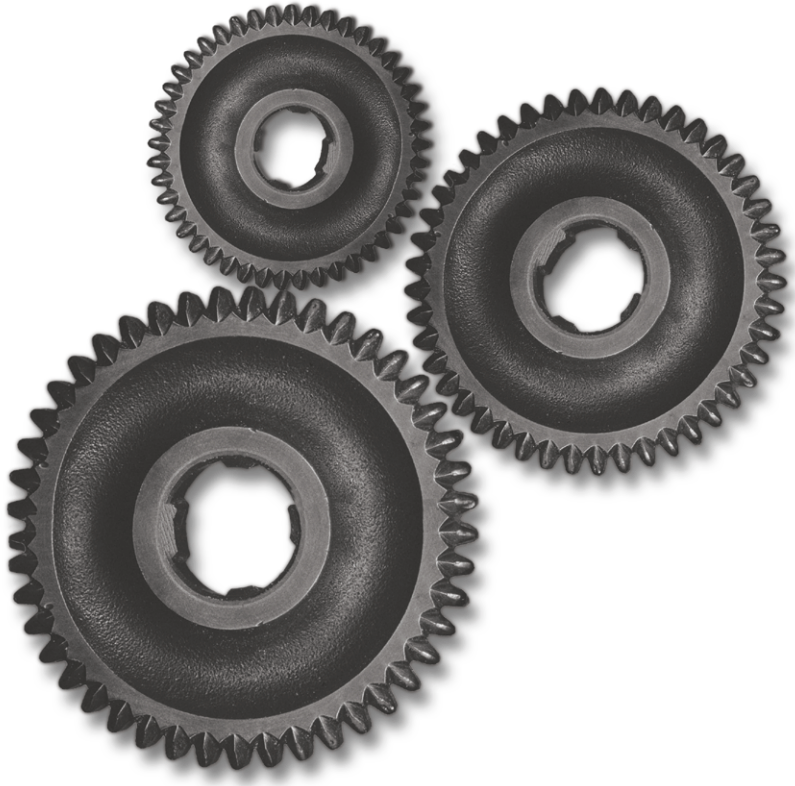
Getting Started Guide



Formula One™

Analytical spreadsheet reporting tool for distributed computing environments

Version 7.0

Tidestone Technologies, Inc.

Published by
Tidestone Technologies, Inc.
12980 Metcalf Avenue, Suite 300
Overland Park, Kansas 66213
phone 913-851-2200
toll-free 1-800-884-8665
fax 913-851-1390

www.tidestone.com
www.f1j.com

Printed in the United States of America.

99/701.11

Contents

Tidestone

First Things First

This section gives a broad overview of Formula One's two components, explains how to install Formula One, introduces the Workbook Designer and the Formula One JavaBean, and shows where to go to find product documentation.

Formula One's AWT and Swing Components

Formula One provides two components: the AWT component and the Swing component. The classes in these components are optimized for Java's AWT and Swing packages, respectively.

Developers who are going to display any of Formula One's GUI elements in their products need to choose between Java's AWT and Swing packages before they begin working with Formula One. This decision may be based on the JDK version they are deploying with (AWT appears in JDK 1.1.5 and later, Swing in 1.1.8 and later) and other factors.

Once developers choose either AWT or Swing, they'll need the following information to work with Formula One.

- **JAR files.** To deploy a project using Formula One's Swing component, developers need to include `F1J7Swing.jar`. To deploy using the AWT component, developers need to include either `F1J7AWT.jar` or `F1J7AWTDesign.jar`. (To deploy Formula One on a database or application server, specialized JAR files can be created from these standard JAR files. See the Technical Guide for more information.)

  For more information, see "Formula One JAR files" on page 5.

- **Main class.** The main class you use to display Formula One's Swing component is `com.f1j.swing.JBook`. The main class in Formula One's AWT component is `com.f1j.View`.

  For more information, see "Formula One AWT Primary Classes" on page 15 and "Formula One Swing Primary Classes" on page 25.

- **Workbook Designer differences.** The Workbook Designer (Formula One's user interface for displaying and editing spreadsheets) looks and acts differently depending on whether it's deployed as an AWT or Swing component. The Swing Designer offers more features than the AWT Designer.

  For more information, see "About the Workbook Designer" on page 8.

- **Upgrading.** Upgrading from Formula One 5.5 to the 7.0 AWT component is straightforward, but upgrading to the 7.0 Swing component is more involved.

  For more information, see "Upgrading to Formula One 7.0" on page 32.

Quick Reference: Formula One's AWT and Swing Components

This table provides at-a-glance information on the differences between Formula One's AWT and Swing components.

	Formula One's AWT component	Formula One's Swing component
JAR files	`F1J7AWT.jar` and `F1J7AWTDesign.jar`	`F1J7Swing.jar`
Main class	`com.f1j.View` (extends `java.awt.component`)	`com.f1j.swing.JBook` (extends `javax.swing.JComponent`)
JDK version	1.1.5 and later	1.1.8 and later
Workbook Designer	No online help, charting, or undo/redo	Full features

Installing Formula One

Obtaining Formula One

You can obtain Formula One by purchasing the CD software and documentation package or by downloading the software over the Internet from the Tidestone Technologies website at www.tidestone.com. You can also find licensing, pricing, technical support, and other information about Formula One at this site.

Platforms Tested

Formula One has been extensively tested and is known to run reliably on Windows 95, Windows 98, Windows NT 4.0, Solaris 2.6, and Macintosh OS 8.5. Other platforms that support JDK 1.1.5 or better should be capable of running Formula One.

System Requirements

Depending on your system, Formula One 7.0 requires about 25 MB of space on the computer's hard drive.

Because it is Java software, Formula One also requires that a Java Virtual Machine (JVM) be installed on the host computer.

Formula One JVM Compatibility

Formula One supports a range of version releases of the JVM, starting with version 1.1.5. Utilizing a version of the JVM that supports Swing will give you access to Formula One's newest functionality, which resides in its Swing component.

This table shows Formula One's support for different versions of the JVM.

JVMs	Formula One's AWT component	Formula One's Swing component
JVM 1.1.5 and later	Yes	No
JVM 1.1.8 and later	Yes	Yes

Formula One Browser Compatibility

Like its JVM compatibility, Formula One's browser compatibility depends in large part on which Formula One component you are using.

Applets created with Formula One's AWT component will run in browsers that support JVMs version 1.1.5 and later. This includes most of the popular browsers on Windows, Solaris, and Macintosh.

Applets created with Formula One's Swing component will run in browsers that support JVMs version 1.1.8 and later. Some browsers on some platforms do not support these more advanced JVMs. Moreover, applet developers who want to access all the functionality of Formula One's Swing component must use the Java Plug-in to deploy their applets. For more information on the Java Plug-in, see "Running the Swing sample code as an applet" on page 28.

Note These are the general guidelines for Formula One's browser compatibility. Various isolated problems in different vendors' software undermine Formula One's browser compatibility on certain versions of certain platforms. Please see the ReadMe for more detailed information.

Installing Formula One

Formula One provides three installation options:

- a Windows self-extracting executable program.
- a class file for installing on all other platforms. (See special instructions for using this class file on Macintosh.)
- a shell script for UNIX customers who install Formula One from the CD. (Due to technical problems, this shell script is not available in the Internet download.)

The Formula One installer requires that a Java platform be located on your path. While the installer will run on JVMs back to the 1.1.5 version, we recommend using Java 2™. For information on how to set up your path, please visit www.tidestone.com and follow the links to the Support section.

The installer lets you identify the components you want to install, select a directory to hold the program files, and choose a folder in which to place the program on your desktop. It then copies the files to your hard disk.

Note Previous versions of the Formula One installer installed documentation files in PDF format along with the software. Tidestone has removed the PDFs from the 7.0 installer to make the download smaller. You may download these PDFs free from the documentation page on the Tidestone website at www.tidestone.com. They are also available on the CD.

➤ **To install Formula One in Windows 95/98/NT:**

1. Locate the file `F1J7Setup.exe` on the CD or from the files you downloaded and double-click on the file to run the installer.
2. Read and follow the instructions that appear in the installation program windows.

The Windows installer creates icons for the Workbook Designer, ReadMe, and Uninstaller in the Start menu.

➤ **To install Formula One on all other platforms:**

1. Open a command prompt or file manager.
2. Change to the directory that contains the file `setup.class` on the root of the CD or from the files you downloaded.
3. Execute the following command to start the installation:

```
java -cp . setup
```

4. Follow the instructions in the installation wizard to complete the installation process.

➤ **To install Formula One from the CD on UNIX machines:**

1. Open a command prompt or file manager.
2. Change to the root directory of the CD-ROM.
3. Execute the following command to start the installation:

```
setup.sh
```

4. Follow the instructions in the installation wizard to complete the installation process.

➤ **To install Formula One on Macintosh OS 8:**

You need to use the JBindery utility on Macintosh to run the Formula One installer. JBindery is provided with the Mac OS Runtime for Java (MRJ) SDK. The SDK can be downloaded from the Apple website at devworld.apple.com/java/.

1. Launch JBindery.
2. In the Command panel of the JBindery dialog, in the Class Name field, enter the following command:

   ```
   setup
   ```

3. Click on the Classpath icon to display the Classpath panel, then click Add .zip file.
4. In the navigation dialog, locate the `setup.class` file on your system. Select it and click Open.
5. Click Run to start the installation.
6. Follow the instructions in the installation wizard to complete the installation process.

Formula One Files

The CD package and the Internet download packet contain the following files, arranged by category. The Formula One installer copies these files to your system.

For information on which of these files are required to deploy applications and applets using Formula One, see “Required Files for Formula One AWT Deployment” on page 14 and “Required Files for Formula One Swing Deployment” on page 23.

Formula One JAR files

Formula One’s AWT and Swing components are placed into separate JAR files so that developers can ship the smallest JARs possible with their applications. The names of the JAR files indicate which Java GUI class those files can be used with; `F1J7Swing.jar` *does not include Java’s Swing classes, just Formula One’s Swing-specific classes.*

`F1J7Swing.jar`	The Swing version of the Formula One JavaBean, applet, and application.
`F1J7AWTDesign.jar`	The AWT version of the Formula One JavaBean, applet, and application.
`F1J7AWT.jar`	The AWT version of the Formula One JavaBean and applet (but not the application).

Java standard extension JAR files

These JAR files are Java extension packages created by Sun Microsystems. Because they are not part of the core Java platform, developers who want to access their functionality must ship them along with their product.

`infobus.jar`	The Java classes for the InfoBus Java extension. Some of these classes may be called from the Formula One API.
`jh.jar`	The Java classes that run the JavaHelp system. You must use this JAR file if you want to view the online help through the Swing Workbook Designer. The help content files are stored in `F1Help.jar` (see below).

Examples and demos

These directories contain examples and demos. For specific information about each demo, see "Formula One AWT Demos" on page 20 and "Formula One Swing Demos" on page 30.

examples/swing	Several sets of example code showing how to use Formula One's Swing component in applications and applets.
examples/awt	Several sets of example code showing how to use Formula One's AWT component in applications and applets.
servlets/	Servlet example code (both the Java and resulting class files) and a readme.

Documentation

These files and directories contain the various types of documentation in the software download. For more information on documentation, see "Getting Help" on page 12.

`F1Help.jar`	Formula One user documentation help set in the JavaHelp format. Includes the contents of the User's Guide and Function Reference in HTML and XML.
help/	Directory that includes PDF files of this Getting Started guide as well as the User's Guide, Function Reference, and Technical Guide.
help/awtdoc	Directory containing Javadoc-generated API documentation for Formula One's AWT component in HTML format.
help/swingdoc	Directory containing Javadoc-generated API documentation for Formula One's Swing component in HTML format.
`readme.html`	Basic information on installation and known bugs and limitations of the product.

`excel_compatibility.html`	A list of features that have some degree of incompatibility with Excel.
`chart_types.html`	A list of the Excel 3D chart types and the type of chart they will be converted to in Formula One.
`function_readme.html`	A list of issues involving worksheet function compatibility with Excel.
`F1JLicen.html`	The license agreement for Formula One. “Licensing Formula One” on page 36 is a copy of this document.
`features.html`	A list of the new and enhanced features of Formula One’s 7.0 version.

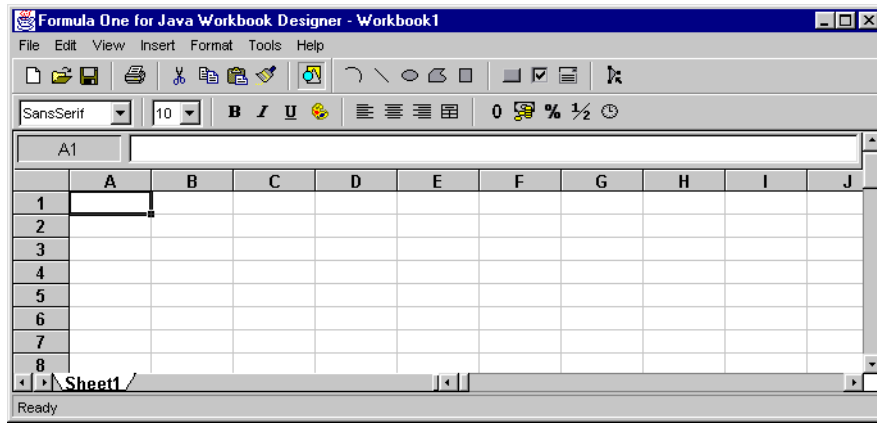
Utilities

These files help with the installation, deployment, and uninstallation of Formula One.

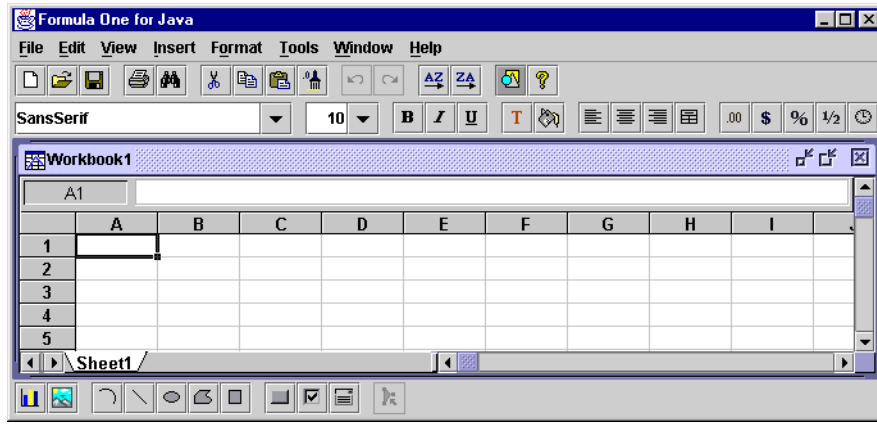
`F1JSplit.class`	A class file that optimizes applet deployment by separating the classes in `F1J7Swing.jar` into eight groups, then putting each group into its own JAR file.
`F1J7.exe`	(installed on Windows only) A Windows executable that launches the Formula One application.
`Uninst.isu`	A Windows-specific log file the Windows uninstaller uses to remove the software from the computer.

About the Workbook Designer

The Workbook Designer is Formula One's spreadsheet data entry and formatting tool. There are actually two Workbook Designers: one that corresponds to Formula One's AWT component, and one for the Swing component. The two designers are shown below.



The AWT Workbook Designer



The Swing Workbook Designer

Differences Between the Two Workbook Designers

The Swing version of the Workbook Designer offers the following features that the AWT designer does not.

- Charting
- Undo and redo menu options and the ability to change the number of undos

- Access to Designer Help, online user help for the Workbook Designer and for the worksheet functions
- A changeable look and feel
- Ability to zoom in or out when viewing a worksheet
- A Window menu that allows users to cascade and tile multiple windows and make different windows active
- More print options (print orientation, paper size, and restart page numbering)
- The ability to apply fill patterns to cells
- Lotus-style formula evaluation
- Toolbar buttons for find/replace, undo, redo, sorting, online help, and inserting picture objects

The AWT Workbook Designer displays a status bar. The Swing Designer doesn't.

Using the Workbook Designers in Applications and Applets

You have the option of providing Workbook Designer functionality automatically on any application or applet that displays a Formula One worksheet.

➤ **To launch the Workbook Designer from a Formula One application or applet:**

- Select a worksheet cell and press F3. The Workbook Designer will appear in a new window.

To provide access to the Workbook Designer, ship the appropriate Formula One JAR file: `F1J7Swing.jar` for the Swing Workbook Designer, `F1J7AWTDesign.jar` for the AWT designer. This JAR file must be on the application's classpath for the software to work.

You can also choose to restrict users from accessing the Designer.

To restrict Workbook Designer access, ship a different JAR file. For AWT developers, ship `F1J7AWT.jar`. Swing developers may use the `F1JSplit.class` file to split up the classes in `F1J7Swing.jar` into 12 separate JAR files. The resulting JAR file containing Workbook Designer functionality is `F1J7Design.jar`, so to restrict Workbook Designer access, don't ship `F1J7Design.jar`. You may choose among the remaining JAR files to provide the functionality your application needs. For more information on splitting `F1J7Swing.jar` and how to use the resulting JAR files, see the Technical Guide.

You may also restrict access to the Workbook Designer through the API.

Launching the Standalone Workbook Designers

You can launch the AWT and Swing Workbook Designers as standalone applications.

Classpath settings. Before launching the Swing Workbook Designer, you may have to change your computer's classpath settings to include the Formula One JAR file `F1J7Swing.jar`. In addition, if you want the Designer Help (including User Help and the Function Reference) to be available from the Help menu in the Workbook Designer, you must also put the JAR files `jh.jar` and `F1Help.jar` on the classpath.

➤ **To launch the Swing Workbook Designer on Windows 95/98/NT:**

- Locate and double-click on the `F1J7.exe` file.

  or

- Click the Start button and choose the Formula One menu item from the Programs menu list.

➤ **To launch the Swing Workbook Designer on Solaris:**

A shell script, included with the installer, launches the Workbook Designer.

1. At the command prompt, switch to the installation directory.
2. Execute the following command:

```
./f1j7
```

➤ **To launch the Swing Workbook Designer on any platform:**

1. Open a command prompt and switch to the directory that contains `F1J7Swing.jar`.
2. Execute the following command:

```
java -cp F1J7Swing.jar com.f1j.swing.designer.Designer
```

   or

   Execute the following command to include the Designer Help documentation:

```
java -cp F1J7Swing.jar;jh.jar;F1help.jar
        com.f1j.swing.designer.Designer
```

Note On Solaris and other UNIX platforms, replace the semicolons (;) in this command with colons (:).

➤ **To launch the AWT Workbook Designer on the Macintosh:**

The Swing Designer does not run reliably on the Macintosh JVM, but Macintosh users can launch the AWT Workbook Designer from the desktop by using the JBindery utility to create a launchable application that packages the Formula One

classes. JBindery is provided with the Mac OS Runtime for Java (MRJ) SDK. The SDK can be downloaded from the Apple website at devworld.apple.com/java/.

1. Launch JBindery.
2. In the Command panel of the JBindery dialog, enter the following string in the Class Name field.

```
com.f1j.designer.Designer
```

3. Click on the Classpath icon to display the Classpath panel.
4. Click Add .zip file.
5. In the navigation dialog, locate the `F1J7AWTDesign.jar` file on your system. Select it and click Open.
6. In the JBindery dialog, click Save Settings.
7. In the Save Settings dialog, enter the name you want to give to your launchable application. Then click Desktop to save the application to the desktop.
8. Click Save to save the launchable application.
9. In the JBindery dialog, click Run.

A clickable icon for the Workbook Designer appears on the desktop. You can leave the icon on the desktop or place it in any folder on your system.

➤ **To launch the AWT Workbook Designer on any platform:**

1. Open a command prompt and switch to the directory that contains `F1J7AWTDesign.jar`.
2. Execute the following command:

```
java -cp F1J7AWTDesign.jar com.f1j.designer.Designer
```

Using Formula One as a JavaBean

Because Formula One is a JavaBean, you can use it in an IDE (integrated development environment) to automatically generate much of your Java code. The Formula One JavaBean can be used in Java IDEs such as Symantec's Visual Cafe, Borland's JBuilder, Microsoft's Visual J++, and Oracle's JDeveloper.

When you add Formula One to a form in an IDE, the IDE generates much of the code required to use the component. If you want to customize the appearance or behavior of a workbook, you will need to access the Formula One API directly rather than depending on the IDE to do it for you automatically.

The process of accessing Formula One's JavaBean functionality in an application varies slightly from one IDE to another. In most cases it consists of:

- adding the JavaBean to your project
- selecting the component and drawing the control on a frame or in a window

For detailed instructions on adding the JavaBean to your IDE, see the file `f1bean.html`, located in the download, on the CD, or on Tidestone's website (www.tidestone.com).

Note When Formula One is used as a JavaBean, only a small set of its API is available through the IDE's windows and menu options. Most developers will find that they have to access other parts of the Formula One API within the Java code to take advantage of Formula One's many features.

Getting Help

Formula One offers many choices for documentation. End-users, developers, and software reviewers will all find printed, online, and web-based options. This table lists the Formula One help sources and where to find them.

Help Source	Audience	Contents	Format(s)	Location and file name
User's Guide	Users	How to use the Workbook Designer: entering and editing data and formulas, formatting, graphical objects, charts, printing.	JavaHelp	Press the Help button in the Workbook Designer.
			PDF	On Tidestone's website at www.tidestone.com.
			printed	Comes with the CD or can be ordered from Tidestone.
Function Reference	Users	Alphabetical listing of the 325 worksheet functions detailing proper usage. Shows syntax, examples, and related functions.	JavaHelp	Press the Help button in the Workbook Designer
			PDF	On Tidestone's website at www.tidestone.com.
			printed	Comes with the CD or can be ordered from Tidestone.

Help Source	Audience	Contents	Format(s)	Location and file name
Technical Guide	Developers	Deploying Formula One in applets and applications, Java security issues, data access using JDBC and InfoBus, attach methods, creating add-in functions, performance tuning, and specifications.	PDF	On Tidestone's website at www.tidestone.com.
			printed	Comes with the CD or can be ordered from Tidestone.
API Help	Developers	Two Javadoc-generated guides to the public classes: one for the AWT component and one for the Swing component. Displays tree structure and describes how classes and methods should be implemented.	HTML	[installdir]\help\awtdoc\`index.html` [installdir]\help\swingdoc\`index.html`
ReadMes	Users and developers	Covers installation, known problems, and how to contact Tidestone. Linked files detail compatibility issues with Microsoft Excel.	HTML	[installdir]`chart_types.html` [installdir]`excel_compatibility.html` [installdir]`f1bean.html` [installdir]`features.html` [installdir]`function_readme.html` [installdir]`readme.html`
Demos	Developers	Applications and applets written in Java that demonstrate various pieces of Formula One's functionality. For descriptions of each demo, see "Formula One AWT Demos" on page 20 and "Formula One Swing Demos" on page 30.	Java applets and applications	On Tidestone's website at www.tidestone.com. Follow the Demo links to the Java demos. [installdir]\examples and [installdir]servlets
Knowledge-Base	Developers	A database of frequently asked technical questions from the Tidestone developer services database. Solutions, explanations, workarounds, code samples.	Database	On the Tidestone website at www.tidestone.com. Follow the Support links to the Online KnowledgeBase.

Using Formula One with AWT

This section contains information specific to developers who want to use Formula One's AWT component to deploy Java software using JDK 1.1.5 or later.

Required Files for Formula One AWT Deployment

To deploy Formula One in an AWT-based application or applet, use either of the following files. For a list of the packages in each JAR file, see "Formula One AWT Packages" on page 14.

File name	Description
`F1J7AWT.jar`	Use this file if you are deploying just the View class, not the Workbook Designer.
`F1J7AWTDesign.jar`	Use this file if you want users to be able to access the Workbook Designer from within your application or applet.

Note Other deployment and documentation files installed with the product (e.g. `F1J7Swing.jar`, `F1Help.jar`, etc.) can't be used reliably with JDKs prior to 1.1.8; therefore, they're not mentioned here.

The Formula One AWT API

Formula One AWT Packages

Formula One's AWT component is divided into the following packages, all of which are public API and available in `F1J7AWTDesign.jar`. (Packages not available in `F1J7AWT.jar` are noted in the table.)

Package Name	Description	Primary Classes

`com.f1j`	This package contains the classes that make up the bulk of the API. Most of the classes used for basic spreadsheet tasks, like data population, cell formatting, graphical objects, JDBC, etc., are found in this package. See the tables below for more information on these basic classes.	`View`
		`HTMLWriter`
		`JDBC`
		`CellFormat`
		`F1Exception`

`com.f1j.addin`	This package contains the abstract class Func, which can be extended to create custom worksheet functions that can be accessed from the spreadsheet.	`Func`

Package Name	Description	Primary Classes
com.f1j.calc	This package contains the Database class, which enables data extraction from one part of a spreadsheet to another.	Database
com.f1j.designer	This package's Designer class displays the Workbook Designer, an AWT-based application that provides Excel-like menus, toolbars, and dialogs for spreadsheet data entry and formatting. This package is not in F1J7AWT.jar.	Designer
com.f1j.dialog	This package contains classes to display AWT-based dialog boxes to collect information from the user. This package is not in F1J7AWT.jar.	FindReplaceDlg, FormatCellsDlg, SaveFileDlg, etc.
com.f1j.ss	This package contains classes to access the spreadsheet model.	Book

Formula One AWT Primary Classes

These are the main classes used to create and deploy Formula One worksheets in an AWT-based application or applet.

Class name	Description
com.f1j.View	AWT-based spreadsheet component. This class can be instantiated from an applet, servlet, or application, or can be used in an IDE form-builder environment. This class provides methods for manipulating the spreadsheet contents.
com.f1j.designer.Designer	AWT-based Workbook Designer. This Java application provides menus, toolbars, and dialogs on top of the com.f1j.View spreadsheet component to help the user create, modify, and format spreadsheet files.

Formula One AWT Utility Classes

These classes can be used to write out HTML files from spreadsheets and to enable JDBC database connectivity.

Class name	Description
`com.f1j.HTMLWriter`	This class provides static methods to convert a spreadsheet to an HTML table.
`com.f1j.JDBC` `com.f1j.JDBCQueryObj`	These two classes provide methods to extract data from a JDBC ResultSet and populate a spreadsheet with values and formatting.

Formula One AWT Classes Returned from View

The `View` class returns these classes.

Class name	Description
`com.f1j.ss.Book`	This is the model portion of the `View` class. Developers can use this class to bypass the `View` methods when building the spreadsheet model.
`com.f1j.CellFormat`	This class is used to set/get formatting information in/from the spreadsheet cells.
`com.f1j.GRObject` `com.f1j.GRObjectPos`	These two classes represent the graphical objects on the spreadsheet.
`com.f1j.FindReplaceInfo`	This class contains information regarding the last find/replace operation called.
`com.f1j.NumberFormat`	This class represents the custom formats stored in the spreadsheet.
`com.f1j.RangeRef`	This class represents a range of cells.
`com.f1j.CellRef`	This class represents a cell.

Formula One AWT Exception

The one exception in Formula One's AWT API is `com.f1j.F1Exception`. It can be thrown from several method calls throughout the API.

Formula One AWT Events and Listeners

Events and Listeners	Description
`com.f1j.CancelEditEvent` `com.f1j.CancelEditListener`	This event is fired when the user exits in-cell editing mode by pressing the Esc key.
`com.f1j.EndEditEvent` `com.f1j.EndEditListener`	This event is fired when the user exits in-cell editing mode by normal means (pressing Enter, using the arrow keys or mouse to move to another cell, etc.).
`com.f1j.EndRecalcEvent` `com.f1j.EndRecalcListener`	This event is fired after the spreadsheet has completed its recalculation cycle.
`com.f1j.ModifiedEvent` `com.f1j.ModifiedListener`	This event is fired when the spreadsheet's data or formatting changes.
`com.f1j.ObjectEvent` `com.f1j.ObjectListener`	This event is fired when the user interacts with a graphical object.
`com.f1j.SelectionChangedEvent` `com.f1j.SelectionChangedListener`	This event is fired each time the cell selection changes.
`com.f1j.StartEditEvent` `com.f1j.StartEditListener`	This event is fired when the user enters in-cell editing mode.
`com.f1j.StartRecalcEvent` `com.f1j.StartRecalcListener`	This event is fired when the recalculation cycle is about to begin.
`com.f1j.UpdateEvent` `com.f1j.UpdateListener`	This event is fired when the component has repainted the spreadsheet.
`com.f1j.ValidationFailedEvent` `com.f1j.ValidationFailedListener`	This event is fired when a validation rule fails.
`com.f1j.ViewChangedEvent` `com.f1j.ViewChangedListener`	This event is fired when the visible part of the spreadsheet changes (e.g. scrolling, tab selection, window resize, etc.)

Sample Formula One AWT Application and Applet

This sample Java code displays a spreadsheet with the formula bar, row and column headings, scroll bars, and formatted text in one of the spreadsheet cells. It demonstrates the very basic code necessary to create and display a spreadsheet. The file containing this code, `WelcomeA.java`, can be found in the examples/awt/welcome directory.

This Java program can be compiled and executed as an application, or it can be used as an applet embedded in an HTML file. The HTML code necessary to embed the applet is shown after the Java code.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import com.f1j.View;

public class WelcomeA extends Applet  {

    static View view1 = new View();

    public WelcomeA() {
    }

    //Initialize the applet
    public void init() {

        setLayout(null);
        setSize(400,300);
        view1.setBounds(24,24,375,182);
        add(view1);

        try {
            // Call the Formula One setText API to place text into a
             cell
            view1.setText(2, 0, "Welcome to Formula One for Java: "+
                                "The Internet Spreadsheet");

            // Add a format to the cell
            com.f1j.CellFormat CellFormat1;
            CellFormat1 = view1.getCellFormat();
            CellFormat1.setHorizontalAlignment(
                   CellFormat1.eHorizontalAlignmentCenterAcrossCells);
            CellFormat1.setFontBold(true);
            CellFormat1.setFontItalic(true);
            CellFormat1.setFontSize(210);
            view1.setCellFormat(CellFormat1,2,0,2,6);

        }
        catch (com.f1j.F1Exception e) {
            // catch any exceptions in case something fails setting the
            // text
            System.out.println(e.getMessage());
        }
    }

    public static void main(String args[]){
        // Create a Frame to place our application in.
        Frame myFrame = new Frame ("Getting Started");

        myFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent event) {
                System.exit(0);
            }
```

```
        });

        // Create an instance of the Welcome class
        WelcomeA myApp = new WelcomeA();
        myFrame.add ("Center", myApp);
        myApp.init();

        // Resize the Frame to the desired size, and make it visible
        myFrame.setSize(600,400);
        myFrame.show();

        // Run the methods the browser normally would
        myApp.start();
    }
}
```

Running the AWT sample code as an application

First compile it into a class file using your IDE compiler or the Java command-line compiler, `javac`. If `WelcomeA.java` is in the same directory as `F1J7AWTDesign.jar`, open the command prompt, change to that directory, then type in the following Java command:

```
javac -classpath F1J7AWTDesign.jar WelcomeA.java
```

Then you can execute the resulting class within your IDE or from a command prompt using the Java command `java`. If `WelcomeA.class` is in the same directory as `F1J7AWTDesign.jar`, open the command prompt, change to that directory, then type in the following Java command:

```
java -cp .;F1J7AWTDesign.jar WelcomeA
```

Running the AWT sample code as an applet

First, compile it into a class file using the instructions above for compiling the application code.

Then, create an HTML file in which to display the class as an applet. Following is sample code for a very simple HTML file to display the applet. Opening this HTML file in a browser will display the spreadsheet.

```
<HTML>
<HEAD>
<TITLE>Live Spreadsheet Page</TITLE>
</HEAD>
<BODY>
<APPLET CODE="WelcomeA.class" ARCHIVE="F1J7AWT.jar" WIDTH=400
        HEIGHT=300></APPLET>
</BODY>
</HTML>
```

This HTML will only work if `WelcomeA.class` and `F1J7AWT.jar` are in the same directory. The `<APPLET>` tag's `ARCHIVE` parameter references Formula One's `F1J7AWT.jar` file. You may choose to reference the `F1J7AWTDesign.jar` file if you want applet users to be able to access the Workbook Designer. However, since `F1J7AWTDesign.jar` is a much larger JAR file, the browser will take longer to download it. For more information, see "Using the Workbook Designers in Applications and Applets" on page 9.

Formula One AWT Demos

Formula One comes with several demos that developers can use to learn about the Formula One API. There are two sets of Formula One AWT example code: basic demos and advanced demos.

The Basic AWT Demos

These demos show basic Formula One coding features. Developers can use snippets of code from these demos to plug that functionality into their own applications.

Each of the basic demos is located in a separate directory in the [installdirectory]\examples\awt\com\f1j\demo directory. Each consists of one Java file that extends the `F1Applet` class, also in that directory. They are all members of the `com.f1j.demo` package.

Before running the demos, you must first compile them into class files using your IDE compiler or using the Java command-line compiler, `javac`. Then you can execute the resulting class within your IDE or from a command prompt using the Java command `java`.

The table below provides a brief description of each basic demo and shows the key methods used in each demo.

Demo name and description	Classes and methods demonstrated	

`attach.java`
Attaching two views to the same workbook.
	`View`
	`setTopRow`
	`setLeftCol`
		`setLayout`
		`setActiveView`

`autofilllist.java` Creating an autofill list.	`setAutoFillItems`	

`cellformat.java`
Modifying cell formats, including text styles, fonts, and borders.
	`getCellFormat`
	`setCellFormat`
	`useAllFormats`
	`setText`
	`setFontBold`
	`setFontItalic`
		`setFontColor`
		`setLeftBorder`
		`setRightBorder`
		`setTopBorder`
		`setBottomBorder`
		`setColWidthAuto`

Demo name and description	Classes and methods demonstrated
definedname.java Creating a defined name to refer to a cell or cells.	setDefinedName setFormula setText setNumber
externref.java Creating an external cell reference to refer to a cell or cells in another workbook.	setWorkbookName setActiveView setGroup setLayout
f1applet.java Creating an applet. All the other Workbook Designer demos instantiate this class.	View DesignerPanel getView
f1frame.java A utility class that allows these applets to run as applications.	View FormatCellsDlg DesignerPanel GRObject PopupWindow F1Exception PageSetupDlg OpenFileDlg PasteSpecialDlg SaveFileDlg
freezepanes.java Freezing rows and columns so that they don't scroll.	setFixedCol setFixedRows setFixedCols setFixedRow
getlockoptimization.java Optimizing recalculation time using the getLock method.	getLock
headings.java Changing row and column heading text.	setColText setRowText
lockcells.java Locking a cell so that it can't be edited or formatted.	setLocked setEnableProtection
populateoptimization.java Populating a spreadsheet to show that loading cells from the bottom up is faster than loading from the top down.	setNumber setText
setentryoptimization.java Populating a spreadsheet with the setNumber and setEntry methods.	setEntry setNumber
validation.java Creating a validation rule.	setValidationRule setValidationText

The Advanced AWT Demos

These demos are more complicated than the basic demos, involving more Java files and other supporting files. Developers can use snippets of code from these demos to plug the functionality into their own applications.

Each of the advanced demos is located in a separate directory in the [installdirectory]\examples\awt directory.

Before running the demos, you must first compile them into class files using your IDE compiler or using the Java command-line compiler, `javac`. Then you can execute the resulting class within your IDE or from a command prompt using the Java command `java`.

The advanced demos are listed here by their directory names.

- **examples\awt\customdesigner.** This demo shows how to extend the `com.f1j.designer.Designer` class to create a custom version of the Workbook Designer. In this example, a custom menu item is added to the standard Workbook Designer menu set.
- **examples\awt\edi.** This demo shows Formula One reading in external data (EDI-formatted addresses stored in VTS, XLS, and TXT file formats) and merging that data with a template spreadsheet. Each of the files `EastRegion.*`, `WestRegion.*`, and `MidwestRegion.*` has address information for 9 customers in the first 9 rows of the file (first column for spreadsheet formats). As the user scrolls though each of the customers, the names are extracted from the data files, merged with the spreadsheet template (`Default.vts`), recalculated, and displayed.
- **examples\awt\format.** This demo shows how to use the formatting API (specifically, the `CellFormat` object) to set borders, text colors, background colors, etc. The application displays several buttons next to a Formula One worksheet. It formats the worksheet data differently based on the button pressed. This action is similar to the AutoFormat feature of Excel.
- **examples\awt\hr.** This demo is a human resources application designed to collect time sheet information stored in VTS files (Formula One's native file format). It illustrates how to programmatically create, delete, and display worksheets.
- **examples\awt\worksheet.** This applet-servlet demo shows how Formula One data can be transferred to and from a servlet. Clients log in with their own user names and passwords to ensure data security, then use a web browser to access, edit, and save spreadsheets on the server. To run this demo, install WorksheetServlet on your web server servlet's directory and WorksheetApplet and supporting files in the public directories. See the demo's readme for specific information about its classes and how to set them up on a server.

- **examples\awt\welcome.** The code in “Sample Formula One AWT Application and Applet” on page 17 of this guide.
- **examples\oracle\webtogo\vacation.** This servlet accesses an Oracle database to track employee vacation requests made both on and offline and route the requests to the appropriate manager for approval. Formula One is used to retrieve the data from the database via JDBC, place the data into a worksheet, summarize it using worksheet functions, then write this summary information out in response to the browser request. Although this demo uses Formula One’s Swing component JAR file (`F1J7Swing.jar`), the code inside it has no GUI references, so you can replace the JAR file with one of the AWT JAR files and it should run equally well.

Using Formula One with Swing

This section contains information specific to developers who want to use Formula One’s Swing component to deploy Java software using JDK 1.1.8 or later.

Required Files for Formula One Swing Deployment

To deploy Formula One in a Swing-based application or applet, use the following files. For more information on each of these files, see “Formula One Files” on page 5.

File name	Description
`F1J7Swing.jar`	This file contains all of Formula One’s Swing functionality, including access to the Workbook Designer. Developers who are concerned about download times may not want to deploy this large JAR file. For these cases, Formula One allows you to split `F1J7Swing.jar` into eight smaller JAR files and deploy whichever of those files suits your situation. For more information about this deployment option, see the Formula One Technical Guide.
`F1Help.jar` and `jh.jar`	(optional) Developers who provide access to the Workbook Designer and want to include the Designer Help online help system should deploy their applications using these two files as well.

The Formula One Swing API

Formula One Swing Packages

Formula One's Swing component is divided into the following packages, all of which are public API and available in `F1J7Swing.jar`.

Package Name	Description	Primary Classes
`com.f1j.addin`	This package contains the abstract class `Func`, which can be extended to create custom worksheet functions that can be accessed from the spreadsheet.	`Func`
`com.f1j.calc`	This package contains the `Database` class, which enables data extraction from one part of a spreadsheet to another.	`Database`
`com.f1j.infobus`	This applet class allows developers to share spreadsheet data within the context of a browser using the InfoBus technology. The `infobus.jar` file must be available for compiling code that references this package.	`InfoBusBook`
`com.f1j.jdbc`	Contains the JDBC class for database connectivity.	`JDBC`

`com.f1j.ss`	This package contains UI-generic classes to save spreadsheets as HTML, format cells, and access the spreadsheet model directly without using the `JBook` class.	`HTMLWriter`
		`CellFormat`
		`Book`
		`Sheet`

`com.f1j.swing`	This package contains the Swing components of Formula One.	`JBook` `JBookApplet`
`com.f1j.swing.designer`	This package's `Designer` class displays the Workbook Designer, a Swing-based application that provides Excel-like menus, toolbars, and dialogs for spreadsheet data entry and formatting.	`Designer`
`com.f1j.swing.ss`	This package contains classes to display Swing-based dialog boxes to collect information from the user.	`FormatCellsDlg`, `PageSetupDlg`, etc.
`com.f1j.util`	This package contains the Formula One exception class.	`F1Exception`

Formula One Swing Primary Classes

These are the main classes used to create and deploy Formula One worksheets in an Swing-based application or applet.

Class name	Description
com.f1j.swing.JBook	Swing-based spreadsheet component. This class can be instantiated from an applet, servlet, or application, or can be used in an IDE form-builder environment. This class provides methods for manipulating the spreadsheet contents.
com.f1j.swing.designer.Designer	Swing-based Workbook Designer. This Java application provides menus, toolbars, and dialogs on top of the com.f1j.swing.JBook spreadsheet component to help the user create, modify, and format spreadsheet files.

Formula One Swing Utility Classes

These classes can be used to write out HTML files from spreadsheets and to enable JDBC database connectivity.

Class name	Description
com.f1j.ss.HTMLWriter	This class provides static methods to convert a spreadsheet to an HTML table.
com.f1j.jdbc.JDBC com.f1j.jdbc.JDBCQueryObj	These two classes provide methods to extract data from a JDBC ResultSet and populate a spreadsheet with values and formatting.

Formula One Swing Classes Returned from JBook

The JBook class returns these classes.

Class name	Description
com.f1j.ss.Book	This is the model portion of the JBook class. Developers can use this class to bypass the JBook methods when building the spreadsheet model.
com.f1j.ss.CellFormat	This class is used to set/get formatting information from the spreadsheet cells.
com.f1j.ss.GRObject com.f1j.ss.GRObjectPos	These two classes represent the graphical objects on the spreadsheet.
com.f1j.ss.FindReplaceInfo	This class contains information regarding the last find/replace operation called.

Class name	Description
com.f1j.ss.NumberFormat	This class represents the custom formats stored in the spreadsheet.
com.f1j.ss.RangeRef	This class represents a range of cells.
com.f1j.ss.CellRef	This class represents a cell.

Formula One Swing Exception

The one exception in the Formula One Swing API is com.f1j.util.F1Exception. It can be thrown from several method calls throughout the API.

Formula One Swing Events and Listeners

Events and Listeners	Description
com.f1j.ss.CancelEditEvent com.f1j.ss.CancelEditListener	This event is fired when the user exits in-cell editing mode by pressing the Esc key.
com.f1j.ss.EndEditEvent com.f1j.ss.EndEditListener	This event is fired when the user exits in-cell editing mode by normal means (pressing Enter, using the arrow keys or mouse to move to another cell, etc.).
com.f1j.ss.EndRecalcEvent com.f1j.ss.EndRecalcListener	This event is fired after the spreadsheet has completed its recalculation cycle.
com.f1j.ss.ModifiedEvent com.f1j.ss.ModifiedListener	This event is fired when the spreadsheet's data or formatting changes.
com.f1j.ss.ObjectEvent com.f1j.ss.ObjectListener	This event is fired when the user interacts with a graphical object.
com.f1j.ss.SelectionChangedEvent com.f1j.ss.SelectionChangedListener	This event is fired each time the cell selection changes.
com.f1j.ss.StartEditEvent com.f1j.ss.StartEditListener	This event is fired when the user enters in-cell editing mode.
com.f1j.ss.StartRecalcEvent com.f1j.ss.StartRecalcListener	This event is fired when the recalculation cycle is about to begin.
com.f1j.ss.UpdateEvent com.f1j.ss.UpdateListener	This event is fired when the component has repainted the spreadsheet.
com.f1j.ss.ValidationFailedEvent com.f1j.ss.ValidationFailedListener	This event is fired when a validation rule fails.
com.f1j.ss.ViewChangedEvent com.f1j.ss.ViewChangedListener	This event is fired when the visible part of the spreadsheet changes (e.g. scrolling, tab selection, window resize, etc.)

Sample Formula One Swing Application and Applet

This sample Java code displays a spreadsheet with the formula bar, row and column headings, scroll bars, and some formatted text in one of the spreadsheet cells. It demonstrates the very basic code necessary to create and display a spreadsheet. The file containing this code, `Welcome.java`, can be found in the examples/swing/welcome directory.

This Java program can be compiled and executed as an application, or it can be used as an applet embedded in an HTML file. The HTML code necessary to embed the applet is shown after the Java code.

```
import java.awt.*;
import java.awt.event.*;
import com.f1j.swing.*;
import javax.swing.*;

public class Welcome extends JApplet {

    static JBook jBook1 = new JBook();
    public void Welcome1() {
    }

    //Initialize the applet
    public void init() {
        this.setSize(new Dimension(400,300));
        try {
            // Call the Formula One setText API to place text
            // into a cell
            jBook1.setText(2, 0, "Welcome to Formula One for Java: "
        +
                                 "The Internet Spreadsheet");

            // Add a format to the cell
            com.f1j.ss.CellFormat CellFormat1;
            CellFormat1 = jBook1.getCellFormat();
            CellFormat1.setHorizontalAlignment(
                     CellFormat1.eHorizontalAlignmentCenterAcrossCell
        s);
            CellFormat1.setFontBold(true);
            CellFormat1.setFontItalic(true);
            CellFormat1.setFontSize(210);
            jBook1.setCellFormat(CellFormat1, 2, 0, 2, 6);
        }
        catch (com.f1j.util.F1Exception e) {
            // catch any exceptions in case something fails setting t
        he
            // text
            System.out.println(e.getMessage());
        }
        this.getContentPane().add(jBook1, BorderLayout.CENTER);
```

```
    }

    public static void main(String args[]){
        // Create a Frame to place our application in.
        Frame myFrame = new Frame ("Getting Started");

        myFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent event) {
                System.exit(0);
            }
        });

        // Create an instance of the Welcome class
        Welcome myApp = new Welcome();
        myFrame.add ("Center", myApp);
        myApp.init();

        // Resize the Frame to the desired size, and make it visible
        myFrame.setSize(600,400);
        myFrame.show();

        // Run the methods the browser normally would
        myApp.start();
    }
}
```

Running the Swing sample code as an application

First compile it into a class file using your IDE compiler or the Java command-line compiler, `javac`. If `Welcome.java` is in the same directory as `F1J7Swing.jar`, open the command prompt, change to that directory, then type in the following Java command:

```
javac -classpath F1J7Swing.jar Welcome.java
```

Then you can execute the resulting class within your IDE or from a command prompt using the Java command `java`. If `Welcome.class` is in the same directory as `F1J7Swing.jar`, open the command prompt, change to that directory, then type in the following Java command:

```
java -cp .;F1J7Swing.jar Welcome
```

Running the Swing sample code as an applet

First, compile it into a class file using the instructions above for compiling the application code.

Then, create an HTML file in which to display the class as an applet. Because the popular browsers (Netscape Navigator and Internet Explorer) diverge in how they deal with applets that require recent versions of the JVM, you must create HTML that contains specialized tags for IE and different tags for Netscape. Sun Microsystems provides an HTML Converter utility that automatically converts

your <APPLET> tags to the appropriate IE and Netscape-specific tags. You can download this utility at http://java.sun.com/products/plugin/1.2/features.html.

Following is sample HTML code to display the applet using the <APPLET> tag. (This code assumes Welcome.class and F1J7Swing.jar are in the same directory.)

```
<HTML>
<HEAD>
<TITLE>Live Spreadsheet Page</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Welcome.class" ARCHIVE="F1J7Swing.jar" WIDTH=400
        HEIGHT=300></APPLET>
</BODY>
</HTML>
```

Following is that same HTML code after being run through Sun's HTML Converter.

```
<HTML>
<HEAD>
<TITLE>Live Spreadsheet Page</TITLE>
</HEAD>
<BODY>
<!--"CONVERTED_APPLET"-->
<!-- CONVERTER VERSION 1.0 -->
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
WIDTH = 400 HEIGHT = 300

codebase="http://java.sun.com/products/plugin/1.2/jinstall-12-
        win32.cab#Version=1,2,0,0">
<PARAM NAME = CODE VALUE = "Welcome.class" >
<PARAM NAME = ARCHIVE VALUE = "F1J7Swing.jar" >
<PARAM NAME="type" VALUE="application/x-java-applet;version=1.2">
<COMMENT>
<EMBED type="application/x-java-applet;version=1.2" java_CODE =
        "Welcome.class" java_ARCHIVE = "F1J7Swing.jar" WIDTH = 400
        HEIGHT = 300
pluginspage="http://java.sun.com/products/plugin/1.2/plugin-
        install.html"><NOEMBED></COMMENT>
</NOEMBED></EMBED>
</OBJECT>

<!--
<APPLET  CODE = "Welcome.class" ARCHIVE = "F1J7Swing.jar" WIDTH = 400
        HEIGHT = 300 >
</APPLET>
-->

<!--"END_CONVERTED_APPLET"-->
</BODY>
</HTML>
```

The <APPLET> tag has been replaced with code containing the <OBJECT> tag (used by Internet Explorer) and the <EMBED> tag (used by Netscape Navigator). Netscape ignores IE's <OBJECT> tag, while IE ignores anything within <COMMENT> tags.

Opening this converted HTML file in a browser will display the spreadsheet.

Note about the Java plug-in This HTML code contains the parameter `pluginspage`. This refers to the Java Plug-in, another Sun Microsystems utility that you must use if you want to run Swing applets in current browsers. The plug-in is necessary because Internet Explorer and Netscape Navigator do not contain the latest version of the JVM. When users whose computers don't already have the latest JVM encounter a web page containing an applet that uses Swing Java software, they must download and install the Java Plug-in software before they can run that applet. The plug-in automatically substitutes a more recent version of the Java Virtual Machine for the browsers' older versions. For information about downloading and using the Java Plug-in, see Sun Microsystems' web site at java.sun.com.

Formula One Swing Demos

Formula One comes with several demos that developers can use to learn about the Formula One API. Developers can use snippets of code from these demos to plug the demo's functionality into their own applications.

Each of the Swing demos is located in a separate directory in the [installdirectory]\examples\swing directory.

Before running the demos, you must first compile them into class files using your IDE compiler or using the Java command-line compiler, `javac`. Then you can execute the resulting class within your IDE or from a command prompt using the Java command `java`.

The Swing demos are listed here by their directory names.

- **examples\swing\demos\amortization.** This demo shows how to use Formula One in a simple web-based application. It converts spreadsheets to HTML or JPG files on the fly using servlet and page-compilation technology.
- **examples\swing\demos\calculate.** This page compilation demo shows how to call the `JBook` API from within a JHTML page. It displays a form and asks the user for a formula. That formula is passed to `JBook`, calculated, and returned to the user.
- **examples\swing\samples\add-in\concat.** This demo shows two ways to handle threading issues in add-in functions. The demonstration functions concatenate strings.

- **examples\swing\samples\add-in\iseven.** This demo shows how to write a simple add-in function that determines whether the entry in a worksheet cell is even or odd.
- **examples\swing\samples\add-in\mydosum.** This demo shows how to write an add-in function that accesses a range of cells in a spreadsheet to perform a simple sum. It demonstrates how to handle cell references and ranges in an add-in function.
- **examples\swing\samples\apps\edi.** This demo shows Formula One reading in external data (EDI-formatted addresses stored in VTS, XLS, and TXT file formats) and merging that data with a template spreadsheet. Each of the files `EastRegion.*`, `WestRegion.*`, and `MidwestRegion.*` has address information for 9 customers in the first 9 rows of the file (first column for spreadsheet formats). As the user scrolls though each of the customers, the names are extracted from the data files, merged with the spreadsheet template (`Default.vts`), recalculated, and displayed.
- **examples\swing\samples\apps\format.** This demo shows how to use the formatting API (specifically, the `CellFormat` object) to set borders, text colors, background colors, etc. The application displays several buttons next to a Formula One worksheet. It formats the worksheet data differently based on the button pressed. This action is similar to the AutoFormat feature of Excel.
- **examples\swing\samples\apps\hr.** This demo is a human resources application designed to collect time sheet information stored in VTS files (Formula One's native file format). It illustrates how to programmatically create, delete, and display worksheets.
- **examples\swing\welcome.** The code in "Sample Formula One Swing Application and Applet" on page 27 of this guide.
- **examples\oracle\webtogo\vacation.** This servlet accesses an Oracle database to track employee vacation requests made both on and offline and route the requests to the appropriate manager for approval. Formula One is used to retrieve the data from the database via JDBC, place the data into a worksheet, summarize it using worksheet functions, then write this summary information out in response to the browser request.

Upgrading to Formula One 7.0

Since the 5.5 version of Formula One uses AWT-based classes, upgrading from Formula One 5.5 to the 7.0 AWT component is relatively straightforward. Upgrading from 5.5 to the 7.0 Swing component is more involved, requiring class and package name changes and other changes.

Note Spreadsheets created in the Formula One 5.5 file format can be read by Formula One 7.0 with no changes needed. However, spreadsheets in the improved 7.0 file format cannot be read by 5.5. The 7.0 version of the software provides the option of saving files in the 5.5 format if you need them to be backwards-compatible.

Upgrading to Formula One's 7.0 AWT Component

Java code that was written using Formula One 5.5 can be used as-is in Formula One 7.0. You do not need to make any coding changes because Formula One's 5.5 and 7.0 versions contain the same AWT packages and classes.

However, since the name and contents of Formula One's JAR file have changed, you need to change the name of the JAR file referenced by your IDE and any HTML pages that contain Formula One applets.

Changes Within Your IDE

Since the name and contents of Formula One's JAR file have changed, you need to change the name of the JAR file referenced by your IDE. Upgrading to a new JAR file (in this case, either `F1J7AWT.jar` or `F1J7AWTDesign.jar`) within an IDE means adding the JAR file as a JavaBean component. This is the same process as adding a new JavaBean. For information on how to add a JavaBean component to your IDE, see the file `f1bean.html`, located in the download, on the CD, or on Tidestone's website (www.tidestone.com).

If your IDE already contains the Formula One 5.5 JavaBean, we suggest you remove it before adding the 7.0 JavaBean. The `f1bean.html` file contains instructions for removing JavaBean components. It is possible to have both JARs available in one IDE; however, the icons for the two beans are the same, which may cause confusion.

Changes in HTML Pages Containing Applets

Replace the name of the JAR file referenced in the `ARCHIVE` parameter of the `<APPLET>` tag. If you reference `F1J.jar`, replace it with `F1J7AWT.jar`. If you reference `F1JDesign.jar`, replace it with `F1J7AWTDesign.jar`.

For sample HTML code to run Formula One 7.0 AWT applets, see “Running the AWT sample code as an applet” on page 19.

Upgrading to Formula One’s 7.0 Swing Component

To make Formula One 5.5 projects work with Formula One’s 7.0 Swing component, you will need to change:

- the Java code
- the JAR file referenced by your IDE
- HTML pages that contain Formula One applets

Changes to Java code

Packages, classes, and methods have been changed, relocated, and in some cases removed in the 7.0 Swing version of Formula One. In order to use Java code written for Formula One 5.5, you need to search and replace package and class names.

The most important of these replacements is the basic class you use to display a workbook: it has changed from `com.f1j.View` to `com.f1j.swing.JBook`. In some small cases you may need to rewrite code for APIs that are no longer available. Then recompile the project.

The table below shows the packages, classes, and methods from 5.5 that need to be changed in order for Java code to work in 7.0.

5.5 Formula One API (AWT-based)	7.0 Formula One API (Swing-based)

`import com.f1j.*`	`import com.f1j.util.*;`
	`import com.f1j.swing.*;`
	`import com.f1j.ss.*`

`f1j.CellFormat`	`f1j.ss.CellFormat`
`f1j.CellFormat.setNumberFormat`	`f1j.ss.CellFormat.setValueFormat`
`f1j.CellRef`	`f1j.ss.CellRef`
`f1j.Database`	`f1j.calc.Database` (Constructor is now null.)
`f1j.designer.ClassFactory`	Not available.
`f1j.dialog.ClearDlg`	`f1j.swing.ss.ClearDlg`

5.5 Formula One API (AWT-based)	7.0 Formula One API (Swing-based)
f1j.dialog.ColWidthDlg	f1j.swing.ss.ColWidthDlg
f1j.dialog.DefaultFontDlg	f1j.swing.ss.DefFontDlg
f1j.dialog.DefColWidthDlg	f1j.swing.ss.DefColWidthDlg
f1j.dialog.DefinedNameDlg	f1j.swing.ss.DefinedNameDlg
f1j.dialog.DefRowHeightDlg	f1j.swing.ss.DefRowHeightDlg
f1j.dialog.DeleteDlg	f1j.swing.ss.DeleteDlg
f1j.dialog.FindReplaceDlg	f1j.swing.ss.FindDlg and f1j.swing.ss.ReplaceDlg
f1j.dialog.FormatCellsDlg	f1j.swing.ss.FormatCellsDlg
f1j.dialog.FormatObjectDlg	f1j.swing.ss.FormatObjectDlg
f1j.dialog.FormatSheetDlg	f1j.swing.ss.FormatSheetDlg
f1j.dialog.GotoDlg	f1j.swing.ss.GotoDlg
f1j.dialog.InsertDlg	f1j.swing.ss.InsertDlg
f1j.dialog.JDBCDlg	f1j.swing.jdbc.JDBCDlg
f1j.dialog.OpenFileDlg	javax.swing.JFileChooser (Not a Formula One class.)
f1j.dialog.OptionsDlg	f1j.swing.ss.OptionsDlg
f1j.dialog.PageSetupDlg	f1j.swing.ss.PageSetupDlg
f1j.dialog.PasteSpecialDlg	f1j.swing.ss.PasteSpecialDlg
f1j.dialog.RowHeightDlg	f1j.swing.ss.RowHeightDlg
f1j.dialog.SaveFileDlg	javax.swing.JFileChooser (Not a Formula One class.)
f1j.dialog.SortDlg	f1j.swing.ss.SortDlg
f1j.F1Exception	f1j.util.F1Exception
F1j.FindReplaceInfo	f1j.ss.FindReplaceInfo
f1j.GRObject	f1j.ss.GRObject
f1j.GRObjectPos	f1j.ss.GRObjectPos
f1j.JDBC	f1j.jdbc.JDBC
f1j.JDBCQueryObj	f1j.jdbc.JDBCQueryObj
f1j.RangeRef	f1j.ss.RangeRef
f1j.View	f1j.swing.JBook
f1j.View.setLaunchWorkbookDesigner	f1j.swing.JBook.launchDesigner

5.5 Formula One API (AWT-based)	7.0 Formula One API (Swing-based)

`f1j.View.setParentFrame`	Does not exist. Instead use:
	`JFrame f= new JFrame();`
	`f.getContentPane().add(Jbook1);`

`f1j.FindReplaceInfo`	`f1j.ss.FindReplaceInfo`
`f1j.HTMLWriter`	`f1j.ss.HTMLWriter`
`f1j.NumberFormat`	`f1j.ss.NumberFormat`
`f1j.*Event`	`f1j.swing.*Event`
`f1j.*Listener`	`f1j.swing.*Listener`

Changes Within Your IDE

Since the name and contents of Formula One's JAR file have changed, you need to change the name of the JAR file referenced by your IDE. Upgrading to a new JAR file (in this case, `F1J7Swing.jar`) within an IDE means adding the JAR file as a JavaBean component. This is the same process as adding a new JavaBean. For information on how to add a JavaBean component to your IDE, see the file `f1bean.html`, located in the download, on the CD, or on Tidestone's website (www.tidestone.com).

If your IDE already contains the Formula One 5.5 JavaBean, we suggest you remove it before adding the 7.0 JavaBean. The `f1bean.html` file contains instructions for removing JavaBean components. It is possible to have both JARs available in one IDE; the icons are different so you can tell them apart.

Changes in HTML Pages Containing Applets

You'll need to make three types of changes to your HTML pages.

1. Replace the name of the JAR file referenced in the `ARCHIVE` parameter of the `<APPLET>` tag to `F1J7Swing.jar`.
2. Convert the HTML pages that use the `<APPLET>` tag to use both the `<OBJECT>` and `<EMBED>` tags, either by hand or by using Sun's HTML Converter.
3. Be aware that Swing applets must be deployed using Sun's Java Plug-in.

For more information and sample HTML code to run Formula One 7.0 Swing applets, see "Running the Swing sample code as an applet" on page 28.

Licensing Formula One

You can license Formula One for use on a single computer (as a user or developer); however, Formula One is usually licensed based on how an application that contains Formula One is deployed after development.

You can find specific licensing information online at the Tidestone Technologies website, www.tidestone.com, by e-mail at sales@tidestone.com, or by calling the Tidestone sales department at (800) 884-8665.

Software License Agreement

Product: Formula One - Developer License
(no redistribution or deployment of Software or documentation permitted)

Developer Licenses: 1

This is your proof of license.

Tidestone License Agreement

IMPORTANT - READ CAREFULLY: This is a legal agreement between You (either an individual or an entity), and Tidestone Technologies, Inc. ("Tidestone") for the Tidestone software product identified above, which includes computer software and may include associated media, printed materials, and electronic documentation ("Software"). Please read these terms carefully. By opening the sealed media package, installing, copying or otherwise using the Software, You agree to be bound to the terms of this agreement completely. If you do not agree to the terms of this Agreement, do not install or use the Software and promptly return the product to Tidestone for a full refund.

LICENSE AGREEMENT

1. GRANT OF LICENSE. Tidestone grants to You a personal, non-exclusive, non-transferable license to use the Software in the quantity indicated above. A Developer License is required for each computer on which the Software is installed or run for development purposes. You may store or install a copy of the Software on a storage device, such as a network server, used only to install or run the Software on your other computers over an internal network; however, you must have a license for each separate computer on which the Software is installed or run from the storage device. A license for the Software may not be shared or used concurrently on different computers.

2. COPYRIGHT. The Software and accompanying documentation is owned by Tidestone and is protected by United States copyright laws and international treaty provisions. Therefore, You must treat the Software and accompanying documentation like any other copyrighted material (e.g. a book or musical recording), and may be legally responsible for copyright infringement.

3. OTHER RESTRICTIONS. This License Agreement is Your proof of license to exercise the rights granted herein and must be retained by You. You may not loan, rent, lease, transfer or sublicense the Software or any copy. You may not modify, alter, reverse engineer, decompile or disassemble the Software. You may not at any time disclose or disseminate the Software to any person who does not need to attain access thereto consistent with Your rights under this license. When installing the Java version of the Software, Sun Microsystem's Java Runtime Environment software (JRE) may be installed on Your computer. When it is installed, you are subject to the terms and conditions of Sun's license agreement for the JRE, which is installed with the Software and can be found in the file Sunlicen.txt, or at http://java.sun.com/products/jdk/1.1/jre/license.

LIMITED WARRANTY

LIMITED WARRANTY. This Software is licensed AS IS without warranty of any kind; if for any reason you are dissatisfied with it, return the software, including all materials, to Tidestone within 30 days of the date of purchase for a full refund. If any materials or media in this package are defective, under normal use, return them within 90 days of the original date of purchase, and Tidestone will replace them at no charge. These warranties are in lieu of any other warranties, express or implied, including the implied warranties of merchantability and fitness for a particular purpose, which are expressly disclaimed. This limited warranty gives You specific legal rights. You may have other rights, which vary from state to state. Tidestone hereby limits the duration of any implied warranties to the periods set forth above. Some states do not allow limitations on the duration of an implied warranty, so the above limitations may not apply to You.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall Tidestone be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this product, even if Tidestone has been advised of the possibility of such damages. Because some states/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to You.

U.S. GOVERNMENT RESTRICTED RIGHTS

The Software and documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227- 7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/manufacturer is Tidestone Technologies, Inc., at the address below.

MISCELLANEOUS

If you acquired this product in the United States, this Agreement shall be governed by the laws of the State of Kansas. If this product was acquired outside the United States, then local law may apply.

Should you have any questions concerning the Software or this license agreement, please contact the Tidestone subsidiary serving your country or write to Tidestone Technologies, Inc., 12980 Metcalf Avenue, Suite 300, Overland Park, Kansas 66213.

Tidestone

Index

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W