

# Getting Started with FORM

*R1.0*



## **ChiMu Corporation**

1220 N. Fair Oaks Ave, #1314  
Sunnyvale, CA 94089

Phone: 408 734-9068  
Email: [info@chimu.com](mailto:info@chimu.com)

[www.chimu.com](http://www.chimu.com)

Copyright © 1997, ChiMu Corporation. All rights reserved.

## Getting Started with FORM

This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by ChiMu Corporation. ChiMu Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of ChiMu Corporation.

## *Table of Contents*

<b>1</b>	<b><i>Introduction</i></b>	<b>1</b>
1.1	Distribution and Installation	1
<b>2</b>	<b><i>FORM Components</i></b>	<b>1</b>
2.1	Java Classes	1
2.2	Documentation	1
2.3	Examples	2
2.4	Included Source	2
2.5	Shell/Batch files	2
2.6	Component Directory Summary	2
<b>3</b>	<b><i>Required Manual Installation</i></b>	<b>3</b>
3.1	Jar files and Class Paths	5
3.2	Shell/Batch files and Executable Paths	5
<b>4</b>	<b><i>Setting up the Examples</i></b>	<b>7</b>
4.1	Using the ExampleRunner	7
4.2	Creating Databases for the Examples	8
	Creating on the Server	8
	Creating an ODBC source (if needed)	8
	Testing the FormExampleDatabase	8
4.3	Choosing Schemes and Database Population	8
4.4	Running the Examples	8
	Tracing	8
4.5	Using the Local Examples	9
4.6	You are Done	9
<b>5</b>	<b><i>Setting up the Examples-2</i></b>	<b>11</b>
5.1	Using the DatabaseTester	11
	ODBC Driver and Database	11
	Direct JDBC URL	11
5.2	Creating Databases for the Examples	11
	Creating on the Server	12
	Creating an ODBC source (if needed)	12
	Testing the FormExampleDatabase	12
5.3	Setup of a FormExampleDatabase	12
5.4	Running the Examples	12
	Recompiling the examples	12
5.5	You are Done	13
5.6	Other Topics	13
	Using the FormDatabaseTester	13
	Running Examples from you own program	13
<b>6</b>	<b><i>Where to go next</i></b>	<b>15</b>
6.1	ChiMu Corporation References	15



# 1 Introduction

---

Thank you for purchasing FORM. FORM is a superior object-relational mapping framework for Java and it will help you build better applications faster. This document describes the FORM release, its components, and how those components are installed on your system. It also describes how to setup FORM's examples to work with your system and database.

## 1.1 Distribution and Installation

FORM is currently shipped as one 'zip' file, which will expand into a single root directory named 'chimu' with multiple subdirectories containing the FORM Components. This root directory can be placed anywhere, but the paths to the executables and the java class files must be set appropriately. We will refer to the full path of the root directory as './chimu'.

Currently manual installation is required for Class Path and Executable Path updates. This will be changed in future releases.

# 2 FORM Components

---

The FORM product includes the following components:

1. Java Classes
2. Documentation
3. Examples
4. Java Source
5. Shell/Batch files

## 2.1 Java Classes

FORM is a Java based product and all of FORM's classes are packaged into two 'jar' files: 'Form.jar' and 'FormTools.jar'. 'Form.jar' contains the main FORM functionality and will be required by your deployed application. 'FormTools.jar' contains classes that are development tools and can not be released with the deployed application. The two jar files together comprise the development environment for FORM and must be located in your Java classpath (or similar configuration for your development environment). FORM's Jar files are located in './chimu/lib'.

## 2.2 Documentation

FORM comes with manuals, supporting documentation, and Javadoc API documentation. The following list some of the documentation included with FORM:

1. Getting Started with FORM (this document)
2. Introduction to FORM
3. Learning FORM
4. FORM Tools
5. Kernel Frameworks
6. FORM, Dome, Kernel API (Javadoc format)
7. ChiMu's Java Development Standards
8. Foundations for Object-Relational Mapping

All the documentation is in './chimu/docs/'. The manuals and supporting documentation is in PDF format and HTML format [Some HTML versions are currently not included]. The Javadoc API is in './chimu/docs/api/' in HTML format.

## 2.3 Examples

FORM is distributed with multiple example programs that demonstrate using FORM. These programs are located in ‘.../chimu/examples/form’ and setting up and running these examples is discussed in a later chapter.

## 2.4 Included Source

Besides the Example source code, the FORM standard distribution includes source to some of FORM’s functionality that you may want to modify. The source code is included in the directory ‘.../chimu/src’ under the ‘com/chimu/’ subdirectory. See your licensing agreement for restrictions on source use and alteration.

## 2.5 Shell/Batch files

The FORM distribution includes shell/batch programs to simplify the command line typing when using FORM tools. These are only convenience programs which call the Java VM to do all the work. The batch programs currently include:

FormVersion	Prints the version numbers of the FORM subsystems.
FormDatabaseTester	Logs into the database and runs examples and tests.
FormPreprocessor	Runs the FORM Preprocessor

See the FORM Tools manual for more information on this functionality.

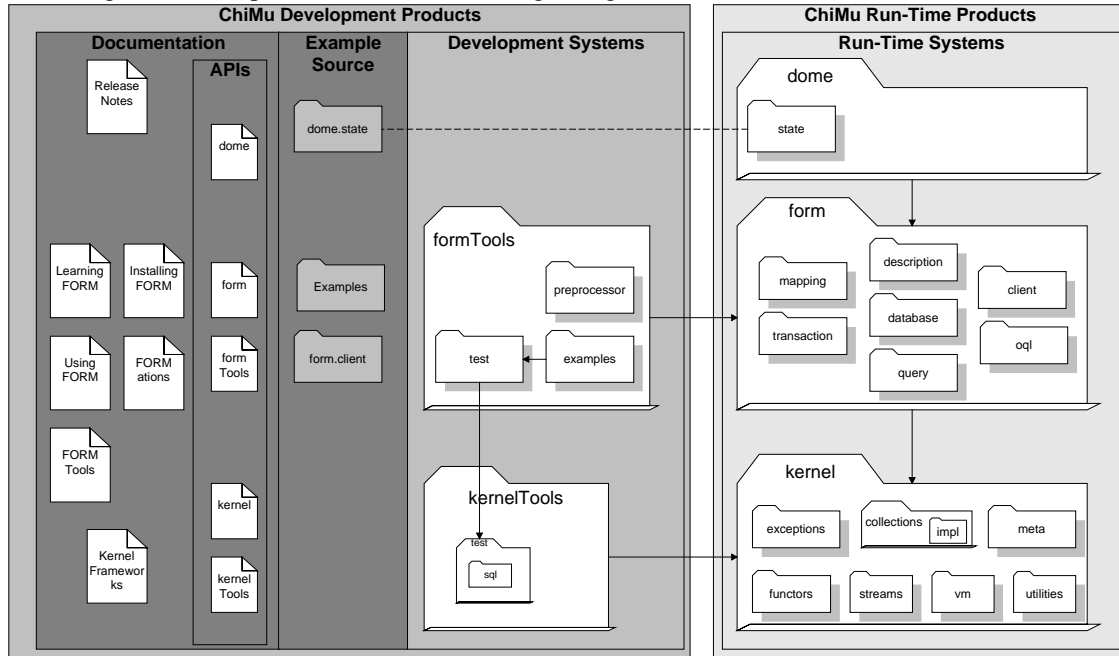
## 2.6 Component Directory Summary

Summarizing the above, FORM’s components are placed in the following locations:

.../chimu/bin	Shell/Batch files
.../chimu/docs	Documentation
.../chimu/docs/api	API Javadoc
.../chimu/examples/form	Examples
.../chimu/lib	JAR files
.../chimu/src	Included Source

## 2.7 Distribution Map

Collecting most of the pieces of FORM into a single diagram looks like this:







## 3 Required Manual Installation

---

The current FORM distribution requires manual installation to get you Java environment to know about the FORM jar files [MS-Windows will change to InstallShield]. You will also need to set up executable paths to include the ‘.../chimu/bin’ directory if you want to use the shell/batch files included there.

### 3.1 Jar files and Class Paths

The standard Java JDK uses a ‘CLASSPATH’ environment variable to know where to look for Java classes. For an application which uses FORM this classpath will need to include:

.../chimu/lib/form.jar (or for DOS ...\\chimu\\lib\\form.jar)

Where the ‘...’ is replaced with your installation location for the FORM distribution.

For development you will also need to include:

.../chimu/lib/formTools.jar

in the class path.

Some development environments (e.g. Symantec’s Café) do not use the ‘CLASSPATH’ environment variable and instead use configuration specific to the development environment. Please see your development environment manuals for difference from the Java JDK installation process.

If the class path is set up correctly you should be able to execute both of the following. ‘FormPack’ will provide version information, and ExampleRunner is an application for running the FORM examples (see the next chapter).

```
> java com.chimu.form.FormPack
> java com.chimu.formTools.examples.ExampleRunner
```

### 3.2 Shell/Batch files and Executable Paths

FORM includes command line components in ‘.../chimu/bin’ which make using FORM from the command line easier. These are all shorthand for interacting with the Java VM directly, so instead of typing

```
> java com.chimu.form.FormPack
> java com.chimu.formTools.examples.ExampleRunner
> java com.chimu.formTools.test.DatabaseTester DatabaseName TestClassName
```

you can type

```
> FormVersion
> FormExampleRunner
> FormDatabaseTester DatabaseName TestClassName
```

To set up the shell/batch files to work with your operating system, you will need to modify you ‘PATH’ to include the ‘.../chimu/bin’ directory.



## 4 Setting up the Examples

FORM includes example programs that can be run from a command line or any program that calls the Java VM with arguments. Because FORM is a database product, it will need a database to use for its examples. To work with you database FORM will need two things: (1) A database area to put the example scheme and data into, and (2) a JDBC connection to that database. FORM uses JDBC and has a database product configuration layer on top of JDBC, so it can create a scheme within and populate most database products.

There are two approaches to interacting with FORM's examples. You can either use the command line based FormDatabaseTester or a GUI application ExampleRunner. The ExampleRunner is simpler and the DatabaseTester is more flexible. The actual examples and the concepts for using the examples are the same no matter which program you use.

### 4.1 Using the ExampleRunner

The ExampleRunner is a GUI application to work with examples. It allows you to connect to the database and run FORM examples within a single window. To launch the ExampleRunner use:

```
> FormExampleRunner
```

or if you do not have '.../chimu/bin' in your path:

```
> java com.chimu.formTools.examples.ExampleRunner
```

This will bring up the main ExampleRunner window:

The screenshot shows the 'FORM Example Runner' window. It contains the following elements:

- Database Section:**
  - Driver:
  - Product:
  - Name/URL:
- Login Section:**
  - Name:
  - Password:
  -
- Examples Section:**
  - Root Directory:
  - Scheme:
  - Example list:  (with scroll arrows)
  - 
  - 
  - (with dropdown arrow)
- Results Pane:** A large empty rectangular area at the bottom of the window.

The window is divided into four sections: Database and Driver information, User login information, Example selection, and the Results pane.

## 4.2 Creating Databases for the Examples

Once you have the ExampleRunner working with your database server and an existing database, you can create one or more databases to hold the FORM example schemes and data. The examples use several different schemes that are different from each other, so you can either use and reuse a single database for all the examples or setup multiple databases with one database for each example scheme. FORM handles the scheme creation and population for you, so reusing a single database only has the disadvantage that examples from different scheme can not be run at the same time. We will assume you are using a single database for all the examples that is named 'FormExampleDatabase'.

### Creating on the Server

You will need to create a new database 'FormExampleDatabase' on your database server and specify a user with the abilities to create scheme objects (e.g. Tables) within that database. See your database product manual for more information on creating databases and user login accounts.

### Creating an ODBC source (if needed)

ODBC requires creating a source for any database you need to access. If you are using ODBC then you must create an ODBC source 'FormExampleDatabase' that connects to the database you created above. This is normally done through the ODBC Control Panel.

### Testing the FormExampleDatabase

To make sure the database is working and accessible you can now try to login to it through the ExampleRunner. Specify the DatabaseName and your login information and then press the 'Test Login' button. If the login is successful this will return database catalog information.

If the login is not successful, check to make sure you have the right login name and password. You will also be given information on the full JDBC URL that was used to log in and the Database driver. This will allow you to verify that the ExampleRunner is producing the connection that you want. JDBC drivers can be especially finicky.

## 4.3 Choosing Schemes and Database Population

After you can successfully log in, you can then focus on running Examples. By default, you will be running the examples included in the FormTools.jar. This will have several schemes, which have different database tables and different population data. Choose one of the schemes from the drop-down list, and click 'Populate Database'. This will remove any previously existing example tables from the database, put the new tables into the scheme, and populate the database.

## 4.4 Running the Examples

After you have populated the scheme you can select one or more examples from the right hand list and run them with 'Run Examples'. If you select 'scheme1' and 'PersonRetrieval\_1' you should get the following output:

```
Test: ---- Ex1_PersonRetrieval_1 ----
<Person#99 Regina Cramer>
Regina Cramer is 64 inches tall and can be contacted at cramere@org.com
```

The actual data may be different because the Person selected is arbitrary.

### Tracing

You can turn on tracing to get more information about the execution of FORM for a particular example (except PersonRetrieval\_1, which has tracing disabled). Turning tracing to 'Level 2'<sup>1</sup> produces the following output for PersonRetrieval\_3:

<sup>1</sup> Be somewhat careful about over-tracing (especially level 3 with large result sets).

```

Test: ---- class schemel.Ex1_PersonRetrieval_3 ----
Test: **875613401393**
Driver Product    = JDBC-ODBC Bridge (SQLSRV32.DLL) [1.1001 (02.65.0213)]
Database Product = microsoft sql server [06.50.0201]
Recognized Database Product is: Microsoft SQL Server
FORM: Build ObjectInitializer From slots:
      [0:oid][1:name][2:email][3:height]done.
Initialized: <Person#11 Patricio Steel>
Patricio Steel is 70 inches tall and can be contacted at steelpa@org.com

```

## 4.5 Using the Local Examples

By default the ExampleRunner uses examples that are stored in the FormTools.jar file. You can also tell the ExampleRunner to use local examples by specifying a 'root directory'. The root directory is the directory above each of the example directories, so should be '../chimu/examples/' for the default installation. You must also have the root directory in your class path for it to find and execute the examples.

## 4.6 You are Done

You are done. The examples work and FORM is working. You can now test out more examples, read the FORM manuals (see the final chapter), look through the example source, and create your own test examples.

If you have any problems with the above, try to make sure each step is working before going to the next. For example, you can use JDBC to talk to the database, you have a FormExampleDatabase with sufficient privileges to create tables, and you can login to the FormExampleDatabase. If you continue to have difficulties, please contact us.



## 5 Setting up the Examples-2

---

The previous chapter discussed setting up the examples in terms of the ExampleRunner. This chapter shows setting up the examples using the command line based DatabaseTester and is an alternative presentation to the previous chapter.

### 5.1 Using the DatabaseTester

The first step to setting up the Examples is to get the 'FormDatabaseTester' working with your database. Full documentation on the DatabaseTester is provided in the FORM Tools manual, and you can get information on its command line options by calling it without any arguments:

```
> FormDatabaseTester
```

or if you do not have './chimu/bin' in your path:

```
> java com.chimu.formTools.test.DatabaseTester
```

Here we will show how to use it in the context of the examples. The FormDatabaseTester connects to a database and then runs one or more test programs that will use the jdbcConnection the DatabaseTester created. The sole responsibility of the DatabaseTester is to login to your database properly. For this it needs to know either the driver and database name or a full JDBC URL to the database.

#### ODBC Driver and Database

The simplest variation of using the DatabaseTester is if you are using ODBC with a preexisting ODBC source. This is the driver the DatabaseTester defaults to, so all you have to tell it is the name of database source. For example:

```
> FormDatabaseTester OdbcSoureName
com.chimu.formTools.test.ConnectionInfo
```

will give you information about the connection to the OdbcSoureName assuming it does not require a login name and password. In the case that a password is required, the arguments '-l' and '-p' specify the login name and password respectively.

```
> FormDatabaseTester -l loginname -p password OdbcSoureName
com.chimu.formTools.test.ConnectionInfo
```

#### Direct JDBC URL

You can also directly specify a JDBC driver class and the full JDBC URL by using '-drc' to specify the class and using a full database URL as the database name. The following is this URL approach to calling the sun supplied JdbcOdbc driver:

```
> FormDatabaseTester -drc sun.jdbc.odbc.JdbcOdbcDriver
jdbc:odbc:OdbcSoureName
com.chimu.formTools.test.ConnectionInfo
```

### 5.2 Creating Databases for the Examples

Once you have the FormDatabaseTester working with you database server and an existing database, you can create one or more databases to hold the FORM example schemes and data. The examples use several different schemes that are different from each other, so you can either use and reuse a single database for all the examples or setup multiple databases with one database for each example scheme. FORM handles the scheme creation and population for you, so reusing a single database only has the disadvantage that examples from different scheme can not be run at the same time. We will assume you are using a single database for all the examples that is named 'FormExampleDatabase'.

## Creating on the Server

You will need to create a new database 'FormExampleDatabase' on your database server and specify a user with the abilities to create scheme objects (e.g. Tables) within that database. See your database product manual for more information on creating databases and user login accounts.

## Creating an ODBC source (if needed)

ODBC requires creating a source for any database you need to access. If you are using ODBC then you must create an ODBC source 'FormExampleDatabase' that connects to the database you created above. This is normally done through the ODBC Control Panel.

## Testing the FormExampleDatabase

To make sure the database is working and accessible you can now try to get ConnectionInfo on the FormExampleDatabase with something like the following:

```
> FormDatabaseTester -l user -p password FormExampleDatabase
com.chimu.formTools.test.ConnectionInfo
```

If this works, the rest is up to the FORM DatabaseSetup scripts.

## 5.3 Setup of a FormExampleDatabase

The FORM examples are divided into subdirectories based on the different example schemes used. In each scheme directory there are multiple classes in a package with the same name as the directory ('scheme1', 'scheme1b', etc.). Among these class are two database oriented '.class' files, one named 'SetupExampleDatabase' and one named 'DropExampleDatabase'. To setup an example database, change to the examples directory (e.g. '../chimu/examples') and use the FormDatabaseTester to execute the SetupExampleDatabase class on the FormExampleDatabase:

```
> FormDatabaseTester -l user -p password FormExampleDatabase
scheme1.SetupExampleDatabase
```

The database should now have populated tables suitable for the particular example. For Scheme1 this will be a single 'Person' table with about 100 rows.

When you are done with the database or need to reuse it for a second scheme you can drop all the Example tables from it with:

```
> FormDatabaseTester -l user -p password FormExampleDatabase
scheme1.DropExampleDatabase
```

## 5.4 Running the Examples

After the FormExampleDatabase is setup you can run the examples. The examples can be run with the same DatabaseTester and database as you used for the database setup:

```
> FormDatabaseTester -l user -p password FormExampleDatabase
scheme1.Ex_PersonRetrieval_1
```

Which should answer something like:

```
Test: ---- Ex_PersonRetrieval_1 ----
<Person#99 Regina Cramer>
Regina Cramer is 64 inches tall and can be contacted at cramere@org.com
```

The actual data may be different because the Person selected is arbitrary.

## Recompiling the examples

The distribution of examples includes source so they can be modified and recompiled with:

```
> javac -d . scheme1/*.java
```

The '-d' is required to put the examples into the appropriate package directory. If you encounter any compile errors, make sure both the 'Form.jar' and 'FormTools.jar' files are in the classpath.



## ***5.5 You are Done***

If you get an output similar to the above, you are done. The examples work and FORM is working. You can now test out more examples, read the FORM manuals (see the next chapter), look through the example source, and create your own test examples.

If you have any problems with the above, try to make sure each step is working before going to the next. For example, you can use JDBC to talk to the database, you have a FormExampleDatabase with sufficient privileges to create tables, and you can use the FormDatabaseTester to talk to the FormExampleDatabase. If you continue to have difficulties, please contact us.

## ***5.6 Other Topics***

### **Using the FormDatabaseTester**

Multiple examples can be executed together and the FormDatabaseTester has options that handle tracing ('-t') and other display aspects. See the FORM Tools document for more information.

### **Running Examples from you own program**

The FORM examples provide a very simple protocol that can be called directly instead of by the DatabaseTester. They respond to 'run(Connection jdbcConnection)', so if you have a program that creates its own jdbcConnection you can use it to run the examples also. See the FORM Tools manual for more information on the testing/run interface.



## 6 Where to go next

---

After installing FORM and getting familiar with the distribution, the next step is to dig into the examples and documentation.

- *Introduction to FORM* provides an overview of FORM capabilities, concepts, and terms.
- *Learning FORM* is the main detailed documentation of FORM.
- The Examples show code that works with FORM.
- The *FORM API* Javadoc gives a complete description of the FORM interface.
- The *FORM Tools* manual describes the command line and graphical FORM tools in detail.

### 6.1 ChiMu Corporation References

**Architecture** *A Good Architecture for Object-Relational Mapping*

**FORM Tools** *FORM Tools*

**Foundations** *Foundations of Object-Relational Mapping*

**Introduction** *Introduction to FORM*

**Kernel** *Kernel Frameworks*

**Learning** *Learning FORM*

**Standards** *Java Development Standards*

**Starting** *Getting started with FORM*





**ChiMu Corporation**

1220 N. Fair Oaks Ave, #1314  
Sunnyvale, CA 94089

Phone: 408 734-9068

Email: [info@chimu.com](mailto:info@chimu.com)

[www.chimu.com](http://www.chimu.com)