FrontBase® A WebObjects® Application



Because of last-minute changes to FrontBase, some of the information in this manual may be inaccurate. Please read the Release Notes on the FrontBase distribution for the latest up-to-date information.

Revised: 02/17/05 gab

FrontBase copyright ©2000 by FrontBase Inc. and its licensors. All rights reserved.

Documentation within these pages may be printed by licensee for personal use. Except for the foregoing, no part of this documentation may be reproduced or transmitted in any form by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from Front-Base Inc.

FrontBase and the FrontBase logo are registered trademarks of FrontBase Inc.

All other trademarks and registered trademarks are the property of their respective owners.

ALL SOFTWARE AND DOCUMENTATION ON THE COMPACT DISK(S) ARE SUBJECT TO THE LICENSE AGREEMENT IN THE CD BOOKLET.

U.S.A. and international	FrontBase Inc.
	27042 Towne Center Drive
	Suite 265
	Foothill Ranch, CA 92610
	FrontBase Inc. Datavej 52 DK-3460 Birkerød Denmark
Ordering	Voice: +1 949 455 2913
	Fax: +1 949 457 0966
	Voice: +45 4582 6262 Fax: +45 4582 0816
World Wide Web	Voice: +45 4582 6262 Fax: +45 4582 0816 http://www.frontbase.com
World Wide Web Technical support	Voice: +45 4582 6262 Fax: +45 4582 0816 http://www.frontbase.com support@frontbase.com
World Wide Web Technical support Information	<pre>Voice: +45 4582 6262 Fax: +45 4582 0816 http://www.frontbase.com support@frontbase.com info@frontbase.com</pre>
World Wide Web Technical support Information Sales,marketing	<pre>Voice: +45 4582 6262 Fax: +45 4582 0816 http://www.frontbase.com support@frontbase.com info@frontbase.com sales@frontbase.com</pre>

How to Contact FrontBase:

Table of Contents

Α	A WebObjects Application		5
	Overview	• •	5
	Package Installation.	• •	5
	Creation of the Contacts Database.	• •	. 12
	FBDatabaseManager	•	. 12
	Table Creation	•	. 14
	EOModeler	•	. 14
	EOAdaptor Connection Dictionary Keys	•	. 24
	Enterprise Object Model Creation	•	. 26
	The Company Table	• •	. 26
	Add the Table Attributes	• •	. 28
	Creating Relationships	•	. 34
	WebObjects Application Creation	•	. 44
	ProjectBuilder	•	. 44
	Create the Contact Entry Page	• •	45
	Create the ProductEntry Page	• •	. 54
	Create the CompanyEntry Page	• •	61
	Create the ContactDetails Page	• •	. 68
	Linking the Pages Together	•	. 71
	Create the Main Page Hyperlink	• •	. 77
	Testing Navigation	• •	. 78
	Contact Details Page	• •	. 79
	Connect the Table to the Component Page	• •	. 89
	Additions to the ContactDetails Page	• •	. 111
	Passing of Contact Objects.	• •	. 128
	Running the WOContacts Application	•	. 130
	Building the Application	• •	. 130
	Running the Application	•	. 131
	Adding Contacts	•	. 132
	Adding Products	• •	. 133
	Adding Companies.	• •	. 135
	Displaying Company Details	• •	. 137

	Summary	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	. 1	40
Index																						14	11

Α

A WebObjects Application

Welcome to the Fun part ! This chapter will guide you through a complete process from where to get FrontBase for FREE, through the database model and table definition/creation, and web application development.

This chapter will guide you through the following:

- <u>"Overview" on page 5.</u>
- <u>"Package Installation" on page 5.</u>
- <u>"Creation of the Contacts Database" on page 12.</u>
- <u>"Enterprise Object Model Creation" on page 26.</u>
- <u>"Table Creation" on page 14.</u>
- <u>"WebObjects Application Creation" on page 44.</u>
- <u>"Running the WOContacts Application" on page 130.</u>

Overview

FrontBase is available for FREE download at www.frontbase.com.

Package Installation

1. After decompressing the downloaded package, you will have a file FrontBase-MacOSX-<version>.mpkg as seen in Figure A.1



Figure A.1 FrontBase-MacOSX-<version>.mpkg

2. You are now presented with a window as seen in Figure A.2, showing you the components that will be installed. Notice the checkboxes are all checked. Since this is our first install, we leave them all checked.



Figure A.2 Components to be Installed

3. Click the Install button and you will be prompted to confirm the installation. To confirm the installation click the "Continue" button as seen in Figure A.3.



Figure A.3 Confirm Installation

4. If you already have a version of FrontBase installed, you will be informed that some of the files are already installed as seen in Figure A.4. That is ok, click on the "Continue" button.



Figure A.4 Files Have Been Found

5. The installer will now proceed to install FrontBase on your computer. When installation is complete you will see the panel as shown in <u>Figure A.5</u>.

	FrontBase SQL 92	
	Installation completed.	
-		-
escription:	Disk space require	ed: 5 MB
This package c server. Install (contains the software for the FrontBase SQL92 database only on computers running MacOS X Server.	
	Package Options	nstall

Figure A.5 Installation Complete

6. The frontBase engine, FBExec is started automatically as part of the installation process. To verify that it is open a terminal window and execute the following command, as seen in <u>Figure A.6</u>.

ps axc | grep FBExec



Figure A.6 FBExec Verification

NOTE: To open a terminal window, from the Workspace Manager navigate and double click /System/Administration/Terminal.app.

If for some reason for some reason the FBExec is not running, try to launch it from the same Terminal window by using the following command:

/Local/Library/FrontBase/bin/FBExec &

If it still does not start, please e-mail all the messages to: support@FrontBase.com.

NOTE: It is normal to see a single start-up message upon manual start-up.

Creation of the Contacts Database

FBDatabaseManager

1. Create a FBDatabaseManager shortcut. From the Workspace Manager navigate to /Local/Applications/FBDatabaseManager.app as seen in Figure A.7



Figure A.7 Create a FBDatabaseManager shortcut

2. Click and drag the FBDatabaseManager.app to your Desktop. Now start-up FBDatabaseManager by double clicking on this icon as seen in Figure A.8.



Figure A.8 Starting FBDatabaseManager

3. From the Menu choose Database->Create and you will be prompted for Host name and Database name. Enter your host, click "Add" and enter Contacts for the Database name, click "Select" as seen in Figure A.9.

Create d	latabase
Host	ethan
ethan 🖻	Contacts
Host name: othan	
Filoschane. Ethan	
Database name: Con	tacts
	Cancel Select

Figure A.9 Create Contacts

4. Your database has been created and it will now be started and displayed in the Monitor as seen in Figure A.10.





Table Creation

EOModeler

1. Create an EOModeler shortcut. From the Workspace Manager navigate to /System/Developer/Applications/EO-Modeler.app as seen in Figure A.11.



Figure A.11 Create an EOModeler Shortcut

2. Click and drag the EOModeler.app to your Desktop. Now start-up EOModeler by double clicking on this icon. Choose Model->New from the menu and choose FrontBase for the adaptor to be used for your application, as seen in Figure <u>A.12</u>.

	New Model Wizard
Choose the adapt	or for your database.
Available Ad	aptors
FlatFile	
FrontBase	
LDAP	
None	
OpenBaseLite	
Oracle	
Cancel	<pre></pre>

Figure A.12 Starting EOModeler

3. Click the "Next" button and enter Contacts for the Database name, your hostname for the Host name and click "Login" as seen in Figure A.13.

Front Base Login FrontBase
Database name: Contacts
Host name: ethan
Database password:
User name:
Cancel Login

Figure A.13 Define the Database

4. You will then be prompted to what to include in you database EOModel, click "Finish" as seen in <u>Figure A.14</u>.

	New Model Wizard
Choose what to	include in your model.
Assign primary keys to all entities	A primary key is a column or combination of columns whose values are guaranteed to uniquely identify each row in a database table.
☑ Ask about relationships	Allows you to specify referential integrity rules for relationships.
Ask about stored procedures	Allows you to choose stored procedures defined in the database (if any) to include in your model.
☑ Use Custom Enterprise objects	Maps custom enterprise object classes to entities in the model. The wizard assumes that the custom classes have the same name as their entities. (You have to create the classes yourself.)
Cance	el (Back Next) Finish

Figure A.14 Defining the Model

 It is a good idea to save your model now.
 Choose Model->Save and click on Library and enter Models/Contacts in the Name field as seen in Figure A.15.



Figure A.15 Saving the Model

6. You might be prompted for a Nonexisting path if you do not have a Models Folder. Click "Create" to create the Folder and model file as seen in Figure A.16.



Figure A.16 Path Creation

7. Your model should now reflect the new database name "Contacts" as seen in <u>Figure A.17</u>.



Figure A.17 The Contacts EOModel

To start creating Entities in EOModeler,

8. Choose Property->Add Entity your window should look like <u>Figure A.18</u>.



Figure A.18 Creating Table Entities

9. Under the "Name" column, double click the Entity entry and replace it with Contact, double click in the field under the "Table" column and enter CONTACT, finally double click EOGenericRecord under the "Class Name" column and enter Contact as seen in Figure A.19.



Figure A.19 Record Creation

10.Repeat the previous 2 steps for the tables Company and Product so your model looks like <u>Figure A.20</u>.



Figure A.20 The Complete Contacts EOModel

11.Click Model->Save to save you model. You will see a Consistency Check, Figure A.21 stating that your tables do not have any primary keys defined. Click OK, because you are now going to define your tables in the next section.



Figure A.21 Consistency Check

EOAdaptor Connection Dictionary Keys

The following keys are known the the FrontBase EOadaptor. The keys are case sensitive and the values of the databaseName has the case sensitivity of the underlying file system.

 databaseName 	NSString	The name of the database
 userName user, default the l 	NSString login name	The database name of the
 hostName faults to current h 	NSString 10st	The name of the host, de-
 databasePasswor password, defaul 	d NSString t none	The digested database
 password for the user, defay 	NSString ult none	The digested password
 debugFile into, opened apped 	NSString end write	A file to write debug info
– debug	NSNUmber	The debug level
0: off		
1: on		

- catalogName NSString The catalog name, defaults to current database
- schemaName NSString The schema name, defaults to the default schema for the user
- isolationLevel NSNumber The isolation level, defaults to SERIALIZABLE
 - 0: ISOLATION LEVEL READ UNCOMMITTED
 - 1: ISOLATION LEVEL READ COMMITTED
 - 2: ISOLATION LEVEL REPEATABLE READ
 - 3 :ISOLATION LEVEL SERIALIZABLE
 - 2: ISOLATION LEVEL VERSIONED
- updateability NSNumber The updatability, defaults to READ WRITE
 - 0: READ WRITE
 - 1: READ ONLY
- lockingDisciplineNSNumber The locking dicipline, defaults to pessimistic
 - 0: LOCKING PESSIMISTIC
 - **1: LOCKING DEFERRED**
 - 2: LOCKING OPTIMISTIC
- firstBatchSize NSNumber The number of rows fetched in the first batch, default 200
- nextBatchSize NSNumber The number of rows fetched in subsequent batches, default 200

The passwords must be digested. This is done with the Class method of the class FBPasswordDigest something like:

[FBPasswordDigest digestPassword:pw user:@"_SYSTEM"];

NOTE: The pw is the password that the user entered, and the user is the uppercase user name. For the database password you must use _SYSTEM and the user.

FBPasswordDigest is defined in the FBAccess framework, in the header file FBPasswordDigest.h

Enterprise Object Model Creation

The Company Table

1. The SQL for the table will be as follows:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE, LOCKING
PESSIMISTIC;
DROP TABLE "Company" CASCADE;
CREATE TABLE "Company"
 "Address1" VARCHAR(255),
(
  "Address2"
               VARCHAR(255),
  "City"
               VARCHAR(255),
  "Contact_Id" INT,
  "Country"
               VARCHAR(255),
  "Counciy
"Email" VARCHAR(04),
VARCHAR(255),
  "Product_Id" INT,
  "State"
               VARCHAR(255),
  "Telephone" VARCHAR(64),
  "Zip"
               VARCHAR(64)
);
DELETE FROM INFORMATION_SCHEMA.EOMODELER_RELATIONS WHERE
"SCHEMA_NAME" = CURRENT_SCHEMA AND SOURCE_TABLE = 'COMPANY';
INSERT INTO INFORMATION_SCHEMA.EOMODELER_RELATIONS VALUES
(CURRENT_SCHEMA, 'toContact',0,0, 'COMPANY', '(CONTACT_ID)', 'CONTAC
T', '(CONTACT_ID)');
INSERT INTO INFORMATION_SCHEMA.EOMODELER_RELATIONS VALUES
(CURRENT_SCHEMA, 'toProduct', 0, 0, 'COMPANY', '(PRODUCT_ID)', 'PRODUC
T', '(PRODUCT_ID)');
```

```
ALTER TABLE "Company" ADD PRIMARY KEY
("Contact_Id", "Product_Id") NOT DEFERRABLE INITIALLY IMMEDIATE;
ALTER TABLE "Company" ADD FOREIGN KEY ("Product_Id") REFERENCES
"Product" ("Product_Id") DEFERRABLE INITIALLY DEFERRED;
ALTER TABLE "Company" ADD FOREIGN KEY ("Contact_Id") REFERENCES
"Contact" ("Contact_Id") DEFERRABLE INITIALLY DEFERRED;
COMMIT;
```

- 2. To create the Company Table click on the "Company" entry in the left frame of EOModeler and make sure that you see the following columns:
 - Name
 - Value Class
 - External Type
 - Width
 - Column
 - Value

as seen in <u>Figure A.22</u>. If you do not have all the columns, choose the "Add Column" drop down button in the Company Attributes section to add the desired column. You may have to add the "Value" column for example.



Figure A.22 Company Table Creation

Add the Table Attributes

1. To add attributes to the Company table you will choose Property->Add Attribute and your window will look like <u>Figure A.23</u>.

	Contacts.eon	nadeld/Librar	w/Medels			88
EMBXAX	うのの回避に	6660	1			
	Company Attributes					
- Carany	- + a Name	Value Class	External Type	Width	Calaran	Value
-O Contect	+ a Atribute					
Stored Procedures						
	Att Crime		- 7767			
	Company Relationships			-		
	+ Name	Destination	Source Att	Dest.	MX:	
						- 11
1	Atlution -					

Figure A.23 Adding Company Attributes

2. The Company table will have a relationship with the Contact and Product table so you will create a key to hold the ID's of each of the table data. To create the key entry first click under the "key" icon, next click under the "diamond" icon to deselect the column, next double click the "Attribute" entry under the Name column and enter contactID, double click under the Value Class and enter NSNumber, next choose INT for the External Type column, double click under the Column column and enter CONTACT_ID, next double click and finally enter i under the Value column. Repeat this for the productID so that your Company table looks like Figure A.24.



Figure A.24 Adding the ContactID Key

3. The rest of the fields will be of NSString Value Class and VARCHAR External Type. I will step you through the name and you will do the same steps for all of the fields in the Company table. Once again choose Property->Add Attribute from the menu. Double click the "Attribute" entry under the Name column and enter name, double click under the Value Class and enter NSString, next choose VARCHAR for the External Type column, double click under the Width column and enter 255, double click under the Column column and enter NAME. Repeat this for all the rest of the attributes defined above so that your Company table looks like Figure <u>A.25</u>.

Contaits	Congary /	dtibules	and the second second				
Congary.		Name	Value Class	External type	Witth	Column	Value
E-O Contact		addresst	NSShing	YARCHAR	265	ADDRESS1	
E O Product		address2	NSString	VARCHAR	255	ADDRESSz	
Stored Procedures		city	NSString	VARCHAR B	255	OTV	
	P P	contactiD	NSNumber	INT 2	1	CONTACT_ID	8
		country.	NSString	VARCHAR .	255	COUNTRY	
		enal	NSString	VARCHAR 3	84	EMAIL	
		name	NSString	YARCHAR 3	255	NAME	
		preductio	NSNumber	INT .	1	PRODUCT_ID	i.
		state	NSString	YARCHAR 3	255	STATE	
		belaphone	NSShing	YARCHAR :	64	TELEPHONE	
		zip	NSString	VARCHAR .	84	ZIP	
	And Case	an_ +		-	_		
	the second se	the second second second					
	Lingsry I	ane	Destination	Saurce Att	Dest	ATT	_

Figure A.25 Adding the Name Attribute

1. The procedure for the Contact table is the same as the Company table. The table definition for the Contact table is as follows:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE, LOCKING
PESSIMISTIC;
DROP TABLE "Contact" CASCADE;
CREATE TABLE "Contact"
  "Address1"
               VARCHAR(255),
(
  "Address2"
               VARCHAR(255),
  "City"
               VARCHAR(255),
  "Contact_Id" INT,
  "Country"
               VARCHAR(255),
  "Email"
               VARCHAR(64),
  "Name"
               VARCHAR(255),
               VARCHAR(64),
  "State"
```

```
"Telephone" VARCHAR(64),
"Zip" VARCHAR(64)
);
DELETE FROM INFORMATION_SCHEMA.EOMODELER_RELATIONS WHERE
"SCHEMA_NAME" = CURRENT_SCHEMA AND SOURCE_TABLE = 'CONTACT';
INSERT INTO INFORMATION_SCHEMA.EOMODELER_RELATIONS VALUES
(CURRENT_SCHEMA,'toCompany',0,1,'CONTACT','(CONTACT_ID)','COMPAN
Y','(CONTACT_ID)');
SET UNIQUE = 1000000 FOR "CONTACT";
ALTER TABLE "Contact" ADD PRIMARY KEY ("Contact_Id") NOT
DEFERRABLE INITIALLY IMMEDIATE;
COMMIT;
```

2. After you have followed the steps to create the Contact table you definition should look like Figure A.26.

onfacte	100	10		Catiline a	Turbus Charas	Internation	Inner	Column .	-
Company			•	Nome	Annie Ours	External type	WIDTH	Column	Anthe
Contect	111-1	٠	٠	20078551	NSSting	VARCHAR	266	ADDRES51	-
mail Procedures		٠	٠	address2	NSSang	VANCHAN	255	ADDHE532	-
no r tooraa na		٠	٠	city	NSSaud	VARCHAR	288	CITY	-
		-	٠	contact/D	NSNumber	INT	-	CONTACT_ID	1
		٠	٠	country	NSSting	VARCHAR 3	255	COUNTRY	_
		٠	٠	ensi	NSString	VARCHAR	64	EMAIL	_
		٠	٠	rame	NSSting	VARCHAR 3	255	NAME	-
		٠	٠	state	NSString	VARCHAR	64	STATE	_
		٠	٠	telephone	NSString	VARCHAR	64	TELEPHONE	_
		٠	٠	2ip	NSString	VAROHAR 3	64	ZIP	
	13	Mile:	1.50	ed on ships					
		٠	N	ime	Destination	Source Att	Dest.	MH:	

Figure A.26 Contact Table Creation

1. The procedure for the Product table is the same as the Company table. The table definition for the Product table is as follows:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE, LOCKING
PESSIMISTIC;
DROP TABLE "Product" CASCADE;
CREATE TABLE "Product"
( "Name"
               VARCHAR(255),
  "Price"
               FLOAT,
  "Product_Id" INT
);
DELETE FROM INFORMATION_SCHEMA.EOMODELER_RELATIONS WHERE
"SCHEMA_NAME" = CURRENT_SCHEMA AND SOURCE_TABLE = 'PRODUCT';
INSERT INTO INFORMATION SCHEMA.EOMODELER RELATIONS VALUES
(CURRENT_SCHEMA, 'toCompany', 0, 1, 'PRODUCT', '(PRODUCT_ID)', 'COMPAN
Y', '(PRODUCT_ID)');
SET UNIQUE = 1000000 FOR "PRODUCT";
ALTER TABLE "Product" ADD PRIMARY KEY ("Product_Id") NOT
DEFERRABLE INITIALLY IMMEDIATE;
COMMIT;
```

- 2. After you have followed the steps to create the Product table your definition should look like <u>Figure A.27</u>.
- 3. Choose Model->Save to save your model. You are now going to add Relationships to your tables in the next section.

0] Co	ntacts.eom	adeid - ~/Librar	which ets				g
同局 周 X 自 X 4	DIA	10	1		delet	100				
		and the second	- APS	the contract of the contract o		<u> </u>				
Conterts		1	Tal.	hame	Value Class	Internal Des	-	WLAN	Column	Value 1
Conpary	1 F	77	÷	1000	USShine	Vabruan.		000	NEME	the second second
- Protect		1	1	100	ASSA mitor	FLOAT	-	000	PRICE	1
Stored Procedures			÷	avoiantiD	NSNumber	INT	-	-	PRODUCT ID	
Contraction and the second			-	P. C. L. L.	1.00.000.000	104.			111 Search and Land	
	H									
		800	Caluty							_
	2	roder	1740	Averters						
	T	17	IN	THE .	Destination	Saurre Ar		Deat		
1			-	HTC.	- De statettett	2000.00.000		-		_

Figure A.27 Create the Product Table

Creating Relationships

- 1. It is now time to create the relationships that will join our tables together. First click on the Contact table entry in the left frame of EOModeler. Then you should see you table definition for the Contact table.
- 2. From the menu choose Property->Add Relation as seen in Figure A.28.

Conses	and a lat	4 Name	Value Class	Internal Date	WLATE	Column	Value			
Contract	and a second	a address i	UCChing	Vabruan -	365	ADDDDSS	1.0.00			
Control Procedures		A address?	All China	VARCHAR -	200	AV/86153	-			
		· ·····	NOShine	Vapcuan a	000	CITY	-			
		 contact(1) 	APOLymphan	INT *	6.00	CONTACT ID				
		A country	A/SObice	VAR'HER	388	COUNTRY	1			
		 a anni 	AUCCHINE	VADCHAD A	64	EMAIL	-			
		 alone 	NICChine I	VARCHAR *	204	NAME	-			
		4 1550	MSSbire	VARCHAR -	154	STATE	-			
		a belephone	AUSSbine .	VARCHAR	64	TELEPHONE	-			
		 200 	NSShire	VARCHAR *	64	20	-			
	Are D Carried	Att Edan + Cardist Metionships								
		Name	Destination	Source Mt	DestAlt					
	0.00	Relationship	1							
				144						

Figure A.28 Adding Contact Relationships

3. From the menu choose Tools->Inspector to bring up the Relationship inspector. You are going to add a One-To-Many relationship to the Company table. Enter toCompany in the name, next select the Company table from the Entity list, next select the "To Many" radio button, finally select contactID from the Source Attributes list and contactID from the Destination Attributes List. Click on Connect and you should see the selections have a "dimple" to show the connection has been completed. See Figure A.29.

A WebObjects Application *Enterprise Object Model Creation*

Relationship Inspector								
Name: toCompany								
Destination								
Model:	Contacts 🗢							
Entity:	Company							
○ To One ● To Many Inner 🔷	Contact Product							
Joins Source Attributes	Destination Attributes							
address1 address2 city contactID country email name	A address1 address2 city contactID country email name ▼							
Disconnect								

Figure A.29 Defining the Relationship

4. Your model should also show the relationship. Make sure that there is a '>>' by the relationship, if not that means that you only have a one-to-one relationship. See <u>Figure A.30</u>.
| Contaits | and a la | harne | Value Close | External Iver | Watth | Calamo | Value |
|-------------------|-------------------------|------------|-------------|---------------|--------|------------|-------|
| Carpey V | | address i | JUSS king | VADCHAD I | 300 | annorse, | |
| O Product | | 4440447 | ALC STREET | VARCHAR - | 500 | ADDRESS | - |
| Stored Procedures | | rib. | NSSking | VADCHAR 4 | 165 | CITY | - |
| | | cantactitD | NSNunber | BIT R | | CONTACT ID | |
| | | country | NSSting | VARCHAR . | 255 | COUNTRY | 1 |
| | | ensi | NSString | VARCHAR A | 64 | EWAL | - |
| | | 1879 | NSShing | VARCHAR | 255 | NAME | - |
| | | ototu | NSString | VARCHAR | 64 | STATE | - |
| | | telephone | NSString | VARCHAR | 64 | TELEPHONE | - |
| | | rip | NSString | VARCHAR . | 64 | 21P | |
| | Add Color
Contact Re | n | | - | 0 | | _ |
| | * N | ine | Destination | Saurce Att | Dest. | AX: | |
| | 30 a 10 | Company | Conpany | contectiO | contac | c#D | |

Figure A.30 One-To-Many Relationship

1. For the Company table, you will have 2 one-to-one relationships. The first to the Contact table and the second to the Product table. Follow the steps that you performed in the Contact table except DO NOT select the "To Many "radio button. These are just to be one-to-one relationships. Your relationship for the Contact table should look like <u>Figure A.31</u>.

A WebObjects Application *Enterprise Object Model Creation*

Relation	Relationship Inspector				
S S 1					
Name: toContact	t				
Destination					
Model:	Contacts 😫				
Entity:	Company				
	Contact				
🖲 To One	Product				
O To Many					
Joins Source Attributes	Destination Attributes				
address 1	▲ address1 ▲				
address2 city	address2 ≡ city =				
contactID	contactID				
country	country				
name	▼ name ▼				
Disconnect					

Figure A.31 Company Relationship To Contact Table

2. The second relationship is to the Product table. Your relationship for the Contact table should now look like <u>Figure A.32</u>.

Relationship Inspector			
S			
Name: toProduct	:		
Destination			
Model:	Contacts 😫		
Entity:	Company		
	Contact		
To One	Product		
O To Many			
Inner 🜲			
Joins			
Source Attributes	Destination Attributes		
address2	▲ name		
City	price		
country			
email			
name			
productiD			
	Disconnect		

Figure A.32 Company Relationships To Product Table

1. For the Product table, you will have 1 one-to-many relationship. You are going to add a One-To-Many relationship to the Company table. Enter toCompany in the name, next select the Company table from the Entity list, next select the "To Many" radio button, finally select productID from the Source Attributes list and productD from the Destination Attributes List. Click on "Connect" and you should see the selections have a "dimple" to show the connection has been completed. See <u>Figure A.33</u>.

A WebObjects Application *Enterprise Object Model Creation*

Relationship Inspector				
ny				
Contacts 🗢				
Entity: Company Contact Product Inner				
Joins Source Attributes Destination Attributes name price productID ContactID country email name productID C				

Figure A.33 Product Table Relationships

1. If you would like to see a Diagram View of what your database looks like, Choose Tools->Diagram View and your model should look like <u>Figure A.34</u>.



Figure A.34 The Diagram View of the Relationships

1. You have now created your model, but you still have to create the actual tables in the FrontBase Contacts database. Select the Contacts entry in the left frame of the EOModeler, from the menu choose Property->Generate SQL and you will see a window as in Figure A.35. Enterprise Object Model Creation

SQL Generation Options	
Drop Database	🗌 Create Database
☑ Drop Tables	☑ Create Tables
	✓ Primary Key Constraints
🗹 Dron Primary Key Sunnort	✓ Foreign key Constraints ✓ Create Primary Key Sunnort
- or op r rinkary key support	a create r many ney support
SET TRANSACTION ISOLATION L	EVEL SERIALIZABLE, LOCKING
PESSIMISTIC;	
DROP TABLE "Company" CASCAI	DE;
DROP TABLE "Contact" CASCADE	; –
CREATE TABLE "Product" CASCADE	=;
("Address 1" VARCHAR(255)	
"Address2" VARCHAR(255)	
"Address2" VARCHAR(255), "City" VARCHAR(255),	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT,	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255),	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255), "Email" VARCHAR(64),	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255), "Email" VARCHAR(64), "Name" VARCHAR(255),	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255), "Email" VARCHAR(64), "Name" VARCHAR(255), "Product_Id" INT,	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255), "Email" VARCHAR(64), "Name" VARCHAR(64), "Product_Id" INT, "State" VARCHAR(255),	
"Address2" VARCHAR(255), "City" VARCHAR(255), "Contact_Id" INT, "Country" VARCHAR(255), "Email" VARCHAR(64), "Name" VARCHAR(64), "Product_Id" INT, "State" VARCHAR(255), "Telephone" VARCHAR(64), "Zivi" VARCHAR(64),	

Figure A.35 Generating SQL

2. Select Execute SQL and you will see an Alert for each of the tables See <u>Figure A.36</u>. This is ok because the SQL is trying to drop the table and create it again. Since the table does not exist yet, it cannot drop it thus you get the alert. Just click the "Continue" button for each of the tables.



Figure A.36 Creating the Physical Tables

Congratulations, you have now defined and created your Contacts database tables. You may quit EOModeler. You are ready to create your WebObjects Application in the next section.

WebObjects Application Creation

ProjectBuilder

1. Create a ProjectBuilder shortcut. From the Workspace Manager navigate to /System/Developer/Applications/Project-Builder.app as seen in Figure A.37.



Figure A.37 Create a ProjectBuilder Shortcut

2. Click and drag the ProjectBuilder.app to your Desktop. Now start-up ProjectBuilder by double clicking on this icon. Choose Project->New from the menu and you will choose WebObjects Application from the Project Type popup button, navigate to the folder where you want the project to be placed (I chose LocalDeveloper) and then enter WOContacts in the Name field and click "OK" as seen in Figure A.38.

New Project				
LocalDeveloper	Web0bjects			
WebObjects 🖻	Backup Movies WOContacts 1 ►			
•	Ш 🕨			
Name: W0Contacts]			
Name: Woundards				
roject type				
WebObjects ApplicationA project for creating dynamic webapplications that use the WebObjectsframework and may access a database				
	Cancel OK			

Figure A.38 Create a New Project

Create the Contact Entry Page

1. You are going to first create the main window that will be for our Contacts using the Contact table. On the first window of the WebObjects Application Wizard, choose the "Wizard" radio button and Primary Language "Java" as seen in Figure <u>A.39</u>, then click "Next".

WebObjects Application Wizard				
Choose typ	oe of assistance and primary language.			
Available Assistance				
🔿 Direct to Web	Create a full-featured application with database access.			
🖲 Wizard	Create a working Main component with database access.			
🔾 Java Client	Create a Java client application with database access.			
🔿 None	Create a basic application (with or without database access) that you design yourself.			
Primary Language				
💿 Java				
○ WebScript				
⊖ Objective-C				
	Cancel (Back Next) Finish			

Figure A.39 Create the Main Component

2. You will now choose the database model that you just created in the last section. Choose the "Open existing model file" radio button. Then choose the "Browse" and navigate to the LocalLibrary/Models/Contacts.eomodel selection as seen in Figure A.40 and click "OK".

	Web0bje	cts App	olication Wizard	
Spec map	Open	ĺ		
	Library		Models	
● Open exis Model File: [Documentation FBCPreferences FindPreferences FrontBase Indexes Models Modelsx OmniWeb		Contacts.eomodeld	owse
	Choose Model Fil	le <mark>Con</mark> t	tacts.eomodeld Cancel OK	Finish

Figure A.40 Choosing the Contacts EOModel

3. Your model selection window should appear as seen in <u>Fig-ure A.41</u>, and click "Next".

WebObjects Application Wizard	
Specify a model that defines your database-to mapping.	o-objects
🔿 Create new model	
Open existing model file	Browse
Model File: /Local/Users/geneb/Library/Models/Contact	s.eomodeld
Cancel < Back Ne:	kt > Finish

Figure A.41 Selecting the EOModel

4. Since the main page will be a search and entry for Contacts, choose the Contact table as seen in <u>Figure A.42</u>, and click "Next".

	WebObjects	Application Wizard
Choo	se the entity to use	in the Main page.
	Entities	
Contact		
Company Product		
	Cancel	<pre></pre>

Figure A.42 Select the Contact Table

5. Select "Selected Record" and "Matching Records" as seen in <u>Figure A.43</u>, and click "Next".

WebObjects Application Wizard			
Choose a layout.			
Title Criteria: Criteria: Query Item 1 Item 2 Item 3 Field 1: Field 2: Field 3: Save	 List Selected Record Table Display All Records Matching Records Paginated 10 per Page 		
Cancel	(Back Next) Finish		

Figure A.43 Choosing a Page Layout

6. It is time to select what fields will be displayed in the window. For this application double click on all of them so your window looks like <u>Figure A.44</u>, and click "Next".

WebObjects Appli	cation Wizard
Choose attributes to display.	
Don't Include:	Include:
Contact	Attributes
	<pre>>> name address1 address2 city state zip country telephone email</pre>
Cancel	Back Next Finish

Figure A.44 Choosing the Attributes To Display

- 7. For this application we are going to make the telephone number the field that will be our hyperlink.
- 8. Double click telephone as seen in <u>Figure A.45</u>, and click "Next".

WebObjects Application Wizard			
Choose an	attribute for the h	yperlink.	
Don't Include:		Include:	
Contact		Attribute	5
zip state name email country city address2 address1		<pre>>> <<</pre>	
	Cancel	< Back Next >	Finish

Figure A.45 Choosing an Attribute for the Hyperlink

9. To query the database for entered Contacts, we will use the name, telephone and email fields. Double click these fields respectively as seen in Figure A.46, and click "Finish".

	WebObjects App	ication Wizard	
Choose att	ributes to query o	n.	
Don't Include:		Include:	
Contact		4	Attributes
zip state country city address2 address1		name telephone email	
	Cancel	« Back Ne	kt > Finish

Figure A.46 Choosing the Attribute to Query On

10. You have now created the project and your Contacts entry screen. You are now in the Project Builder's Main Window as seen in Figure A.47. From here you will add more windows, build and edit all of you files and run your WebObjects Contacts application.



Figure A.47 ProjectBuilder's Main Window

Create the ProductEntry Page

1. Click on the "Web Components" entry in the left column and you will see "Main.wo", this is your main Web Component. You will now create a screen for the Product table by selecting File->New in Project from the Project Builder's menu. Click on the "Web Component" suitcase and enter ProductEntry in the Name field as seen in Figure A.48, and click "OK".



Figure A.48 Adding the ProductEntry Component

2. Make sure the "Component Wizard" and "Java" are selected as seen in Figure A.49, and click "Next".

WebObjects Component Wizard				
Choose type of assistance and primary language.				
Available Assistance				
Component Wizard	Create a working component with database access.			
🔿 Java Client	Create a Java client component with database access.			
🔿 None	Create an empty component.			
Component Language Java				
⊖ WebScript				
⊖ Objective-C				
Ca	ncel (Back Next > Finish			

Figure A.49 Using the Component Wizard

3. Choose the Product entity from the list as seen in <u>Figure</u> <u>A.50</u>, and click "Next".

	WebObjects Component Wizard				
Choo	se the entity to use f	or this component.			
	Entities				
Contact					
Company					
Product					
	Cancel	<pre></pre>			

Figure A.50 Choosing the Product Table

4. Select "Selected Record" and "Matching Records" as seen in <u>Figure A.51</u>, and click "Next".

WebObjects Component Wizard			
Choose a layout.			
Title Criteria: Criteria: Query Item 1 Item 2 Item 3 Field 1: Field 2: Field 3: Save	 List Selected Record Table Display All Records Matching Records Paginated 10 per Page 		
Cancel	Gack Next > Finish		

Figure A.51 Choosing the Page Layout

5. Double click on name and price as seen in <u>Figure A.52</u>, and click "Next".

WebObjects Component Wizard				
Choose att	ributes to display.			
Don't Include:		Include:		
Product			Attributes	
productID		>> <<		
	Cancel	< Back N	ext > Finish	

Figure A.52 Choosing the Attributes to Display

6. Double click on the name field that will be the hyperlink for the product window, as seen in <u>Figure A.53</u>, and click "Next".

WebObjects Component Wizard				
Choose an attribute for th	he hyperlink.			
Don't Include:	Include:			
Product	Attributes			
price productID	>>> <<			
Cancel	c Back Next > Finish			

Figure A.53 Choosing the Attribute for the Hyperlink

7. Double click on the name field that will be the search field for the product, as seen in <u>Figure A.54</u>, and click "Finish".

WebObjects Component Wizard				
Choose att	ributes to query o	n.		
Don't Include:		Include:		
Product			Attributes	
price productID		>> <<		
	Cancel	c Back	Next > Finish	

Figure A.54 Choosing the Attributes to Query On

8. Notice that now in ProjectBuilder you will see ProductEntry.wo under the Main.wo entry as seen in <u>Figure A.55</u>.



Figure A.55 Addition of the ProductEntry Component

Create the CompanyEntry Page

1. You will now create a screen for the Company table by selecting File->New in Project from the Project Builder's menu. Click on the "Web Component" suitcase and enter Company-Entry in the Name field as seen in <u>Figure A.56</u>, and click "OK".



Figure A.56 Creation of the CompanyEntry Component

2. Make sure the "Component Wizard" and "Java" are selected as seen in Figure A.57, and click "Next".

WebObjects Component Wizard				
Choose type	of assistance and primary language.			
Available Assistance				
Component Wizard	Create a working component with database access.			
🔾 Java Client	Create a Java client component with database access.			
() None	Create an empty component.			
Component Language Java				
○ WebScript				
⊖ Objective-C				
Ca	ncel (Back Next > Finish			

Figure A.57 Choosing the Component Wizard

3. Choose the Company table from the list as seen in <u>Figure</u> <u>A.58</u>, and click "Next".

	WebObjects Component Wizard				
1	Choose the entity to use for this component.				
Г	Entities				
	Contact				
	Company				
	Product				
	Cancel Cack Next > Finish				

Figure A.58 Choosing the Company Table

4. Select "Selected Record" and "Matching Records" as seen in <u>Figure A.59</u>, and click "Next".

WebObj	ects Component Wizard
Choose a layout.	
Title Criteria: Criteria: Query Item 1 Item 2 Item 3 Field 1: Field 2: Field 3: Save	 List Selected Record Table Display All Records Matching Records Paginated 10 per Page
Cancel	c Back Next > Finish

Figure A.59 Choosing the Page Layout

5. Double click name, address1, address2, city, state, zip, country, telephone and email as seen in <u>Figure A.60</u>, and click "Next".

WebObjects Component Wizard				
Choose att	ributes to display.			
Don't Include:		Include:		
Company		Attri	butes	
toProduct r toContact r		<pre>>> name address1 address2 city state zip country telephone email</pre>		
	Cancel	<pre></pre>	Finish	

Figure A.60 Choose the Attributes to Display

6. Double click name for the hyperlink value as seen in <u>Figure</u> <u>A.61</u>, and click "Next".

	WebObjects Com	ponent Wizard		
Choose an	attribute for the h	yperlink.		
Don't Include:		Include:		
Company			Attributes	
Zip telephone state email country city address2 address1 toProduct ► toContact ►		>> <<		
	Cancel	< Back	Next > Fin	nish

Figure A.61 Choosing the Attribute for the Hyperlink

7. Double click name for the name to query on as seen in <u>Figure</u> <u>A.62</u>, and click "Finish".

	WebObjects Com	ponent Wizard	
Choose att	ributes to query o	n.	
Don't Include:		Include:	
Company			Attributes
zip telephone state email country city address2 address1 toProduct r toContact r		name	
	Cancel	K Back	lext > Finish

Figure A.62 Choosing the Attribute to Query On

8. Notice that now in ProjectBuilder you will see CompanyEntry.wo under the Main.wo entry as seen in <u>Figure A.63</u>.



Figure A.63 Addition of the ProductEntry Component

Create the ContactDetails Page

1. It is now time to create the Contact Details page. This is the page that will show all the relationships for a selected Contact. For this we will not automatically include a database table, so all that has to be done is to create an empty component page. To do that, choose File->New in Project from the menu. Click on the "Web Components" suitcase and enter ContactDetails in the Name field as seen in Figure A.64, and click "OK".



Figure A.64 Creation of the ContactDetails Component

2. In the Component Wizard make sure that "None" and "Java" are selected and click "Finish", as seen in <u>Figure A.65</u>.

WebObjects Component Wizard						
Choose type of assistance and primary language.						
Available Assistance						
🔿 Component Wizard	Create a working component with database access.					
🔿 Java Client	Create a Java client component with database access.					
🖲 None	Create an empty component.					
Component Language Java WebScript						
⊖ Objective-C						
Ca	ncel (Back Next > Finish					

Figure A.65 Creating an Empty Component

3. Notice that now in ProjectBuilder you will see ContactDetails.wo under the Main.wo entry as seen in <u>Figure A.66</u>.



Figure A.66 Addition of ContactDetails Component

Linking the Pages Together

1. Now that all our pages are created, we will need a way to navigate from the main page to the other pages and from each of the other pages back to the main page. This is accomplished by using hyperlinks. In ProjectBuilder, double click the Main.wo entry and you should see your main page as in <u>Figure A.67</u>. If not, select Edit->Layout View from the menu.

🔲 🔄 🗋 Main.wo 🗕 ~/LocalDeveloper/WebObjects/WOContacts 🔤 🛛 🛛	IE			
	» Յ			
Query By Example segment Search for Contacts				
specify which Contacts to display below: «⊣				
rame: contactDisplayGroup.queryMatch.name				
telenhone: contactDisplayGroup.queryMatch.telephone				
contactDisplayGroup gueryMatch email				
<pre></pre>				
Main				
application >	٦			
session >				
contact				
saveChanges				
selectObject				
Edit Source				

Figure A.67 The Main Contact Page

2. Scroll down to the bottom of the page (below the separation line) and click to place the cursor on the form. From the menu select WOObjects->WOHyperlink and enter "Company Entry" to replace "Hyperlink", as seen in Figure A.68.
| 🗉 📄 Main.wo – ~/LocalDeveloper/WebObjects/WOContacts 🛛 🕮 🗄 |
|---|
| |
| email: ContactDisplayGroup.selectedObject.email
Insert/New Save to
database |
| Image: Company Entry <body> <wohyperlink></wohyperlink></body> |
| Main |
| application >
session >
contact >
Y contactDisplayGroup > |
| saveChanges
selectObject |
| Edit Source |

Figure A.68 Adding the Company Entry Hyperlink

3. From the popup button labelled "Edit Source" click and select "Add Action" and enter enterCompanies for the Name: and select Company Entry for the Page returned, as seen in <u>Figure A.69</u>.

Add Action
Name: enterCompanies Page returned: CompanyEntry Cancel Add

Figure A.69 Adding the enterCompanies Action

4. You will now see enterCompanies selected. Select and drag this entry to the "<WOHyperlink>" symbol and click action, as seen in Figure A.70.

	Main.wo - ~/localDevelone	r/Weh0hierts/W0Contacts	PE
 ● ● ● ■ 	/ U T 3 ≧ None ഈ আ ≝ ■ ■ ₽ ₽ 0 •	I = := := - ≥ I = := := - ≥ I = := := - ≥ I = := := := := := := := ::::::::::::::	シェ੶≈
email: contact	DisplayGroup.selectedObject.ema Insert/New Save to databas	11 Delete Se A	
Company Entry	action		III ▼
application session contact r contactDisplayG	directActionName disabled fragmentIdentifier href otherTagString		
enterCompanies saveChanges selectObject Edit Source	pageName queryDictionary string target Connect to new binding		

Figure A.70 Linking the Action

5. Click to the right of the "Company Entry" hyperlink (outside the entire link) and hit the "Enter" or "Return" key to place

the next hyperlink on the next line. Repeat the previous steps for the "Product Entry" page as seen in <u>Figure A.71</u> and <u>Fig-</u><u>ure A.72</u> to connect the action "enterProducts".

Add Action
Name: enterProducts Page returned: ProductEntry Cancel Add

Figure A.71 Adding the enterProducts Action

	UT3E None	= = P (≝ – ≝ Ø J ∓ ∞ ? & & © & ★ ★
	Insert/New Save to database	Delete	
Company Entry			
<body> <p><wohyper< th=""><th>action actionClass</th><th></th><th></th></wohyper<></p></body>	action actionClass		
application session contact ✔ contactDisplayGro	disabled fragmentIdentifier href otherTagString		
enterCompanies enterProducts saveChanges Edit Source	pageName queryDictionary string target Connect to new binding		

Figure A.72 Linking the Product Entry Hyperlink

6. Click to the right of the "Product Entry" hyperlink (outside the entire link) and hit the "Enter" or "Return" key to place the next hyperlink on the next line. Repeat the previous steps for the "Contact Details" page as seen in Figure A.73 and Figure A.74 to connect the action displayDetails.

Add Action
Name: displayDetails Page returned: ContactDetails
Cancel Add

Figure A.73 Adding the displayDetails Action

	🗋 Main.wo 🗕 ~/LocalDevelo	per/WebObjec	ts/W0Contacts	
🦉 🟮 🎒 B	IUT3 ENone	¢ ¶		₿∅↓ FF ↔
<u>A</u> D =	I 📴 🖽 🖬 🗷 🗹 🗹	• 📰 🖻 👤	00?B	🕅 🗘 🛃 🏂 🗶
	inscription outvoid data	base	۹	-
Company Entry	वि			
Product Entry	त्र भ			
Contact Datails	ି ଜ			
<body> <wohyper1< th=""><th>action</th><td>L</td><th></th><td></td></wohyper1<></body>	action	L		
	actionClass directActionName			
application session	disabled fragmentidentifier			
contact ✓ contactDisplay	href otherTreft			
displayDetails	pageName			
enterCompanie enterProducts	queryDictionary string			
Edit Source •	target			

Figure A.74 Creating the Contact Details Hyperlink

7. Save your project.

Create the Main Page Hyperlink

1. With the other windows, you will create a single hyperlink to navigate back to the Main window. From ProjectBuilder double click ProductEntry.wo. Scroll down to the bottom of the page and add a new WOHyperlink as you did on the main page. This time the hyperlink will be "Main Page" and the action will be returnToMainPage, as seen in <u>Figure A.75</u> and <u>Figure A.76</u> respectively.

	Add Action
Name:	return I o Main Page
Page returned:	Main
	Cancel Add

Figure A.75 Creating the returnToMainPage Action



Figure A.76 Creating the Main Page Hyperlink

2. Repeat the above steps for the "CompanyEntry.wo" and "ContactDetails.wo" pages. Save the Project and close all of the web components. This will bring you back to the Project-Builder window.

Testing Navigation

1. To test our navigation we first need to build the project, click on the "Hammer" button to bring up the builder. In the Project Build window, click on the "Hammer" button to begin the build process. When the project has finished building you should see the result as in <u>Figure A.77</u>.



Figure A.77 Building the Application

2. To execute your WebObjects application, go back to the ProjectBuilder window and click on the "Terminal" button. The button next to the "i" information button. This will bring up a Launch panel. Click on the "Terminal" button to execute your application. You are not ready for any data entry yet, just test all the hyperlinks to make sure they work.

Contact Details Page

- 1. From ProjectBuilder, double click the ContactDetails.wo. Select the left of the "Main Page" hyperlink .
- 2. Select Elements->Heading->H1. Then enter Contact Details and hit the "Return" or "Enter" key.
- 3. Next choose Elements->Table. For the Dimensions, choose Rows:3 and Columns:2 as seen in Figure A.78, and click "OK".

New Table		
Configure the new table: Dimensions: Size:	Preview: (shown without using the width or height settings)	
Rows:3Width:unsetColumns:2Height:unset		
Layout: 🗌 Captioned	Cell Cell	
Border: O At bottom	Cell Cell	
Spacing: 📃 🔿 At top		
Padding: 💿 Unspecified		
 First row cells are header cells (<th>)</th> Second row is wrapped in a WORepetition)	Cancel OK

Figure A.78 Adding Page Tables

4. The first "Cell" will be highlighted, enter Name:, then hit the "Tab" key twice to get to the next row. Enter Telephone:, then hit the "Tab" key twice to get to the last row and enter "EMail". Your window should look like <u>Figure A.79</u>.

🗖 👘 🗅 ContactDetails wo 🗕 ~/localDeveloner/WebOhiects/WOContacts 🔤 🖽				
Contact Details				
Name: Cell Telephone: Cell EMail: Cell Main Page				
<body> <table> <tr> <td> <text></text></td></tr><tr><th>ContactDetails</th></tr><tr><th>application > session > returnToMainPage ></th></tr><tr><td>Edit Source</td></tr></table></body>	<text></text>	ContactDetails	application > session > returnToMainPage >	Edit Source
<text></text>				
ContactDetails				
application > session > returnToMainPage >				
Edit Source				

Figure A.79 Adding Table Row Titles

5. Now click to the right of this table and hit the "Return" or "Enter" key for a new line. Next choose Elements->Table. For the Dimensions, choose Rows:2 and Columns:3 also select "First row cells are header cells" and "Second row is wrapped in a WORepetition" as seen in <u>Figure A.80</u>, and click "OK".

New Table		
Configure Dimensions:	the new table: Size:	Preview: (shown without using the width or height settings)
Rows: 2 Columns: 3	Width: unset + Height: unset +	
Layout:	Captioned	Cell Cell Cell
Border:	🔿 At bottom	Cell Cell Cell
Spacing:	🔿 At top	
Padding:	Output	
Padding: O Unspecified		Cancel OK

Figure A.80 Adding a Repetition Table

6. The first "Cell" will be highlighted, enter Name:, then hit the "Tab" key to get to the next column. Enter Price:, then hit the "Tab" key to get to the last column and enter Company:, now your window should look like <u>Figure A.81</u>.

🗉 📄 ContactDetails.wo 🗕 ~/LocalDeveloper/WebObjects/WOContacts	JE	
	×> *	
Ivane. Den	-	
Telephone: Cell		
EMail: Cell		
Name: Price: Company:		
Cell Cell Cell	=	
	_	
1 ⁴		
Main Page	-	
<pre><body> <table> <worepetition> <tr> <td> <text></text></td></tr></worepetition></table></body></pre>	<text></text>	
<text></text>		
ContactDetails		
application >		
session >		
returnToMainPage >		
Edit Source		

Figure A.81 Adding the Table Row Title

7. Now click to the right of this table for a new line. Next choose Elements->Table. For the Dimensions, choose Rows:9 and Columns:2, also de-select "First row cells are header cells" and "Second row is wrapped in a WORepetition" as seen in Figure A.82, and click "OK".

	New Table		
Configure the new table:		Preview: (shown without using the width or height settings)	
Dimensions:	Size:		
Rows: 9 Columns: 2	Width: unset \$ Height: unset \$	Cell Cell Cell Cell Cell Cell Cell Cell	
Lavout:	🗆 Cantioned	Cell Cell	
Border:	O At bottom		
Spacing:	🔿 At top	Cell Cell	
Padding:	Our State of the state of th	Cell Cell	
☐ First row cells are ☐ Second row is wra	e header cells (<th>) apped in a WORepetition</th> <th>Cancel</th>) apped in a WORepetition	Cancel

Figure A.82 Adding Company Detail Page Data Tables

8. The first "Cell" will be highlighted, enter Name:, then hit the "Tab" key twice to get to the next row. Enter Address1:, then hit the "Tab" key twice to get to the next row and enter Address2:, then hit the "Tab" key twice to get to the next row and enter City:, then hit the "Tab" key twice to get to the next row and enter "State:" then hit the "Tab" key twice to get to the next row and enter "Zip:, then hit the "Tab" key twice to get to the next row and enter "Tab" key twice to get to the next row and enter "Zip:, then hit the "Tab" key twice to get to the next row and enter "Country:", then hit the "Tab" key twice to get to the next row and enter "Telephone:", then hit the "Tab" key twice to get to the next row and enter "EMail:". Your window should look like Figure A.83.



Figure A.83 Adding the Table Row Titles

9. Close this window and from Project Builder double click on the CompanyEntry.wo section. Click next to the right of the "Search for Companys" heading and hit the "Return" or "Enter" key. Next choose Elements->Table. For the Dimensions, choose Rows:2 and Columns:2, also select "First row cells are header cells" and "Second row is wrapped in a WORepetition" as seen in Figure A.84, and click "OK".

	New Table		
Configure the new table: Dimensions: Size:		Preview: (shown without using the width or height settings)	
Rows: 2 Columns: 2	Width: unset \$ Height: unset \$		
Layout:	Captioned	Cell Cell	
Border:	🔿 At bottom	Cell Cell	
Spacing:	🔿 At top		
Padding:	Our State of the state of th		
Padding: ● Unspecified ✓ First row cells are header cells (<th>) ✓ Second row is wrapped in a WORepetition</th> <th>Cancel OK</th>) ✓ Second row is wrapped in a WORepetition	Cancel OK

Figure A.84 Adding a Repetition Table

10.The first "Cell" will be highlighted, enter "Contact Name:", then hit the "Tab" key to get to the next column and enter "Contact EMail:". Your window should look like <u>Figure A.85</u>.



Figure A.85 Adding the Contact Table Rows

11. Click next to the right of the table you just created and choose Elements->Table. For the Dimensions, choose Rows:2 and Columns:2 also select "First row cells are header cells" and "Second row is wrapped in a WORepetition" as seen in Figure A.86, and click "OK".

🗉 📄 CompanyEntry.wo 🗕 ~/LocalDeveloper/WebObjects/WOContacts 📃 🖽 🗄
S I U T 3 E None ⇒ ¶ = = = - B Ø ↓ F ↔ A D ■ F F F E S I V € E R Ø Ø ? B B € € S S
Contact Name: Contact EMail: Cell Cell
Cell Cell
Specify which Companys to display below: «
CompanyEntry
application > session > company > y companyDisplayGroup >
returnToMainPage saveChanges selectObject

Figure A.86 Adding the Product Table

12.The first "Cell" will be highlighted, enter "Product Name:", then hit the "Tab" key to get to the next column. "Product Price:". Your window should look like <u>Figure A.87</u>.

CompanyEntry.we	o – ~/LocalDe	eveloper/WebObjects/WOContacts 📃 🗏 🗏
③ ● B I U T 3 ▲ □ ■ ■ ■ ■ ■	None	 ♦ ¶ = ::::::::::::::::::::::::::::::::::
	Contact Name:	Contact EMail:
	Cell	Cell
	Product Name	Product Price:
	Cell	Cell
		1
Spec	fy which Compar	e≓ ys to display below: «⊣
<body> <center> point</center></body>		
CompanyEntry		Ĩ
application	> 4	
session	2	
company companyDisplayGroup	Ś.	
returnToMainPage		
selectObject		
Edit Source 👻		

Figure A.87 Adding the Table Column Titles

Connect the Table to the Component Page

1. In order to display the Contact name and Contact EMail, we have to add the Contact table to the page. Click on the ProjectBuilder window. Select the "Resources" from the left column. Double click on the "Contacts.eomodeld" to bring up the EOModler application. In EOModeler select the Contact table and drop it onto your CompanyEntry page as seen in Figure A.88.

WOContacts - ~/LocalDeyeloper/WebObjects 回日 CompanyEntry.wo - -/LocalDeveloper/We × Q 0 2 3 0 0 B I U T 3 E Nove . · Contacts exmodeld Classes Headers Main api Contact Name: Contact EMai Other Sources ProductEntry.api Cel Col Contacts.comodeld - -/La **Product Name: Product Price** Cell Cell 臣, 圖 圖 🐰 Contect Athibutes Specify which Companys to darplay be Contects - - - Name B Company B Contact same tomperofisionLeyferoup.querymate address1 <0001> <CENTER> point B-O Product a address2 - Stored Procedures CompanyEntry application a country session + a creail company Contact • a name a state a telephone a zip Enit Source -Add Celuen . Contact Relationships + Name Source Att Dest.Att Destination >> + toCompany Company contactID contac#D

Figure A.88 Adding the Contact Table to the Component

2. Click "Add and Configure" on the panel for your contactDisplayGroup as seen in Figure A.89.

	Add Display Group
Name: contac	tDisplayGroup
Cancel	Add Add & Configure

Figure A.89 Create contactDisplayGroup

3. Select "Fetches on Load" to ensure that data is loaded into the form when it is displayed, and name for Sorting Ascending and click "OK" as seen in Figure A.90.

A WebObjects Application

WebObjects Application Creation

	Display Group Options
Entity:	Contact 🗾
-Master/Detail-	
	🗌 Has detail data source
Master Entity:	
Detail Key:	×
Entries per batch:	0
Qualification:	Prefix 🗢
	☑ Fetches on load
Sorting:	name 🗢
	Ascending
	O Descending
	O Not Sorted
Fetch Spec:	<none></none>
Clear	Revert Cancel OK

Figure A.90Configure the Contact displayGroup

4. Repeat the same for the Product table so you can display the Product names and Product Prices. Notice that you now have three display groups in your page as seen in Figure A.91.

CompanyEntry.we	o – ~/LocalDev	eloper/WebObjects/WOContacts	UE
③ ● ● I U T 3 ▲ □ ■ ■ ■ ■ ■ ■	None SRIV(़ • ¶ ⊒ ⊟ ≟ − ⊠ Ø J FF © ∭ ⊑ Q Ø 9 & ⊒ ≝ 0	<>>
	Contact Name: C	ontact EMail:	
	Cell C	ell	
	Product Name: H	roduct Price:	
	Cell	Jell	
Spec name: C <body> <-CENTER> point</body>	≪⊐ ify which Company: ompanyDisplayGro	s to display below: ← up.queryMatch.name _←	•
CompanyEntry		WODisplayGroup	
application session company y companyDisplayGroup y contactDisplayGroup y productDisplayGroup returnToMainPage	> > > > >	allObjects allQualifierOperators batchCount currentBatchIndex displayedObjects hasMultipleBatches queryBindings queryMatch	»▲ »■ »
Edit Source			

Figure A.91 Create the productDisplayGroup

5. Now that you have access to the Contact and Product tables, you need to have an object that will map to the data. This is where you will add two objects. The first will be a contact object. From the web page's Edit Source popup button in the lower left frame of the window, select Add Key and enter contact of Type "Contact" and check "An instance variable", "A method returning the value" and "A method setting the value" and click "Add" as see in Figure A.92.

	Add Key
Name:	contact
Tune	(tune as niven)
iype.	Array of
	O Mutable array of
	Contact -
Genera	ate source code for:
	🗹 An instance variable
	A weathed waterwain a the realize
	A method returning the value
	Prepend "get" to method name
	A method setting the value
	Cancel Add

Figure A.92 Create the contact Data Object

6. Repeat the same for a product of object type Product. When you have completed this your page should look like <u>Figure A.93</u>.



Figure A.93 Create the product Data Object

- 1. Now that you have all the data links in place, it is time to make page variables to hold the database values. Double click "Cell" under Contact Name: .
- 2. Choose WebObjects->WOString from the menu. Click the WOString you just created and choose WebObjects->WOHyperlink from the menu. Your page should look like <u>Figure A.94</u>.



Figure A.94 Adding contact Name Field for the Contact Table

3. Tab to the next "Cell" and choose WebObjects->WOString from the menu, and your window now should look like <u>Figure A.95</u>.



Figure A.95 Adding contact EMail Field for the Contact Table

4. Repeat the same for the Product Name and Product Price so your page will look like <u>Figure A.96</u>. Don't forget to make the WOString and WOHyperlink for the Product Name as you did for the Contact Name:.



Figure A.96 Complete the Product data Area

5. To display what contact and product was chosen we will have to place a table to display these values. Click to the right of the Product Name Product Price table and select Elements->Table from the menu. Make the Rows:2 and Columns:2 and deselect the "First row cells are header cells" and "Second row is wrapped in WORepetition" and click "OK", as seen in <u>Figure A.97</u>.

	New Table		
Configure the new table:		Preview: (shown without using the width or height settings)	
Dimensions:	Size:		
Rows: 2	Width: unset 🜩		
Columns: 2	Height: unset 🗢		
Layout:	Captioned	Cell Cell	
Border:	🔿 At bottom	Cell Cell	
Spacing:	🔿 At top		
Padding:	Unspecified		
☐ First row cells ar ☐ Second row is wr	e header cells (<th>) rapped in a WORepetition</th> <th>Cancel OK</th>) rapped in a WORepetition	Cancel OK

Figure A.97 Configure the Selected Values Table

6. The first "Cell" will be highlighted, enter Contact:, then hit the "Tab" key twice to get to the next row. Enter Product:, so your window should look like <u>Figure A.98</u>.

🗉 📄 CompanyEntry.wo 🗕 ~/LocalDeveloper/WebObjects/WOContacts 📃 🛛
Contact Name: Contact EMail:
Product Cell Specify which Companys to display below: «
CompanyEntry application session company y companyDisplayGroup contact y contactDisplayGroup product y productDisplayGroup v

Figure A.98Add the Selected Values Row Title

7. To hold the selected values you will double click the "Cell" next to Contact: and select WebObjects->WOString from the menu. Repeat this for the Product "Cell" and your table should look like <u>Figure A.99</u>.

Image: Second and Seco] 🔄 🗋 CompanyEntry.wo 🗕 ~/LocalDeveloper/WebObjects/WOContacts 📃 🛛][
Contact Name: Contact Name: Contact Name: Product Name:	S I U T 3 E None < ¶ = := := - BØ J : F < A D ■ F = := := 0 Ø J : F <	>
CompanyEntry application session company y companyDisplayGroup contact y contactDisplayGroup product y productDisplayGroup	Contact Name: Contact EMail: Image: Contact Name: Contact EMail: Image: Contact Name: Product Price: Image: Contact: Image: Conta	
application > session > company > y companyDisplayGroup > contact > y contactDisplayGroup > product > y productDisplayGroup > v contactDisplayGroup > y contactDisplayGroup > y roductDisplayGroup > v contactDisplayGroup > v conta	CompanyEntry	
	application > session > company > v companyDisplayGroup > contact > v contactDisplayGroup > product > v productDisplayGroup > v contactDisplayGroup > v contactDisplayGroup > v product > v productDisplayGroup > v productDisplayGr	

Figure A.99 Add the Selected Values Column data

8. Now you have to make the data connections that will connect the database data values to the proper page elements. Click the WOString field under Contact Name and from the list of instance variables choose contact, name and drag it to the WOString and select value from the selections as seen in <u>Figure A.100</u>.



Figure A.100Make the Contact Name value connection

9. Repeat this for the corresponding values for the Contact EMail and for the product name and price values as seen in Figure A.101.

🗆 📄 Company	Entry.wo – ~/LocalDev	eloper/WebObjects/WOContacts	
	T 3 돌 None 파 떼 의 의 니 / ⓒ		F <>> ≝ ★
	Contact Name: (2) 2 contact.name 2) Product Name: (2) 2 product.name 2) Contact: 2	Contact EMail: 2. contact.email 2 Product Price: 2. product.price 2	
<body> <center> point</center></body>	Product: 👤		•
Compai	nyEntry	EOEnterpriseObject	
application session company companyDisplayGroup contact contactDisplayGroup product productDisplayGroup		address 1 address 2 city country email name state telephone toCompany	
Edit Source 🔻			

Figure A.101 All Connections completed

1. To know what object was selected from the Contact list and the Product list we will add the selected object from each of the display groups that represent the Contact and Product tables. Click on the WOString field next to the "Contact:" cell and from the list of instance variables navigate to the contact-DisplayGroup.selectedObject.name and drag it to the WOString and select value as seen in Figure A.102.



Figure A.102 Contact Selected Value Connection

2. Repeat the same for the productDisplayGroup.selectedObject.name as seen in Figure A.103.



Figure A.103 Make the Product Value Connection

3. The final task is to connect the repetition of the Contact Name to the Contact table and the Product Name to the Product table. Select the contact.name WOString field. Then click on the <WORepetition> entry. select the contactDisplayGroup.displayedObjects as the list as seen in Figure <u>A.104</u>.



Figure A.104 Connecting the Data Value List Connection

1. Now that you have cancelled the list objects, you have to declare of what object type is the list made of. This is where we will connect the contact object to the item. Select the contact object and drag to the <WORepetition> and choose "item" from the list as seen in Figure A.105.

CompanyEntry.w	o – ~/LocalDeve	eloper/WebObjects/WOContacts	JE
③ ◎ ◎ B I U T 3 ▲ ○ ■ □ □ □ □ □ □]≣ None s R I V ⊙		<>
Search for Companys			
Ca All Ca Pr	ntact Name: ontact.name 20	Contact EMail: Contact.email Q Product Price:	
<pre> Contact: Q contactDisplayGroup.selectedObject.name Contact: Q contactDisplayGroup.selectedObject.name SODY> <center> <table> </table></center></pre>			
CompanyEntry	identifier	EOEnterpriseObject	
application session company	index item • list		•
 ✓ companyDisplayGroup contact ✓ contactDisplayGroup 	Connect to ne	ew binding email name	
product ⊮ productDisplayGroup	> > •	state telephone toCompany >>>	•
Edit Source 🔻			

Figure A.105 Connect the Data Object for the Repetition

- 2. Repeat the above steps for the product object and product-DisplayGroup.displayedObjects
- 3. Now we need a method to respond to the <WOHyperlink> selection of a contact object. So a selectContactObject will have to be created as follows:

Listing 0.1 selectContactObject method

```
public void selectContactObject() {
    contactDisplayGroup.selectObject(contact);
}
```

Place this code right below the selectObject() method.

4. Now that the method is in our project, it is time to connect the WOHyperlink to this action. Select and drag the select-

ContactObject entry to the <WOHyperlink> entry and choose action from the list as seen in <u>Figure A.106</u>.



Figure A.106 Connecting the Action for the Hyperlink

5. Now we need a method to respond to the <WOHyperlink> selection of a product object. To edit the source, select from the "Edit Source" popup button "View Source File". Add the following selectProductObject method to the java source.

Listing 0.2 selectProductObject method

```
public void selectProductObject() {
    productDisplayGroup.selectObject(product);
}
```

Place this code right below the selectContactObject () method.
6. Now that the method is in our project, it is time to connect the WOHyperlink to this action. Select and drag the select-ProductObject entry to the <WOHyperlink> entry and choose action from the list as seen in <u>Figure A.107</u>.



Figure A.107 Connect the Action for the Product Name Hyperlink

Congratulations, all of your connections have been made. Your page should look like Figure A.108.



Figure A.108 All Connections Completed

1. The last thing that you have to do is modify the saveChanges() method to get the selected values from the page to actually be stored in the database. The modifications to the method appear in bold below:

Listing 0.3 saveChanges method

```
public void saveChanges() throws Exception {
    EOEnterpriseObject aCompany =
    (EOEnterpriseObject)companyDisplayGroup.selectedObject();
        EOEnterpriseObject)contactDisplayGroup.selectedObject();
        EOEnterpriseObject aProduct =
    (EOEnterpriseObject)productDisplayGroup.selectedObject();
```

aCompany.addObjectToBothSidesOfRelationshipWithKey(aContact, "toContact");

aCompany.addObjectToBothSidesOfRelationshipWithKey(aProduct, "toProduct");

```
try {
    this.session().defaultEditingContext().saveChanges();
    catch (Exception exception) {
        // An error occurred during the save. You could
    present an error page which
        // explains the reason for the save failure.
        // The default is to raise an exception which
    presents a diagnostic page.
        System.err.println("Cannot save changes ");
        throw exception;
    }
}
```

2. To edit the source select from the "Edit Source" popup button "View Source File" and find the saveChanges method and enter the modification and save the project.

Additions to the ContactDetails Page

1. From the ProjectBuilder window, choose the "ContactDetails.wo" page. Double click the "Cell" field next to Name:. Choose WebObjects->WOString from the menu, hit the "Tab" twice and add another WOString for the Telephone: and hit "Tab" twice to add another WOString for the EMail: "Cell" value, as seen in Figure A.109.



Figure A.109 Adding the data Value Fields

 Double click on the "Cell" value on the table, under the Name: column. Choose WebObjects->WOString from the menu, hit the "Tab" and add another WOString for the Price: and hit "Tab" to add another WOString for the Company: "Cell" value, as seen in <u>Figure A.110</u>.

🔲 📄 ContactDetails.wo – ~/LocalDeveloper/WebObjects/WOContacts 📃 🗉				
Name: V. V. Telephone: Q. EMail: Q. Name: Price: Company: Q. Name: Cell Address1: Cell				
<body> <table> <worepetition> <tr> <td> <wostring></wostring></td></tr><tr><th>ContactDetails</th></tr><tr><td>application >> session >> returnToMainPage</td></tr><tr><td>Edit Source</td></tr></worepetition></table></body>	<wostring></wostring>	ContactDetails	application >> session >> returnToMainPage	Edit Source
<wostring></wostring>				
ContactDetails				
application >> session >> returnToMainPage				
Edit Source				

Figure A.110 Add the data Value Fields for the Repetition

3. Double click the "Cell" field next to Name:. Choose WebObjects->WOString from the menu, hit the "Tab" twice and add another WOString for the Address1: and hit "Tab" twice to add another WOString for all the other table: "Cell" values, as seen in Figure A.111.

🗉 🕒 ContactDetails.wo – ~/LocalDeveloper/WebObjects/WOContacts 📃 🗉
Name: Address1: Address2: Q Address2: Q City: Q State: Q Zip: Q Country: Q EMail: Q
ContactDetails
application > session > returnToMainPage
Edit Source

Figure A.111 Adding the data Value Fields for the Company Details

4. To connect the selected contact coming from the main page, a contact object will be added to this page. From the "Edit Source" popup button select Add Key with the Name of selectedContact and Type of Contact with "An instance variable", "A method returning the value" and "A method setting the value" checked as seen in Figure A.112, and click "Add".

Add Key		
Name:	selectedContact	
Туре:	Image: March Strength (1998) (1998	
	O Array of	
	O Mutable array of	
	Contact -	
Genera	te source code for:	
	☑ An instance variable	
	☑ A method returning the value	
	Prepend "get" to method name	
	☑ A method setting the value	
	Cancel Add	

Figure A.112 Create the Data Variable for the selectedContact

5. Now it is time to make the connections to the database values. Select the WOString to the right of the "Name:" field at the top of the page. Choose the selectedContact.name value and drag it to the <WOString> entry and select value from the list as seen in Figure A.113.



Figure A.113 Connecting the selectedContact Name Value

6. Repeat the above for the Telephone and EMail values as seen in Figure A.114.

🗌 🔄 🗋 ContactDetails.wo – ~/LocalDev	/eloper/WebObjects/WOContacts 📃 🗄 🗄			
(3) (3) (3) (3) (3) (3) (3) (3) (3) (3)	◆¶≡∷≕−≧øø↓∓↔ ™⊡₽Ø0?⊌®©¢*≫★			
Contact Details	_			
Name: Q_selectedContact.name Telephone: Q_selectedContact.telephone EMail: Q_selectedContact.email	∃			
Name: Price: Company: QQQQQ Name: QQ Address1: QQ				
<b0dy> point</b0dy>				
ContactDetails	E0Enterprise0bject			
ContactDetails application session selectedContact	EOEnterpriseObject			
ContactDetails application > session > selectedContact > returnToMainPage	EOEnterpriseObject			
ContactDetails application session selectedContact returnToMainPage	EOEnterpriseObject			

Figure A.114 selectedContact data Values connected

1. To be able to access the Company table values, first a company object will be added, then a companyDisplayGroup will be added. To add the company object, select from the "Edit Source" popup button select Add Key with the Name of company and Type of Company with "An instance variable", "A method returning the value" and "A method setting the value" checked as seen in Figure A.115, and click "Add".

	Add Key
Name	company
Type:	(type as given)
	O Array of
	O Mutable array of
	Company 🔹
Gener	ate source code for: ☑ An instance variable
	A method returning the value
	Prepend "get" to method name
	A method setting the value
	Cancel Add

Figure A.115 Create a company Data Object

2. Now to add the companyDisplayGroup, select the Company table in EOModeler and drag it into the ContactDetails page as seen in Figure A.116.



Figure A.116 Add the Company Data Table to the Component

3. Click on the "Add & Configure" button as seen in <u>Figure</u> <u>A.117</u>.

	Add Display Group
Name: compar	nyDisplayGroup
Cancel	Add Add & Configure

Figure A.117 Create the companyDisplayGroup

4. For the Display Group Options, check "Has detail data source", Master Entity: Contact and Detail Key: to Company, check "Fetches on load and name in Ascending Order as seen in <u>Figure A.118</u>, click "OK".

	Display Group Options
Entity:	Company
– Master/Detail –	
	🗹 Has detail data source
Master Entity:	Contact 🔹
Detail Key:	toCompany _
Entries per batch:	0
Qualification:	Prefix 🗢
	Fetches on load
Sorting:	name 🗢
	O Ascending
	O Descending
	Not Sorted
Fetch Spec:	<none></none>
Clear	Revert Cancel OK

Figure A.118 Configure the Company Display Group

- 5. To add the values for the Product Name, Product Price and Company Name the following will be connected. For the repetition select the WOString under the Name: title in the "Name, Price and Company" repetition.
- 6. Then select the <WORepetition> field. From the list select the companyDisplayGroup.displayedObjects and drag to the <WORepetition> and select list as seen in Figure A.119.

rms Window Services	Help		2:42 PM	WebOb
ContactDe	tails.wo/loca	Developer/WebObjects/	WoContacts	日日
3 0 0 B / U	T 3 또 Nore			新 第 8
Telephone Enslected	ontact telephone 3			-
Name: Price: Com R. G. A. A. S. Name: R. B.	party:			
Address TABLES	count		Cherry	•
aliObjects aliQualiterOperators	index item	rig sount	SHITEY	2
currentBatchindex displayedObjects hasMultipleBatches	Connect to new I	endingnn @sus count		~ ~ ~
masterObject uveryMatch Exit Source - •				-

Figure A.119 Define the Display Group Repetition List

Now with the list identified, the list object type must be declared. Select the company object and drag it to the

 WORepetition> and select item as seen in Figure A.120.

Telephone Delected on	act telephone Q		-
EMul Elected or	act email 1		
Name: Price: Compar	WI.		
Address1 2 2			
Admas2 10 10	-		
SECONS STREETS SALESPECTO	count	Call International	
Contacto	index	EDENterprised	object
application	nem	1551	-
SESSION COMPOSIT	• list	1352	
✓ companyOisplayGroup selectedContact	Connect to new bind	enall	
utumTaMainPage		state	

Figure A.120 Define the Object for the Repetition

8. Now we will make the field connections to the database. Select the WOString for the Name field in the repetition and



drag the company.toProduct.name for the value as seen in <u>Figure A.121</u>. Do the same for the company.toProduct.price.

Figure A.121 Making the Field Connections to the Database

9. For the final column in the repetition select the company.name and drag that to the selected WOString for the "Company" field and select value. All three of the fields are now connected as seen in <u>Figure A.122</u>.

🗆 🔛 🗋 ContactDetails.w	o – ~/LocalDev	/eloper/WebObjects/WOContacts		
 ● ● B I U T 3 ▲ ● ■ ■ ■ ■ 		↓ ↓ = = = = = • ↓ ↓ • ★ 0 & & & = = • • • • • • • • • • • • • • •	-; ↔ ≝ ★	
Telephone 9 selectedContact.t EMail: 9 selectedContact.e	elephone Q mail Q		Ê	
Name:	Price	:: Company:		
🞗 company.toProduct.name 🞗	2 company.toProc	duct.price 🞗 🎗 company.name 🞗		
Name: Q Q Address1: Q Q Address2: Q Q				
ContactDetails		E0EnterpriseObject		
application session	>	country email	-	
company	>	name		
✓ companyDisplayGroup selectedContact	>	state telephone toContact	,∎	
returnToMainPage		toProduct zip	>	
Edit Source 🔻				

Figure A.122 The Repetition Defined

- 10. Finally we need to connect the company's database values to the last table. The database values will come from the companyDisplayGroup.selectedObject value.
- 11. Select the WOString field next to the Name entry. Click companyDisplayGroup.selectedObject.name and drag it to the <WOString> and select value as seen in Figure A.123.

ms Window Services Help		3.91 PM 📝 Web0
		ects/W0Centacts DE
Marac B B Addrawn E B Addrawn E B Cryv B B Stote B B Zyv B B		•
+500% «TABLE» «TR» «TD» (TABLE» (TR» «TD») WOOksplayGroup Wotterorpizer quaryMatch	dateformat escapeHIML formatter numberformat	Company
gueryMax gueryMin gueryOperator relationalQualifierOperators selectedObject sthrp2/usiliterOperators	Connect to new binding	-
Edit Source •	Tell methodole	

Figure A.123 Making the Company data Value Connections

12. Follow the same steps for all of the company fields. When you are finished, your page should look like <u>Figure A.124</u>.



Figure A.124 All Company data Connections Completed

1. Finally if there are more than one company we want to display the details for the selected company. To allow for the selection of a particular company, select the company.name WOString in the table. Choose WebObjects->WOHyperlink from the menu as seen in Figure A.125.

🗈 📄 ContactDetails.wo	o – ~/LocalDeveloper/W	eb0bjects/W0Contacts	EE	
🖲 🕄 🞒 B I U T 3	None 🗘	¶⊒ΞΞ= Β ØJF	<x></x>	
	SRLV® E	₽Ø0?&₽₽₽	*	
Name:	Price:	Company:	-	
🞗 company.toProduct.name 🎗 🖠	2 company.toProduct.price	9 🖉 🔍name 9 😥		
Name: 👤 companyDisplayGrou	up.selectedObject.name 🞗			
Address1: 👤 companyDisplayGrou	up.selectedObject.address1	9		
Address2: 👤 companyDisplayGrou	up.selectedObject.address2	9	=	
City: 🔍 companyDisplayGrou	up.selectedObject.city 👤			
State: 👤 companyDisplayGrou	up.selectedObject.state 🞗			
Zip: Q companyDisplayGrou	up.selectedObject.zip 🞗			
Country: 🙎 companyDisplayGrou	up.selectedObject.country	2		
<pre> Telephone CompanyDisplayGrou <body> <table> <worepetition> <t< pre=""></t<></worepetition></table></body></pre>	<u>un selectedObject telenbone</u> R> <td> <wohyperlink> <wos< td=""><th>lol</th><td>-</td></wos<></wohyperlink></td>	<wohyperlink> <wos< td=""><th>lol</th><td>-</td></wos<></wohyperlink>	lol	-
ContactDetails				
application session company Y companyDisplayGroup selectedContact returnToMainPage				
Edit Source 🔻				

Figure A.125 Adding the Company Hyperlink

2. From the "Edit Source" popup button select View Source. and add the following method:

Listing 0.4 selectObject method

```
public WOComponent selectObject() {
    companyDisplayGroup.selectObject(company);
    return null;
}
```

3. Now back in the page select the company.name and select the <WOHyperlink> and from the list click on selectObject and drag to the <WOHyperlink> and choose action as seen in Figure A.126.



Figure A.126 Adding the Action to the Hyperlink

Passing of Contact Objects

1. Add the following <u>Listing 0.5</u> displayDetails method to Main.java. This will pass the selected Contact object to the Contact Detail Page.

Listing 0.5 Main.java

```
public ContactDetails displayDetails() {
    ContactDetails nextPage =
(ContactDetails)pageWithName("ContactDetails");
```

```
// Initialize your component here
    EOEnterpriseObject selection =
(EOEnterpriseObject)contactDisplayGroup.selectedObject();
    nextPage.setSelectedContact(selection);
```

return nextPage;

}

2. Add the following highlighted line of code, see <u>Listing 0.6</u> to the selectedContact method to the ContactDetails.java. This will set the contact into the Master Company object.

```
Listing 0.6 ContactDetails.java
```

```
public void setSelectedContact(EOEnterpriseObject
newSelectedContact) {
    selectedContact = newSelectedContact;
    companyDisplayGroup.setMasterObject(newSelectedContact);
  }
```

Congratulations, you have completed your first WebObjects application using FrontBase as you database. In the next section you are going to build and run your application.

Running the WOContacts Application

Building the Application

1. From the ProjectBuilder window, you will see the "Hammer" button as in <u>Figure A.127</u>.

WOContacts - ~/localDeveloper/WebObjects		
<u><u> </u></u>	Web Components	
Classes P Main wo P Headers P Other Sources P Robources P Subprojects P Subprojects P Frameworks P		
	Done cellecting classes information	

Figure A.127 The Hammer Button

2. Click on the "Hammer" button to bring up ProjectBuilder's Build Panel as seen in Figure A.128.



Figure A.128 The Build Panel

3. Click on the "Hammer" button from this window and you will start the application building process. When the project has completed building you will see Figure A.129.



Figure A.129 A Successful Build

Running the Application

1. Back in ProjectBuilder's window, you will see an "Terminal" button. Click on this and you will bring up the Launch window.Click on the "Terminal" button in this panel and your web application will launch and bring up your web browser with your Main.wo page. See <u>Figure A.130</u>.



Figure A.130 The Launch Panel

Adding Contacts

1. Click on the Insert/New Button and enter a new contact, and then click on the Save to Database as seen in Figure A.131.



Figure A.131 Add a New Contact

2. Add one more contact as seen in <u>Figure A.132</u>. After saving to the database, clicking on the telephone hypertext fields will bring up the entered information for each contact.

Contacts Cli		
err and Re 1 + C	Construction of the Constr	
	Search for Contacts	
	Equally which Constants to deploy below: a state telephony a state Match Glob a link to a select that contact. INVERSE	
neme Anna 399 eddeanti: 122.8.003.020001 eddeanti: 122.8.003.020001 eddeanti: 122.8.003.020001 eddeanti: 122.8.003.02000 eddeanti: 122.8.003.0200 eddeanti: 122.8.003.0000 eddeanti: 122.8.003.0000 eddeanti: 122.8.003.0000 eddeanti: 122.8.003.0000 eddeanti: 122.8.0000 eddeanti: 122.8.00000 eddeanti: 122.8.0000 eddeanti: 122.8.00000 eddeanti: 122.8.0000 eddeanti: 122.8.00000 eddeanti: 122.8.00000 eddeanti: 122.8.00000 eddeanti: 122.8.00000 eddeanti: 122.8.000000 eddeanti: 122.8.00000 eddeanti: 122.8.000000 eddeanti: 122.8.00000 eddeanti: 122.8.00000 eddeanti: 122.8.000000 eddeanti: 122.8.00000000 eddeanti: 122.8.0000000000000000000000000000000000	kaonifikara Bangadanan Databa	
Company linter Parlar, linter		
Gustart Datada		

Figure A.132 Displaying the Entered Contacts

Adding Products

1. Now click on the Product Entry hyperlink and enter two products as you entered two contacts. Remember, click Insert/New and enter the information. Then click Save to database as seen in Figure A.133.

Predacts P	
	-
Page address:	+
Search for Products	
Ryandly which, Products to display below.	
Matten Click s.lad. to refere ther product <u>Broduct Data</u> Handwar Ten	
James Freedowsk fere Jaine 20 Freedows James Denke	
StatZup.	

Figure A.133 Saving the Products

2. From the Product Entry page, click on the Main Page hyperlink to return you to the Contact Entry page. In the Contact Entry page click on the Match button to bring up your contact list and select your second contact from the telephone hyperlink as seen in <u>Figure A.134</u>.

Centerts III	
	0
legie address: 🛐 [http://lacalitest:1223/cgi-bin/WebObjects/WPContacts.wea/wo/Wg9bDHQig1MLNEM1/13.7.1.1	+
Search for Contacts	
Spring which Contains to display below: name Childphone American Chick to reflect the represet. Chick to reflect the represet.	
amia Jan Johnrow Jan Johnson 1212 Januard Atemati addresse 1782 222 1782 222 178	
Company Electry	
Yudina Ewage	
Instant Details	

Figure A.134 Selecting a Contact

Adding Companies

1. Now click on Company Entry hyperlink to enter two Companies. To enter the company information you will have to first select a contact name. Next, select a product name as seen in <u>Figure A.135</u>.

Genzenys	
Pege eddress: 🛐 [http://acolhest.1223/cgi-bin.WishObjects/IPDContacts.wee/wo/Wy9hDHQigIMINEMI/16.5.8.1	-
Search for Companys	
Contact Numeri Contact EMails Jan Johanni veriffword, rom Jahn Dan medfinnen som Product Neuer: Product Price: Frankrat Tars 20 Context fine Johanne Product: Product Con Signify which Companyor to Signify balawe	
SARCE] Click a link to calent that roughny	
addeast	
mand New Start Date	

Figure A.135 Adding the Company

2. To enter the company will now be the same as all the other entry steps. Click on the Insert/New, enter the information then click on the Save to Database. Using these steps, enter Three Companies as seen in Figure A.136.

NOTE: Remember to select a different Contact and Product **BE-FORE** clicking the Insert/New button and entering the information then clicking the Save to Database button.

Gregarys	
(m m 金 四 · · · · · · · · · · · · · · · · · ·	0
Page address: D Inter/Acalhest1223/cgi-bin/WebObjects/PE0Contacts.wee/wo/Wg9E0804g1MINEM1/	79.19 +
Search for Companys	L
Contact Name: Contact EMail:	
Jan Jahorem wereflword, cost	
Julan Data sure @Roose com	
Product Name: Product Price:	
Product Cine 30	
Roduit Tex 20	
Courses: Am. Johnson	
Photo: Photo: Tes	
Specify which Companys to display below	
and a	
and a local second	
SADO B	
Click a link to reduct that roughny	
Company, Jon	
Gompany Three	
name: Toopany Three	
address? 100 Waard ofcurt	
sidewse2 [res 100	
alp: Shrué furs	
stain. In.	
aip Transf-man	
scently: sile	
Telephone: [1101]800-1222	
and Gasafteria in	
mother last for	8

Figure A.136 Saving the Company

Displaying Company Details

1. With the entry completed, click on the Main Page hyperlink to return to the main Contact Entry page. At the Main page click on the "Match" button and choose a Contact, then click on the Contact Details page. 2. From the Contact Details page, the First Company is displayed <u>Figure A.137</u>. Click on your second company in the list and you will see that company's information listed as seen in <u>Figure A.138</u>.



Figure A.137 Displaying the First Contact Detail Entry



Figure A.138 Displaying the Second Contact Detail Entry

Congratulations, you have completed this exercise. While this is not a ready-for-primetime application, that would include page formatting and error checking, it demonstrates how the WebObjects framework was able to integrate with the FrontBase database framework with relative ease.

Summary

It cannot be overlooked what the database has stored for this application. There are three tables; Contact, Company and Product, all with relationships to each other that are resolved on the Contact Details page. This is quite an accomplishment for a web based application.

I hope this exercise has offered some insight into web based database applications. As always, if you have any questions please visit the FrontBase website at www.frontbase.com, for further information, support or to download your FREE copy of FrontBase and license string.

From all of us at FrontBase, Thank You for taking time to read/develop the steps to build this application.

Index

A

Adding Companies 135 Adding Contacts 132 Adding Products 133

В

Building the Application 130

С

Connect the Table to the Component Page 89 Create the CompanyEntry Page 61 Create the Contact Entry Page 45 Create the ContactDetails Page 68 Create the Main Page Hyperlink 77 Create the ProductEntry Page 54 Creation of the Contacts Database 12

D

Displaying Company Details 137

Ε

Enterprise Object Model Creation 26 EOAdaptor Connection Dictionary Keys 24

0

Overview 5

P Package Installation 5

R

Running the Application 131

S

Summary 140

Т

Table Creation 14The Company Table 26

W

WebObject Application Creation 44

Index