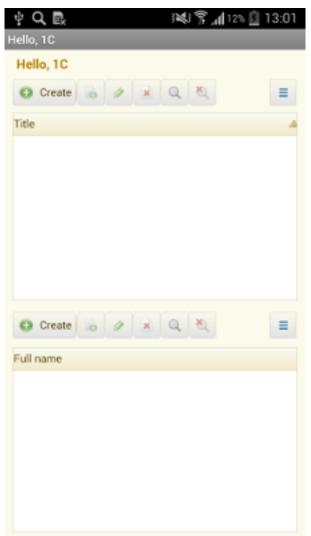


Figure 12-31. The Mobile desktop form

If you tap menu button of your mobile device you will see the application menu. Notice that the command interface of the **Main section** is displayed in the application menu in mobile platform.

ψ́Q, 🛃 🕺 🛱 📶 12% 🙆 13:0	1
Functions Desktop]
People	
Cash payment	1
Cash receipt	
Contact types	
Event categories	
Events	
People relation types	
See also	
Financial transactions	
Create	
Person	
Event	
Cash receipt	
Cash payment	

Figure 12-32. The mobile application menu

The important notice: the application that you developed and filled with data, when opened on the mobile device contains no data. This is not an error. The data synchronization between several applications is an interesting, but separate topic. The 1C:Enterprise platform can solve these problems with ease. For an example, see Homework 3 of 1C mobile application rapid application development tutorial.

In this chapter spending only a few minutes you developed a mobile version of your CRM application, which is compatible with Android and iOS mobile devices.

To distribute your application for end users you need to create a solid package containing the mobile platform, the applied solution, and the database. For this, 1C:Enterprise platform has packing tools. You can read more about it in Mobile application building example chapter of 1C mobile application rapid application development tutorial. The packed application can be published in Google Play and Apple App Store.

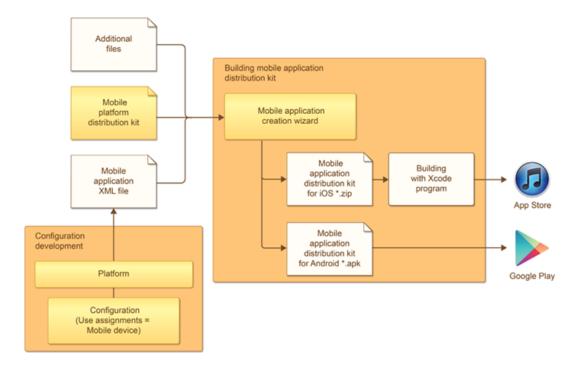


Figure 12-33. Publishing the mobile application for end users

Where and how to study 1C:Enterprise

The experience of many professionals prove that it is easy to successfully master 1C:Enterprise on your own. To help you with that, 1C Company publishes training versions of the platform, demo versions of applications, documentation, books, and tutorials. They are available free of charge on 1C:Developer Network.

1C:Enterprise 8 (training version)

1C:Enterprise 8 (training version) is an affordable solution for a wide range of users who want to get started with 1C:Enterprise 8. It allows you to learn development methods: create and edit the metadata structure, write script modules, configure forms and interfaces, and administrate the application.

You can also use the training version to modify existing applied solutions, keeping in mind its limitations. Applied solution configurations are the same for both demo and commercial versions. There are no limitations on the configuration complexity in the training version. However, the training version cannot be used for keeping of actual records, for this you need to purchase a commercial version of 1C:Enterprise platform.

Training version limitations are listed below.

- The volume of data is limited for documents, catalogs, registers, charts of accounts and other object tables:
 - Maximum number of records in account tables is 2000.
 - Maximum number of records in main object tables is 2000.
 - Maximum number of records in tabular sections of objects is 1000.
 - Maximum number of records in record sets is 2000.
 - Maximum number of records from external data sources is 200.
- Client/server mode is not supported.
- Distributed infobases are not supported.
- COM connection is not supported.
- It is not possible to use passwords and operating system authentication.
- Printing and saving spreadsheet documents are only supported in the Designer mode.
- Performance is lower compared to the commercial 1C:Enterprise 8 version.
- It is not possible to copy content of multiple cells of a spreadsheet document in the 1C:Enterprise mode.
- Operations with the configuration repository are not supported.
- Configuration delivery features are not available.
- Only one concurrent infobase session in possible.

To get 1C:Enterprise (training version) support, use the public Studying 1C:Enterprise platform forum.

You can download updates for 1C:Enterprise (training version) free of charge on 1C:Developer Network.

1C:AccountingSuite demo

1C:AccountingSuite is a small business accounting and inventory software. The applied solution supports US GAAP and IFRS accounting, and reporting standards. The software provides full visibility into purchasing and sales transactions, automates accounts receivable and accounts payable. 1C:AccountingSuite makes managing payments, receipts, and bank transactions a breeze, providing business owners with a real-time financial snapshot of operations.

The demo version lets you:

- emulate accounting of the live business;
- learn accounting and taxes calculation;
- learn how to adjust documents and reports;
- estimate economy on the manual labor cost;
- generate reports, including Balance sheet and Profit and loss statement;
- adjust and adopt the applied solution.

The demo version is not intended to be used in live business accounting according to limitations of 1C:Enterprise (training version) (see page 149).

The demo version includes:

- 1C:AccountingSuite applied solution;
- 1C:AccountingSuite user manual PDF book;
- 1C:Enterprise (training version);
- Installation instructions.

You can find this application description, download files and support in 1C:AccountingSuite section of 1C:Developer Network.

Business automation

1C Company has built a software development, distribution, and implementation business, which has enabled thousands of companies to start and grow profitable business. By automation of management and accounting fields 1C Company provided hundreds of thousands of companies and individuals with an access to the mission-critical information.

1C Company created a prestigious profession and even industry. Hundreds of thousands of competent professionals are helping companes and individuals to improve their efficiency by using the state-of-art 1C:Enterprise automation platform (see page 151).

Most of employees of 1C Company and its partners graduated from universities and colleges. Therefore, one of 1C Company priorities, as the leading software developer, is to cooperate with students and educational institutions to give young people the opportunity to gain practical skills of working with 1C:Enterprise platform. As well as to assist in employment to the position that allows to utilize acquired skills.

1C:Enterprise 8

1C:Enterprise 8 application collection is designed for the automation of management and accounting based on the state-of-art 1C:Enterprise functionality, platform. The platform provides extensive flexibility, and scalability starting from single user applications in file mode and until cloud applications with server clusters deployment. Features and architecture of 1C:Enterprise 8 platform is designed in anticipation of global trends in the business automation. Many of 1C:Enterprise solutions are unique and are beyond the competition.

1C Company (see page 154) and its partners (see page 152) using 1C:Enterprise 8 platform develop mass-market, industrial, and custom business solutions, including following:

- 1C:AccountingSuite (see page 150)
- 1C:Small Business
- 1C:Translator

1C:Enterprise 8 offers:

- To Director of Development: tools for analysis, planning, and flexible resource management, that gains its competitive advantages.
- To Department Director, managers, and employees who are directly involved with production, marketing, logistics, and other activities: tools to gain the efficiency of their daily work.
- To accounting department: tools for automated record keeping that is in full compliance with both the legal requirements and the corporate standards of the company.
- To IT professionals: tools and environment for the development, modification, deployment, administration, and maintenance of corporate information systems that meet latest standards.

Mass-market, industrial, and custom business solutions powered by 1C:Enterprise let you:

- Find the optimal automation strategy.
- Implement the application spending minimum time and resources.
- Quickly get real benefits from the implementation.
- Simplify the user training, maintenance, and administration of the application.
- Develop the system in accordance with the needs of the company without any downtime.

All of this ensures the high efficiency of 1C:Enterprise specialists and is a foundation of their success on the automation market.

1C partners

International experience shows that bulk sales of technically complex products can only be effectively managed through a well-organized and diversified dealer network.

1C partners are underlying distribution and technical support centers. Clients perceive 1C partners to be representatives of 1C Company that work side by side with them. If one of users finds it difficult to use a new application to prepare an urgently needed Balance sheet or P&L, for example, he would want to be able to obtain sound advice as soon as possible. The fastest way to get such advice is from a local partner.

1C Company always strives to understand and take account of the interests of its partners, not just focus on the benefits for itself. 1C Company has a direct interest in growing total sales through the partner network and consequently tries to provide its partners with the best terms and conditions for cooperation and provide all possible manner of support. This has enabled 1C Company to attract more than 10,000 regular partners in more than 800 cities in Eastern Europe.

1C partner program consists of several levels. **1C:Official Partner** introductory status was established for partners who have not developed their own applied solutions powered by the 1C:Enterprise platform and need to acquire 1C:Enterprise licenses to implement projects for their clients. **1C:Solution partner** status is intended for partners who have an applied solution, which has been certified by 1C Company. The **1C:Service provider** status is established for those who are planning to organize and provide services based on products created on 1C:Enterprise platform. **1C:Regional partner** status is intended for partners who have their own certified applied solutions and sell them through their own regional partner network.

More details on how you can become a 1C partner you can read in Partner program information section of 1C:Developer Network.

Useful online resources

1C:Developer Network (http://lc-dn.com)

1C:Developer Network is the information resource for developers, creating business solutions on 1C:Enterprise platform. 1C:Developer network is a good choice for both novice and experienced developers. It is possible to learn from grounds and create a business solution by using 1C:Developer network documentation and support.

Everyone can create an account and use the materials provided on 1C:Developer Network without any fee. To obtain free of charge all needed tools for learning and beginning of development. Each user can ask a question or share his ideas and experience in 1C:Enterprise at the online forum on 1C:Developer Network.

On the website you can find up-to-date information on all aspects of 1C Company activities:

- Support
- List of 1C partners
- News for partners and users
- Information on training and certification
- Press information
- Price list
- Jobs

To help you continue learning 1C:Enterprise use following references:

- 1C:Enterprise platform documentation: user, administrator and developer guides.
- Learn section allows you to study 1C:Enterprise platform using samples and tutorials.
- 1C:Developer Network forums for developers, as well as users and partners.
- 1C:Enterprise platform 1C:Enterprise overview and key features.
- Best practices section contains standards and best practices for 1C:Enterprise developers.
- Demo applications for studying specific features of 1C:Enterprise platform.
- Video section is a collection of training videos for quick and easy 1C:Enterprise learning.
- Applications section is a collection of advanced applications based on 1C:Enterprise platform.
- System requirements contain a list of supported OS and DBMS.

About 1C Company

1C Company was established in 1991, for the purposes of software development, distribution, publishing, and support of computer applications and databases for business and home use. 1C Company works with clients through more than 10,000 partners in 600 of 23 countries to provide integration services for the automation of businesses.

Apart from 1C:Enterprise, the best-known 1C Company business lines are software products for home use and educational applications.

The 1C:Enterprise collection of applications is used daily by several million users in business and government to automate operations, accounting, finance, HR, and management activities. 1C Company provides an array of vertical solutions for manufacturing, distribution, and service businesses. With its innovative 1C:Enterprise platform, and a range of other applications, 1C Company has achieved wide popularity for its openness, fast modifications and software updates. 1C:Enterprise is a very flexible and scalable platform, which meets the needs of companies ranging in size from a single user to hundreds of users. 1C Company is the market leader in enterprise automation in Russia, Ukraine, Kazakhstan, and Belarus and is used widely in the global market.

For references visit: 1C:Developer Network.