FrontBase® Building a Database with Qilan®



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

How to Contact FrontBase:

Table of Contents

Α	Building a Database with Qilan® 5
	Overview
	The Database.
	Database Creation \ldots \ldots \ldots \ldots \ldots \ldots
	Starting the Database
	Starting Qilan
	The Schema
	Schema Creation
	Exporting to FrontBase
	The User Interface
	Building the Framework
	Template Construction
	Processing Q Tags
	Accessing Existing Data
	Exporting the Webtemplate
lı	idex 43

Index

A

Building a Database with Qilan®

Qilan is a tool for manipulating data on a web server. It handles database connections, business logic and user interface screens.

Overview

Qilan sits between the web server, which handles client requests, and the database. It functions as a common gateway interface (cgi). When the web server receives a request, the web server launches a special Qilan document which in turn is executed by qilan.cgi. These specialized documents, also known as 'webtemplates', are created by the Qilan application (developer).

As we proceed through this tutorial, we will be working with the Qilan developer.

The topics in this section are:

- <u>"The Database" on page 5.</u>
- <u>"The Schema" on page 11.</u>
- <u>"The User Interface" on page 19.</u>

The Database

1. The first step is to create an empty database. Qilan works with an external database after it has been created and named. From the Viewer, locate FBManager.app and launch it.



Figure A.1 The FBManager.app

Database Creation

2. Select 'New' from the Monitored Databases window. You can name the database almost anything you want; just remember the name is case sensitive. If you are running the database on the local machine, name the host, "localhost". You can also use the machine's IP number. Click, CREATE.

Monitored Databases	121
Manitared Databases	
Database Host Create Database	
Database Name: NewDB Run on Host: Localhost	2
Cancel	Create

Figure A.2 Creating the Database

Starting the Database

start the database.

3. The database is now created, but not yet started. Double click on the NewDB cylinder () to start the database and connect to it. The Connect Information window will open. Note that a default user name has been entered. The name, "_SYTEM" is equivalent to the UNIX logon as 'root'. In other words, it should be changed at some point to protect the database from unintended alterations. For the purposes of this tutorial, just leave it alone. Click CONNECT to connect and



Figure A.3 Connecting to the Database

4. The database cylinder has turned green. This is your indication that it is running. The management window will open, but for now, just ignore it. Let's proceed to the Qilan developer.

	Monitored Databases EE	
	NewOB(_SYSTEM) @ locathost : Management	
- Hithe escape VALUES(SERV VALUES(CURR SET TRANSAC	V Database V Table cache V Usage V Editor V B & Ikey to execute SGL statements ER_NAME]: ENT_USER, CURRENT_SCHEMA, CURRENT_TIMESTAMP TION ISOLATION LEVEL SERIALIZABLE, LOCKING OPTIM TION ISOLATION LEVEL SERIALIZABLE, LOCKING PESSI	W Show to Remove Append Abways If new Show Plan Auto Comm

Figure A.4 Database is Running

Starting Qilan

5. Returning to the Viewer, double click on Qilan.app. This will start the Qilan developer.

	Applications - Aoral	08
📇 to iterus, anz MB a	validate on SCSI stat.	
CGS_OS_X	r a r a r	Olian App
12 gilanitarigz bin consis dev etc FBWebEnabletLog lib Local mach_kemel Mac_OSX_Volume /Ad/	Administration F Applications F P Developer P Developer P Developer P Developer License Agreement.r. Overvlopage P License Agreement.r. P Developer P License Agreement.r. P Desterrigt TXT distiff Users F opendia Stf difs_0.wed r Jopendia Stf	
NewOB Incalifient	ntured Databases	

Figure A.5 Starting the Qilan Developer Application

6. Upon opening, Qilan will present an untitled window and a palette. From the palette, drag an Access icon into the untitled window. The Access icon reflects the database you are connecting to - in this case, FrontBase. Double click on the small access icon to reveal its contents.

Untitled-1

Figure A.6 Connect to FrontBase

7. The first aspect of the 'accessing' of a database is to login. Qilan provides two mechanisms to access a database: as a user and as a designer. A user has only data access. A designer has schema access. As we accepted FrontBase's default user name, "_SYSTEM", the same name/password will be used for user and schema login. Double click on the User Login parameter line to open the login panel.

	Untitled-1
[market]	
Name	
Access	
7	
	Untitled-1/Access
Attribute	Value
Attribute	Value FrontBase
Attribute Adaptor	Value

Figure A.7 Open the Login Panel

8. Enter the database name, host name and user name as required by FrontBase. Again note the database name is case sensitive, but user names are not. Leave password blank



Figure A.8 Enter Login Information

9. Repeat this for the Schema login parameter line.

The Schema

1. The next step is create our database schema (tables and fields). These objects are represented in Qilan, but created in our FrontBase database.

Schema Creation

2. This tutorial will create fields, but Qilan can also create relationships within or between tables. From the palette, drag the table icon into the untitled/access window.

9	Untitled-1/Access	Ð
		-
Attribute	Value	
Adaptor	FrontBase	
User Login	Host: locathost Database: NewD8 User: _SVSTEM	
Schema Login	Host: localhost Database: NewDB User: _SVSTEM	
Commant		
		_
Name		Uses
a lisance		(uses
NAME OF		
Table		

Figure A.9 Begin Schema Creation

3. Rename the table icon to 'Names', then double click on the icon to reveal it's contents. The window, Untitled-1/Access/Names will open.

External name is how you want the database to store this table. It may be the same as your Qilan name, but some organizations have existing naming conventions which you may be required to use. In any case, let's name it, "Names".

0	Unt	tted-1/Access	Ð	
Attribute	Value			
Adaptor	FrontBase			
User Login	Host localhost	Host, localhost Database, NewOB User: _SYSTEM		
Schema Login	Host localhost	Database NewOB User: _SYSTE	N	
Connent				
1	464.4		0.0000	
· Name	1	Untitled-1/Access/Names	E B B	
Names	-		WEATER O	
	Attribute	Value		
	External Name	Names		
	Comment			
	Name		Uses	
1	2		114	

Figure A.10 Rename Table to Names

Tables usually have fields, which are the essential data storage mechanisms. From the palette, drag a field icon into the Untitled-1/Access/Names window. Rename the default name to: "First_Name".

4. As with the table, double click on the small icon to open the field window. Type an external name, which is how the database will name this object.

The default field type for the database is 'character'. Qilan is very flexible about data types, but offers basic interpretations, such as string, date, float, etc. As data logic is handled in Qilan, you can default data types instead of data typing for each calculation or manipulation. As you will see later on, we can override these default settings as necessary.

Additional settings, such as, 'Allows Null', can also be set in Qilan. Make settings to match the Untitled-1/Access/Names/First_Name window below.

				*		5
Attribute		Value	_			QNs
Adaptor		Frante	lase			
User Login	1 - E	-	al host Databas	a.		
Schewallo	gin	Host loc	alhost Databas			- And
Comment					4	Chernle
		Untitle	f-1/Access/N	ames EE		
· Name				Untilled-1/	Access/Names/First_Name	一日
 Name Names 	Attribute Edomal Nas	10	Value		Access/Names/Tirst_Name	
and the second	and the second s	10	and the second division of the second divisio		Access/Names/Tirst_Name	
No. of Concession, Name	E-domal Nas	ne -	and the second division of the second divisio	Untitled-1/	A	
Name and Address of the Owner, where the	External Nas Comment	0	and the second division of the second divisio	Untitled-1/ Attribute	Value	
No. of Concession, Name	Erdomal Nas Comment		and the second division of the second divisio	Attribute Edemal Name Edemal Type Type	Value Fint_Name	
No. of Concession, Name	External Nas Comment		and the second division of the second divisio	Attribute E-demai Name E-demai Type	Value First_Name OHARACTER VARVING	
Names	Erdemal Nas Connent Name First_Na		and the second division of the second divisio	Attribute External Name External Type Type Primary Key Allows null	Value First_Name OHARACTER VARVING	
Name and Address of the Owner, where the	Erdemal Nas Connent Name First_Na		and the second division of the second divisio	Attribute Extensi Name Extensi Name Type Primary Key	Value Fint_Name OHARACTER VARVING string	
Names	Erdemal Nas Connent Name First_Na		and the second division of the second divisio	Attribute External Name External Type Type Primary Key Allows null	Value Fint_Name OHARACTER VARVING string	

Figure A.11 Maker Settings for First_Name

5. We need to create two additional fields, "Last_Name" and "MyPrimaryKey". Last_Name is the same as First_Name, a character field. You can duplicate First_Name, but insure you change the external name. No two external names can be the same within any table.

Although you can create a table without a primary key, this practice is not advised. So let's go ahead and create a primary key field.

6. With the window Untitled-1/Access/Names/MyPrimaryKey open, type an external name, then choose the external type, 'Integer' from the external type popup. Also choose int from the type popup. Double click on the Primary Key so that a check mark appears.

You might wonder why we are using a integer, as primary keys can be almost field type. The reason is that we will be using FrontBase's ability to automatically create primary key values. Alternatively, the designer would be responsible for this function.

	- Contraction				
ttribute	Value				
deptor	Front				
serLogin		calhost Dat	And a state of the		
chema Login onmant	Host loc	ally out Date	00.850		
Value of the local division of the local div	1.1.1.1.1			(mark)	
1	Untitle	st=1/Acces	ss/Names	민물	
Tana				RFT I	
Name			Untitle		BOOLEAN
Names				0=1/ACCES	
				d-LACCES	CHAR
		10		d-1/Acces	CHAR CHARACTER
AL	tribute	Vat		d-1/Acces	CHARACTER CHARACTER CHARACTER VARYING
	tribute temal Name	Vat		0-1/ACCES	CHAR CHARACTER
Ex				d-1/Acces	CHAR CHARACTER CHARACTER VARYING DATE DECIMAL DOUBLE PRECISION
Ex	temal Name		Attribute	0-1/ACCEN	OHAR OHARACTER OHARACTER VARVING DATE DECIMAL DAUBLE PRECISION FLOAT
Ex	ternal Name vessent		Attribute External Name	0-17ACEEN	CHAR CHARACTER CHARACTER VARYING DATE DECIMAL DECIMAL FLOAT INT
Ex	ternal Name		Attribute	0-17ACCEN	CHAR CHARACTER CHARACTER VARVING DATE DECIMAL DOUBLE PRECISION FLOAT INT ✓ INTEGER INTERVAL DAY TO SECOND
Ex	temal Name www.ent Name Fint_Name		Attribute E-demai Name E-demai Type	0-1/ACCEN	CHAR CHARACTER CHARACTER VARVING DATE DECIMAL DOUBLE PRECISION FLOAT INT INT INTEGEN INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH
	temal Nane www.ent Name First_Name Last_Name		Attribute External Name External Type Type	0-1/ACCEN	CHAR CHARACTER CHARACTER VARYING DATE DICINAL DICINAL DICINAL DICINAL INT INTERVAL DAY TO SECOND INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH NUMERIC
Ex	temal Name www.ent Name Fint_Name		Attribute External Name External Type Type Primary Key	0-1/ACCES	CHAR CHARACTER CHARACTER VARVING DATE DECIMAL DOUBLE PRECISION FLOAT INT INT INTEGEN INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH
	temal Nane www.ent Name First_Name Last_Name		Attribute External Name External Type Type Primary Key Allows null	0-1/ACCES	CHAR CHARACTER CHARACTER VARYING DATE DECIMAL DOUBLE PRECISION FLOAT INT VISTOCH INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH NUMERIC REAL
	temal Nane www.ent Name First_Name Last_Name		Attribute External Name External Type Type Pomary Key Allows null Width	0-1/42285	CHAR CHARACTER CHARACTER VARYING DATE DECIMAL DOUBLE PRECISION FLOAT INT VISTOCH INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH NUMERIC BEAL SMALLINI TIME VINE WITH TIME ZONE
	temal Nane www.ent Name First_Name Last_Name		Attribute External Name External Type Primary Key Allows null Writh Precision	0-1/42285	CHAR CHARACTER CHARACTER VARVING DATE DECIMAL DOUBLE PRECISION FLOAT INT INTEGER INTERVAL DAY TO SECOND INTERVAL VEAR TO MONTH NUMERIC REAL SMALLINI TIME

Figure A.12 Setting External Type

7. Close field and table windows so just the Untitled-1/Access window is open.

Exporting to FrontBase

1. Now we export our schema to FrontBase. This action writes our access objects (tables and fields) to the database, "NewDB". From the file menu, choose Export Schema.



Figure A.13 Exporting the Schema

2. The Export Schema window will open informing you of the impending action. You can update immediately or defer your export by creating a file. Export files may be viewed, edited or stored or applied at a later date.

Click the UPDATE NOW button.

3. Qilan automatically imports database schema to Qilan immediately after the export. This is performed in the event the database creates objects as the result of the Qilan export and helps to insure synchronization.



Figure A.14 Synchronizing the Schema

Creation of the Primary Key

- 1. To invoke FrontBase's ability to automatically create primary key values, we will need to return to FrontBase. If FBManager is not running, launch it.
- 2. Double click on the NewDB icon in the Monitored Databases window. The Connect Information window will open; click CONNECT.

Connect In	formation	
Host name:	locathost]
Database name:	NewDB	1
Database password:]
User name:	SYSTEM *	1
User password:		1
E.	Cancel Connect	1
		2
1 Manita	ored Databases	01
New08 localhost		
tor autor		
		÷
1		

Figure A.15 Connecting to the Database

3. After connecting, the Management window will open. Highlight '_SYSTEM' in the first column; then 'Tables'; followed by, 'Names'.



Figure A.16 Choosing the Names Table

4. With 'Names' highlighted, select Definition Editor from the Management menu.

D N D	rop IIM	lanagement	E
/ squar y user y Data Co		YEditor B&W	Show Log
Schemata	SYSTEM	Table	s
SYSTEM P INFORMATION_SCHEMA	Tables	EOMODELER, BELA	UIIONS
Monitored Da	tabases		

Figure A.17 Selecting the Definition Editor

5. Now we can modify our table fields. Locate the field named, 'MyPrimaryKey'. Note how it is an integer, just as we created it in Qilan. Type in the word, "UNIQUE" in the default column. This keyword invokes FrontBase's automatic key value generation.

Commit Changes

6. Now close your windows. Click COMMIT when FrontBase asks if you want to commit your changes.

ame	Datatype	Default	
	VARCHAR(50) 3		_
	VARCHAR(SR) -	G	
EV	INT ·	UNIQUE	
Columns	MYPRIMARYKEY		
Columns		Type	
	EV Y <u>UNIQU</u> Name Columns	VARCHAR(S8) -	

Figure A.18 Commit the Table Changes

The User Interface

Returning to Qilan's Untitled-1 window, drag out a Framework icon from the palette. The Framework is used to build your user interface forms and create most of your database logic.

	9
Name	Uses
Access	D
Framework	D

Figure A.19 Build UI through the Framework

Building the Framework

Although we can create fields in our Framework, just like we did in Untitled-1/Access/Table, it might be easier if we could copy, then paste them directly into the Framework.

1. To accomplish this, open the Untitled-1/Access/Names window and highlight the fields. Choose Copy from the Edit menu.

Undo New Framework %2		Untilled-Laprj	町
Cut #3	n	and the second s	-
Copy R.			
Puste 80			3/
Paste into -0.87 Gear	4		
Duplicate 80	· Name		Uses
Select All 80			0
Spelling W:	D 10	titled-1/gprj/Access	E
Check Spelling 82			1
Project Settings			1000
			1.000
175			-
	Attribute	Value	
Field	Adaptor	FrontBase	
	User Login	Host: local host Datab	9666
	Schema Login	Host local host Data!	base:
	Connect		
Relationship			
	A Name		Uses
	EOMODELER_R	ELATIONS	0
~	Names		0
DataFlow	TT Contribu	d-Lopg/Access/Names	
		a reflets second and a	
			1000
			11-012
	Attribute	Value	-
	External Name	Names	
	Convert		
	and the second se	and the second sec	1
	Contraction of the local division of the loc		[tree!
	Name		Uses
	First_Name		0
	Lust_Name		0

Figure A.20 Copy and Paste the Fields

2. Now open the Framework by double clicking on its icon in the untitled window. Select Paste from the Edit menu.

NOTE: If you double click on a field within a Framework, only the comment line appears. This is because Framework fields are used to hold data during qilan.cgi execution and do not actually represent database storage fields.



Figure A.21 Pasting the Fields

Modify Project Settings

Before we proceed, we need to set a few project settings. These settings control how Qilan will handle data, where webtemplates are exported and what type of dtd (document data definitions) will be used.

1. Select Project Settings... from the Edit menu. The default settings appear acceptable, but we want to treat empty values as undefined. Double click this line until a check appears.

Why do we want to do this For most database designers, a field value is either defined or undefined. This is in contrast to a submitted html input value, which may be defined, empty or undefined. For our purposes, let's use the database designer's perspective.

2. Closing this window saves the settings.

Project Se	ettings for "Untitled-Lapry"
Attribute	Value
Beautify names on import	
Export WebTemplate on Clo	
Export All WebTemplates on	
HTML file root directory	/Local/Library/WebServer/Documents
HTML file extension	gin
Iteration limit	200
Recursion limit	100
Command Timeout	60
New WebTemplate dtd	html32 ÷
Log Errors To	(2), (2), (2), (2), (3), (3), (3), (3), (3), (3), (3), (3
Error Message	
Treat Empty as Undefined	4
Debug SQL	none :

Figure A.22 Save the Settings

Returning to the Untitled-1/Framework window, our first step is to create a user interface form.

Template Construction

1. From the palette, drag a webtemplate into the framework window. Rename the default name to, "Names", then double click to open the window. As specified in Project Settings, the webtemplate uses a html 3.2 dtd. Qilan will conform the webtemplate construction to the chosen dtd, including available html tags and their placement. In other words, Qilan prevents syntax errors.

To use a different dtd, just change the dtd in Project Settings before dragging out a new webtemplate.

As you construct your webtemplate, observe that Qilan does not require closing tags and uses an outline structure.

The "HTML file" is how this webtemplate will be referred to by client users. Type, "Names". The URL will be something like this:

[hostname]/cgi-bin/qilan.cgi/[HTML file] localhost/cgi-bin/qilan.cgi/Names

	_	Untitled-Lapej = /Local/Public	CB.	9
		Untitled-1.qprj/Framework	08	Mac
			0.00	
1		Name	Uses	
-		First_Name	0	
		Last_Name Names	0	
	6	Untitled=Lqprj/fra		26
				teni22
		Attribute Value		
		HTML file Names Connent		
		a <html> o<head></head></html>		
		o «TITLE» o «BOOV»		

Figure A.23 Construct the Template

2. Using the html tags from the palette, select tags that permit the submittal of data, such as 'form' and 'input'. In this tutorial, we have chosen to use a table structure.

Modify Tag Attributes

1. Most tags allow for attributes. Highlight a specific tag, then select the attributes submenu. Double click directly on the webtemplate (adjacent to the attribute) to enter a specific value. As you will see later on, you can also insert calculated or database values to create dynamic attributes.

9] Untitled-La	prj/Fram	ework/Nas	nes	E	
-	⇒						
Attribute	Value						
HTML file	Names						
Comment							
O-SHTML>							
D «HE/							
	TITLE=						
	o Nanes						
0-BO							
0<	FORM>	nettod		post			
	d <table></table>	action		Names			
	Ø-TR>						
	Ø×TD+						
		t Namo					
	O-TD-						
	D-dN	PUT>	type		text	+	
			TIRE		First Name		
	Ø-TR+						
	o <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
	OLA	dName					
	Ø <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
	D-dN	PUT>	type		text	•	
			name		Last_Name		

Figure A.24 Modify Tag Attributes

- 2. Now for some creative elements. Not wanting to have more than one window, let's show the records submitted (and created in the database) as we create them. This combines submittal and display elements on the same form.
- 3. After the html FORM tag in the webtemplate, drag to insert a QTABLE tag from the palette. 'Q' tags are special Qilan processing elements that extend functionality or reach directly into the database. The QTABLE tag combines the formatting

of a traditional html table tag with an interactive retrieval function.

- 4. Once the QTABLE is dragged into the webtemplate, the attribute, 'fields' will appear. Double click on this attribute; the field list window will open.
- 5. The field list window asks, 'What Access', 'What Table', What Fields'. Choose, 'Access', 'Names' from the pulldowns at the top, and then select all the fields by clicking in the column to the left of each field name. Click the OK button.

	Field List	Names	EDE	
 	Access Access Table Names	•	Tana ar	
Attribute Value HTML file Names Comment	Tata Anacus			
o <html> o <head+ o <tttle+ o Namos d <bodv> o <form> o <tabl o <tt< td=""><td>✓ Fint_Name ✓ Lest_Name ✓ MyPrimaryKey</td><td></td><td></td></tt<></tabl </form></bodv></tttle+ </head+ </html>	✓ Fint_Name ✓ Lest_Name ✓ MyPrimaryKey			
0 T+0 0 0	Cancel OK <td> d<input/> type</td> <td>Text</td> <td>•</td>	 d <input/> type	Text	•
0 eHP5	name	Last_Name		
Ø <gtables< td=""><td>- fields</td><td></td><td></td></gtables<>	- fields			
D <tr></tr>				

Figure A.25 Choosing the Field List

6. The selected fields will then appear within the QTABLE. Although we are not using a query for this tutorial, queries of considerable sophistication can be constructed without the need for SQL syntax.

- 7. As we build our QTABLE, we insert traditional html table tags, such as table rows (TR) and table data (TD). From a formatting standpoint, we want to display certain database field elements. This is performed by using a QVALUE tag; meaning, "Display this value here".
- 8. From the palette, click on the Values tab, then choose the Access and Table from which the field values will be selected. Drag the fields to the webtemplate as QVALUE attributes.

Falelle	Divelori.	apri/Transput/Names	E)#
/ Tegs / Values / Links /			-
Toble Names	attravate Value		10412
Toble Names	HTM, Re Names		
	Carnet		
Find_Ivana Last_Name	extrates		
WyPrim bryPldg	afrener		
	a-EODV+		
	-D-FORM+ MIDIOS	good .	
	a-TABLE-		
	0+779+		
	ø-tD+		
	U Fed Nana		
	a-(TD-		-
	D-SBPUT>	type best name Find Name	
	a stra	time to the time t	
	g-4TD+		
	a Last Name		
	a-cTD+		
	C+SPUT-	type text rome Lot_Nens	•
	Land Sec. 2010		
	e-GTABLE> Neth	AccessNeeseTist_Here AccessNees_NeesLec_Nees AccessNeesLec_Nees	128 J. J.
	g 478+		
	¢∢TD+		
	CHOWALLEY	Ican Access/Italies/	First_Name
	Ø-TD-	National State	
	a-gvalue-	Kan Katurate	

Figure A.26 Selecting Field Values

The basic construction and formatting are finished. We have created an entry form and a mechanism to retrieve and display the data. What remains is the logic for moving the data to the database (upon submittal) and some method to retrieve selected data for editing.

Processing Q Tags

A Qilan webtemplate processes 'Q' tags in a hierarchal order, meaning top to bottom. This allows the designer to create logical constructs. In the following screen, we have added several 'Q' tags to create data when certain conditions are met. Let's cover them one by one.

The QLOGIN tag

The QLOGIN tag instructs Qilan to log into FrontBase. The login is handled by the settings in our Access. Qilan can log into as many databases as necessary, there is no limit.

The next 'Q' tag is an 'if' statement. When the 'valid_entry' abacus returns "yes", we proceed to the QUPDATE, otherwise this is skipped. The QUPDATE performs an 'insert' if a link does not exist, otherwise an 'update' is performed.

When is this logic performed Each time the client accesses the html file, "Names".

NOTE: The FORM action is, "Names". This means that each time the submit button is pressed the html file will be reloaded and the Qilan tags will be processed.

0	1	Untitled-Lapri/T	ramework/Names	08
-	⇒			
Attribute	Value			10000
HTML file Comment	Names			
	nitle» o Names		-	3
0.4	QLOGIN>	databasename	CC011	
	GFBLOCK> o <cifthen> .o<cupdate></cupdate></cifthen>	icon create dataflow	A valid_entry	
	FORME	method action	post. Names	:

Figure A.27 Modify Access Settings

The validation is simple, but prevents an empty database record. Valid_entry tests for the presence of a First_Name and Last_Name. An abacus is employed, with 'defined' operators selected from the palette.

		100
E	Neter	Uses
k	hasf	
2	valid_entry	
	Pirst_Hatta	
-	Lait Name	
	MyPrestyter Tranet	
5	the second s	-
8	Unitted-Lapit/Transwork/valid.ext	Y 88
		4
9	sulpuis 19835 virien beth first and lost name are defined	4

Figure A.28 Validation Tests

The QUPDATE

The QUPDATE uses a dataflow object. This object instructs Qilan to insert or update target values (those existing in the database) from source values (those submitted by the user). Dataflows are always based on a link, or relationship.

NOTE: We are leaving 'MyPrimaryKey' undefined. This is because we have requested FrontBase to automatically value this field.



Figure A.29 The QUPDATE Validation

The relationship "link_names" used in the middle of the above dataflow defines matching or unique values to be used as linkage. In this relationship window below, we are saying that when ((First_Name = 'First_Name) and (Last_Name = Last_Name)) we have a state of uniqueness. Dataflows (triggered by QUPDATE) will create a new database record when the relationship does not exist.

	_ Untitled-Lapri/Tr	amework/link_Names
new record being c		the first and last name. This will result in a drive last name is different from existing
values.		-
€ One-To-One	O One-To-Many	
🖲 One-To-One	O One-To-Many	Access SACCESS Table Names Show All fields
@ One-To-One	O One-Te-Many	Table 📷 Names
	O 0ne-Te-Many	Table Names
Source	One-To-Many	Table Names Show All fields Target

Figure A.30 Defining Relationship Validation scenarios

Accessing Existing Data

The next part of the tutorial covers accessing existing data for editing.

Before we start, consider we have already created a QTABLE to retrieve our data. Rather than creating new structures, let's just use what we have. A web-centric approach typically involves the use of links. The user clicks the link and the record should automatically be retrieved for editing.

To perform this type of action, we will need to obtain a unique link for each record created. Let's use MyPrimaryKey.

- 1. First, we double click on the QTABLE fields attribute in the webtemplate and select MyPrimaryKey. This retrieves the value.
- 2. Second, we create a new framework field, MyPrimaryKey. Other framework objects will use the value of this icon.
- 3. Third, we value this new framework icon using the value from the database table. Note how QASSIGN is used to accomplish this within the QTABLE. For each record retrieved by QTABLE, the framework icon will be valued by MyPrimaryKey. This will give each record a unique value.

8	3 Untitled-	-Lapri-Tra	meneral/s	Australia II	16
>				-	ł
Attribute Value	_	_	_		'n
HTML file (Nahe).					۲
Conmert					1
2.		_			С
O STITLES					П
or fillings					L
0-800%					
D-FORMA	mathing.		pert.		
a starter	actor.		Normats		
d+148.E>					
d (Th)					
oktore a Find to					
entite.	Line				
4-99-01	199	April 1		(test (4)	
10000		1000		Fett flame	
134TH+		10000		0.0000000	
e-m-					
clastic					12
#+TD+					18
0-8870	1	200		test =	H
The second se		2,502.0		LateName	
D+UTABLE>	THE R		-		
Dettimeter	1000			uhlaneuFind_Nane uhlaneut.nit_Name	
				UNIXABLE UNIVERSITY OF STREET, STRE	
d-(TR)-			12000		
d v Tijbe					
#=0A5809	ile .	1008		Mot marshay	T
E-STALE		1004		Accessifianes/MyPrman/Fay	ł
B-TD-	-	1144		D ICCONTRACTOR	
S-STALLE		1000		Accessivenes/Last_Mane	11.
		1000			٠

Figure A.31 Accessing Existing Data

4. To create a html link, we need to use the 'A' tag with a href value attribute. We have chosen to make First_Name the clickable element.

E 0	Unithed-Lope	/Transverse	Alatari E	11
< ⇒			-	ł
Attribute Vetor	_	_	1944	i
HTML file Pulaties				ï
Conmont				1
1				Ľ
- Names				T
4+8001+				I
0-P09M-	witted active	Norman I		ł
dia TABLE-	1000			I
11+78+				
a-m-				I
o Find Nat	-			I
DeTHE				I
1-89PST-		P7	Text 2	I
0.478+				I
p <td+< td=""><td></td><td></td><td></td><td>I</td></td+<>				I
cl.astrian				I
D-TD+				I
1.000	. 1	H.	Test #	I
		att a	Last_Name	I
C et El-				I
Ø KUTABLE-	Net de la	Acce	uchanecFint_Name	l
o-m-		- ACC.	cuhiana uMyPranayilay	I
d-TD-				I
D-CALSENNA	1004		MuPlease Nev	I
	Vitiat	() () () () () () () () () ()	AccessitianestayFeaseres	I
0.00	head			1
# +GWALL	E- 6	80.5	ALCONTRANOV FILMEN	I
8+70+				ŀ
E-477ALLE>	1004		Access/Ivenes/Last_Name	ŀ
				Ľ

Figure A.32 Creating the HTML Link

5. The href value attribute is constructed as a dynamic element. This abacus is a formatted 'GET' request that reloads 'Names' and submits the value of MyPrimaryKey.

E Cretitied-Lapri/Transvark/Irmf El
4 -
This creates a "get" repared when the W link is clicited. The carried form is relocated and the field, "MyProvary/sey" is then valued.
("Harriss" Addresed by "P" Addresed by "MyPhicaryRey" Addresed by "+" Indicess by MyPhicaryRey (

Figure A.33 Setting the href value attribute

6. With the webtemplate open, select the abaus, "href" from the palette and drag it to the href value attribute.

8		Outtet	-Laptj-Tex	mewark/San	111	8
-	•				-	ł
Attribute	Value	_	_	_		ñ
	Nahet.					т
Conmort						1
						I.
S	- NAMES		1			Т
4-800	Wa .					н
0-1	PORM-	action in the second		peril Nomes		h
	#+TABLE+					н
	0.4778					
	a-ms					
		t Name				
	Della					
	11-89	PETH	het		Test 2	
			1904		Fint_Name	
	10-TR-					
	D-(TD+					
		triane				
	Della.		2.50			
	1000		1000		Last Name	Т
100	1971-				care num	
	UTABLE>	Tel al		AccesuA	steacFirst_Name smecKast_Name anexMyPressydies	
	o-tR-			State of the local division of the		
	g-30+					
		10Pie	ICCR VALUE	8	MyPanarysey AccessitianauthyPressyllary	
	Dicko		funt.	2	trat	
	0-01	WALKER	6.81		AccessTamon/Fact_Name	
	diatter					1
	S-4074AL	UE>	ACC N		Access/Venes/Last_Name	

Figure A.34 Drag the HREF Attribute

7. How will the value of 'MyPrimaryKey' be used HTML inputs permit default values, also known as the 'value' attribute.

Let's default these input values to the record being selected from the list.

8. To do this, will be need to obtain First_Name and Last_Name. With MyPrimaryKey in hand, we can create a relationship to our table.

The relationship below has been named "link_key".

	Contitled-Lapry/	framework/link_Key
		Names to the existing Primary Key. The the value of the Primary Key by the
🖲 One-Ta-One	O One-To-Many	Table 📷 Names
0		Show All Fields
		A second
Source		Target
Source		First_Name
Source M		

Figure A.35 Creating the Relationship

Choose the Value to Retrieve

Based on this relationship, we merely select the value we wish to re-trieve.

1. From the palette, we first choose the link_key relationship at the top of the window, then drag the First_Name field into the abacus window.



Figure A.36 Identify the Relationship

2. With the webtemplate window open, we select the abacus we created above, 'get_first_name', and drag it to the appropriate html input value. Now, when MyPrimaryKey is valued, Qilan will establish the link and return the desired value.

		Citation	h-Laprj-Fra	darwark/	Names	116
(=	->					-
Attribute	Veloc					
HTML file	Triales.					1
Conmert						
			_			100
0.	FORME	actes		post. Names		
	@ <table></table>					
	delle D					
	@=TD=					
	oPhili	44790				
	ØKTD+					
	0-8970	Tr-	101.0		dest fit rans	
			304		text 0	
			1000		Past_Nono	20 2
	0.000					
	e=TD=					
	CLARIN	107.8				
	D-CTD-F				100000000000000000000000000000000000000	
	CARPS	Er-	1.01.0		Cotlations test t	(A)
			Nor.		Last_Name	
	0.25		2000		CallGoans	
	g-rD+		-			
	Parily.		calips .		200	
	0.00	Ts.	her		satered a	
			rah.m		Enter Name	<u>^</u>
L.,	1.00					

Figure A.37 Establish the Link and Return Value

Exporting the Webtemplate

1. Our final step is to export the webtemplate for access by internet clients. Select Export HTML from the File menu.

a file Edit Icon Window Services		_
Gen. No Di	Outlied-Laps/Transwork/Names	DE
Close MS Server Als. UNS E Server Te UNS E Server Te	Attribute Value HTM, Ro Hanst Connect	-
Export A& HTML Page Setap. URP Print. RP	o-FORM- method post	
Quet #0	O-TRELE-	
land.	D-(TR)	
velic, entry	a+TD+	
	d Faithann di-TD-	
	civiteTut> value & pot first name	
	type text 4	
	rane Fertinane	1
	asile.	
	of Last Name	
	s-TD+	
	ci-chiPUT> value di pet_list_name	
	type text 4	
	Note: Lot_Name	
	dieTDe Bign right Longos a	
	D-dBPUTe type sales 5 value EnterName	

Figure A.38 Export the webtemplate

Here is how our form looks from the browser. Note the path.

3	Names	8
err mb f	ttp://ocalhost/cgi-bin/gilan.cgi/Names) 🧊 ୭ 📦
Pirst Name Last Name	Enter Name	

Figure A.39 The Viewable Form

2. Enter 4 or 5 names. Try entering some names without a first or last name. If the validation was performed correctly, nothing should happen. Also observe how the first name is highlighted as the link.

	Netra
derr on	ener Statution de California a california
Pere Nam Lest Nem	*
MAR	Mr. Naenara
Ling	

Figure A.40 Testing the Form Validation

3. Move the cursor over the first name in the Names window so that your web browser displays the link. This is the result of the href abacus.



Figure A.41 Display the HREF Value

4. Clicking the link will submit MyPrimaryKey and reload Names. Qilan will perform the link (according to the relationship) and default the input values.

der and and a state	
	ss: 🚡 http://localhost/cgi-bin/gilan.cgi/Names?MyPrimaryKeys1i -
Pirst Name	Stanzon
Last Name	Broderick
	Enter Name
Stephen C	aine
	(c Namara
Mike P	rench
Shannon B	røderick
Larry A	tkin

Figure A.42 Display the Link Values

NOTE: FrontBase may require specific field typing. If you receive errors, use an abacus to convert MyPrimaryKey to an integer, then use this abacus as the relationship link.

	Chtitled-Laprj/Framework	9 9 9 9
I	Name	Uses
1	get_first_name	1
d	get_last_name	1
1	tref	
4	in(key)	1
ł	valid_entry	1
	Untitled-1.uprj/Framework/int(key)	
	FrontBase likes all datatypes to match, so we will happin comply by converting our framework value to an integer	2
	(our Primary Key datatype)	
	(Indepen My Primary Key)	

Figure A.43 Convert MyPrimaryKey to an integer

5. And the new relationship...

	D Untitled-Lapri/	Framework/link_Key	E
			~
		Names to the existing Primary Key. The value of the Primary Key by the	The
🖲 One-Ta-One	🔿 One-To-Many	Access Access Table Names	
Source		Target	
*		First_Name	1
		Last_Name	
🛆 int(kay)		MyPrimaryKey	
			_

Figure A.44 Display the New Relationship

Index

0

Overview 5

Т

The Database 5 The Schema 11 The User Interface 19 Index