

**RapidPLUS™ Xpress
User Manual**

RapidPLUS™ Xpress User Manual

© 2003 e-SIM Ltd. All rights reserved.

e-SIM Ltd.
POB 45002
Jerusalem
91450
Israel

Tel: 972-2-5870770

Fax: 972-2-5870773

Information in this manual is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this manual is furnished under a license agreement and may be used or copied only in accordance with the terms of that agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of e-SIM Ltd.

Microsoft, Windows, Windows NT, Visio, and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Java and the Java logo are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.

Other product and company names mentioned in this manual may be trademarks or registered trademarks of their respective owners.

Contents

PREFACE	vii
Requirements	vii
About this User Manual	vii
Conventions Used in This Manual	vii
CHAPTER 1: INTRODUCING RAPIDPLUS XPRESS	1
The RapidPLUS Xpress Tools	3
From Application to Screen Transition Chart	6
Designing RapidPLUS Xpress Prototypes for RapidPLUS Applications	8
CHAPTER 2: TUTORIAL 1—BUILDING A BASIC XPRESS APPLICATION	9
Launching RapidPLUS Xpress	10
The RapidPLUS Xpress Tools	11
Stage 1: Defining the System's States Using Modes	12
Developing States Using RapidPLUS Xpress Modes	13
Saving the Application	14
Stage 2: Defining Background Graphics	14
Adding Graphics to the Application	15
Attaching the Frame Object to the Mode Tree	18
Viewing the Background Frames in the Screen Transition Chart	18
Stage 3: Designing the Screen Elements	20
Designing the Screen for Off Mode	20
Designing the Screen for Powering On Mode	20
Designing the Screen for On Mode	24
Designing the Screen for Powering Off Mode	24

- Stage 4: Attaching Screen Elements to Modes 25
 - Attaching Screen Elements to Powering_On Mode 25
 - Attaching Screen Elements to On Mode 25
 - Attaching Screen Elements to Powering_Off Mode 26
 - Viewing the Screens in the Screen Transition Chart 27
- Stage 5: Defining Transitions 28
 - Defining the Transition from CellPhone1 to Off Mode 28
 - Defining the Transition from Off Mode to Powering_On Mode..... 28
 - Defining the Transition from On Mode to Powering_Off Mode..... 31
 - Defining the Transition from Powering_On Mode to On Mode..... 32
 - Defining the Transition from Powering_Off Mode to Off Mode..... 34
 - Designing the Charts 34
 - Viewing the Screens in the Screen Transition Chart 36
- Running the Application in the Prototyper..... 37
- Saving System Settings 38
- CHAPTER 3: TUTORIAL 2—EXPANDING THE BASIC APPLICATION 39**
- Starting the Tutorial 41
- Stage 1: Defining the System’s Modes 41
 - Using Concurrent Modes 42
 - Adding Modes to the Tree..... 43
- Stage 2: Defining the Battery Functionality 48
 - Adding the Battery Graphics 48
 - Attaching the Battery Graphics to the Battery Modes 49
 - Defining Transitions for the Battery..... 50
 - Testing the Application in the Prototyper 52
 - Viewing the State Chart..... 53
 - Viewing the Transition Charts..... 54
- Stage 3: Defining the Backlight Functionality 56
 - Adding the “Illuminated” Frame Object 56
 - Attaching the Illuminated Frame to Backlight/On Mode 57
 - Defining the Transitions for the Backlight 58
 - Testing the Application in the Prototyper 61
 - Viewing the Screen Transition Chart 61

Stage 4: Defining The Functionality for Ready Mode	63
Designing the Screen for Idle Mode	64
Designing the Screens for Dial Mode	66
Defining Transitions for Ready Mode	69
Testing the Application in the Prototyper	71
Viewing the Screen Transition Charts	72
Stage 5: Defining the Functionality for Call_Control Mode	74
Defining the Screen for Send_Call Mode	74
Defining the Screen for Receive_Call Mode	76
Defining the Screen for Talk Mode	77
Defining the Screen for End_Call Mode	78
Testing the Application in the Prototyper	80
Viewing the Screen Transition Charts	81
CHAPTER 4: WORKING IN RAPIDPLUS XPRESS	85
Workflow for Developing an Application	86
RapidPLUS Xpress Objects	88
Defining States	89
Attaching Graphic Objects to the Mode Tree	90
Attaching Nongraphic Objects to the Mode Tree	92
Behavior of Attached Objects	92
Working with Attachments	93
Deactivating Attached Objects in a Mode Tree	96
Working With Transitions	97
Making Transitions	99
Deleting Transitions	100
Creating Screen Transition Charts	100
Screen Transition Charts	101
Modifying the Appearance of Screen Transition Charts	102
Single-State Transition Charts	104
Modifying the Appearance of Single-State Transition Charts	105
Exporting Transition Charts	108
Adding Notes to Screen Transition Charts	109

Working With Transition Charts in Microsoft Visio	112
Modifying Transition Charts	112
Printing Transition Charts	113
Distributing Transition Charts Over the Web	113
Tips for Building Successful Applications.....	114
Changing the Order of Modes.....	114
Building Uniform States in Screen Transition Charts	115
Team Development of RapidPLUS Xpress Applications.....	117
Developing an Application Without User Objects	117
Developing an Application With User Objects	119
Using Common Graphics During Team Development	125
Adding Supplemental Logic (using RapidPLUS)	125
Adding Logic	126
Using Logic That is Accessible in Standard RapidPLUS Only.....	126
INDEX	127

PREFACE

Requirements

Microsoft® Visio® 2000 or higher must be installed on your system in order to view, edit, and print generated screen transition charts.

About this User Manual

This manual provides the information you need to build RapidPLUS Xpress applications. It comprises the following chapters:

- **Chapter 1: “Introducing RapidPLUS Xpress”**
This chapter provides an overview of RapidPLUS Xpress.
- **Chapter 2: “Tutorial 1—Building a Basic Xpress Application”**
This tutorial will teach you how to use the RapidPLUS Xpress tools to build a simple prototype. It will also introduce you to the screen transition charts that are generated in RapidPLUS Xpress and viewed in Microsoft Visio 2000 or higher.
- **Chapter 3: “Tutorial 2—Expanding the Basic Application”**
This tutorial expands on the first tutorial to build a more complex prototype. It introduces you to the RapidPLUS Xpress hierarchical structuring capabilities that enable fast construction of product prototypes. It also teaches you how to work efficiently with the RapidPLUS Xpress tools and features.
- **Chapter 4: “Working in RapidPLUS Xpress”**
This chapter provides instructions for using features specific to RapidPLUS Xpress and for creating screen transition charts. It should be used as a reference *after* you have performed Tutorials 1 and 2.

Conventions Used in This Manual

This documentation uses the following conventions:

Menu Conventions

- “Choose File|Import Bitmaps” means to select the Import Bitmaps command from the File menu.
- All of the RapidPLUS Xpress tool windows have a File menu, from which you can open and save applications, and exit RapidPLUS Xpress.

Typographic Conventions

- Information to be typed into dialog boxes appears in **Courier New, bold**.
- Information to be selected from lists in dialog boxes appears in **bold**.
- File names and extensions appear in *italics*.

Pictures of Toolbar Buttons and Object Buttons



Ctrl+A

When a step instructs you to click a toolbar button or object button in the Object Palette, a picture of the button appears to the left of the step (as shown here). If the toolbar button has a shortcut key, the shortcut key is presented below the picture.

Tips

Information about designing and building RapidPLUS Xpress applications, separate from procedures, appears between double lines.

This manual was written with the assumption that you are familiar with Microsoft® Windows® conventions and with Microsoft® Visio® 2000 (or higher).

Introducing RapidPLUS Xpress

Welcome to RapidPLUS™ Xpress—the tool that helps HMI (Human Machine Interface) designers, product managers, marketing managers, and other people involved in product definition and design to quickly and easily create working, interactive graphical models of their future products. This ease of use includes outlining a product’s functionality, prototyping the product, and generating screen transition charts from the prototype.

RapidPLUS Xpress lets you quickly realize your design concepts and ensure that your ideas look and behave as you envisioned them. It also saves you a lot of paperwork because it can automatically create the screen transition charts that document your ideas. With RapidPLUS Xpress, you can modify and test your design at any stage of development. Once your product design is finished, the prototype will ensure that the people who will be implementing your design understand exactly what you envisioned.

RapidPLUS Xpress is easy to learn and requires no programming experience. The basic steps for designing your product with RapidPLUS Xpress are:

1. Define the different states (comprised of RapidPLUS Xpress “modes”) the product can be in. For example, a cell phone can be in the following states: “On,” “Off,” and “Dialing.” The On state could consist of just one mode, “On” or it could comprise several more modes, such as “ReadyForInput” and “Idle.”
2. Define the objects that will appear in the product prototype. Objects can be: (i) graphics of the product and its different screens; (ii) graphic input elements such as pushbuttons and switches; and (iii) nongraphic input elements such as timers and sounds.

3. Define the objects that will appear in the screen transition charts by attaching the graphic objects to the appropriate modes. For example, in the On state, various messages (such as company name, the current time, and soft key text) will appear on the cell phone screen in the chart. Objects that would be attached to the mode(s) include a bitmap of the screen and label objects that hold text.
4. Define transitions between modes (such as from “Off” to “On”) and the triggers that cause them to happen (such as the PowerOn pushbutton being pressed).

That’s it! After these steps are completed you can run the prototype and test its behavior. If you are not satisfied with the result, you can easily repeat these steps until you are satisfied.

At any point in the development of your prototype, you can distribute the result of your efforts throughout your organization. You can distribute the prototype via the Internet or a file. You can distribute clear, graphic screen transition charts that capture your design ideas in a printable format. These screen transition charts are generated in RapidPLUS Xpress and exported to Microsoft® Visio® (2000 or above).

RapidPLUS Xpress is the way to realize your design ideas quickly and ensure that everyone involved in the development of the product understands your design.

This chapter presents:

- An introduction to the RapidPLUS Xpress tools.
- How an application’s states appear when they are generated to screen transition charts and exported to Microsoft Visio.
- How RapidPLUS Xpress prototypes are used as guidelines for RapidPLUS applications.

THE RAPIDPLUS XPRESS TOOLS

A RapidPLUS Xpress application is a virtual prototype. It presents a product's user interface and the functionality of each of the prototype's states.

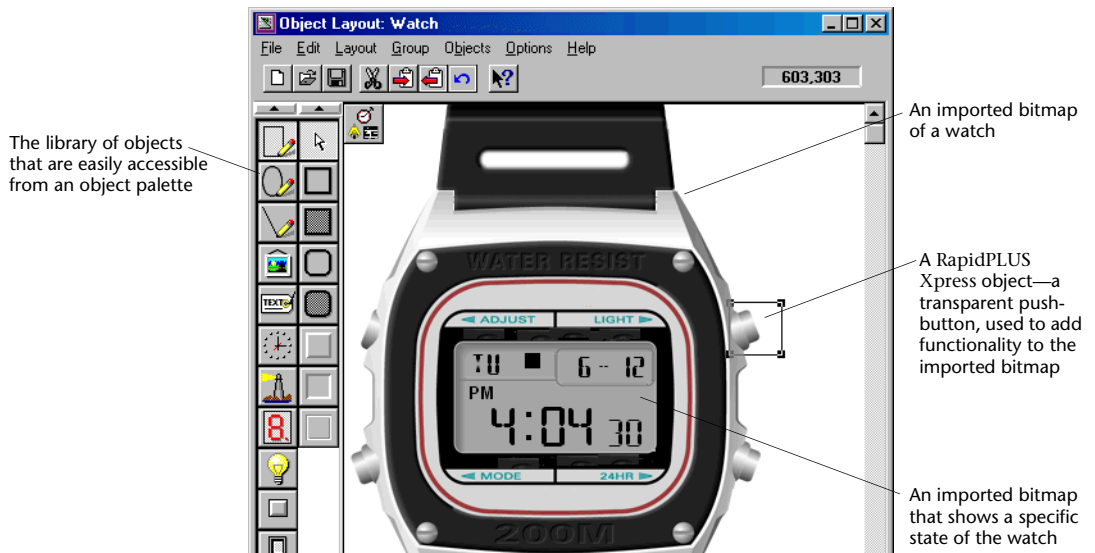
RapidPLUS Xpress comprises several tools, each of which is used to develop a different aspect of the application. Together, they produce an application made up of user interface elements, states, and transitions among states. The tools that comprise RapidPLUS Xpress are:

- Object Layout
- Mode Tree
- State Chart
- Prototyper

An additional tool, the Object Editor, is available for customizing graphic elements of the user interface.

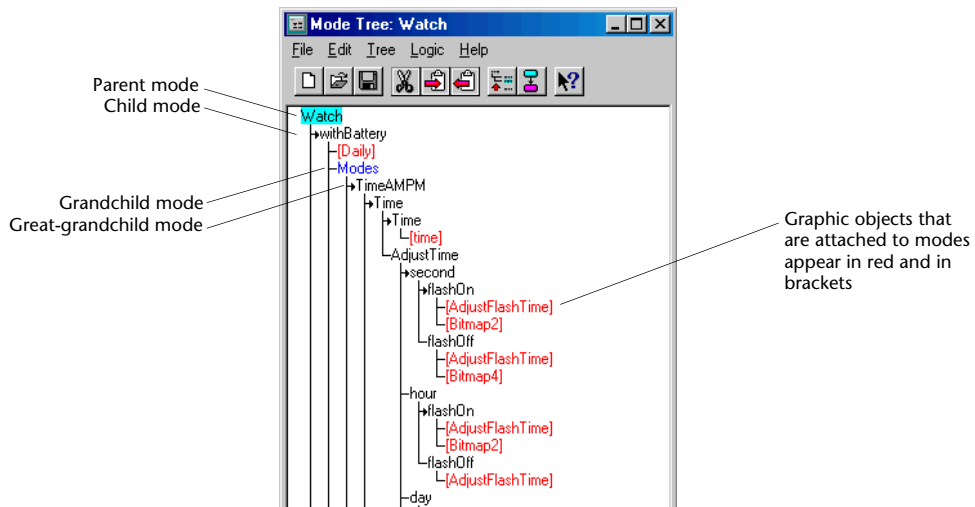
The Object Layout Tool

The Object Layout tool is used to develop the prototype's user interface. It can import graphics created in other graphics programs and has its own library of objects. Here is an example of a watch that was designed in the Object Layout:



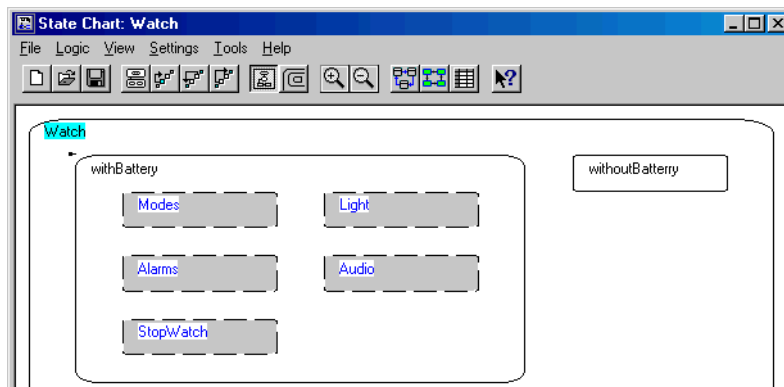
The Mode Tree Tool

The Mode Tree tool is used to develop the prototype's functionality. It presents a hierarchical tree structure comprising modes: parent mode → child modes → grandchild modes, etc. Graphic objects are attached to the modes to create the look of each state in screen transition charts. Here is the tree for the Watch application:



The State Chart Tool

The State Chart tool is used to present the modes graphically. It has options for modifying the appearance of the exported screen transition charts. Here is the state chart for the Watch application:



The Prototyper Tool

The Prototyper tool is used to run and test the application. When the Watch application runs in the Prototyper, its various buttons can be pressed and the display changes accordingly.

In the following illustration, a button has been pressed to start the stopwatch feature:



FROM APPLICATION TO SCREEN TRANSITION CHART

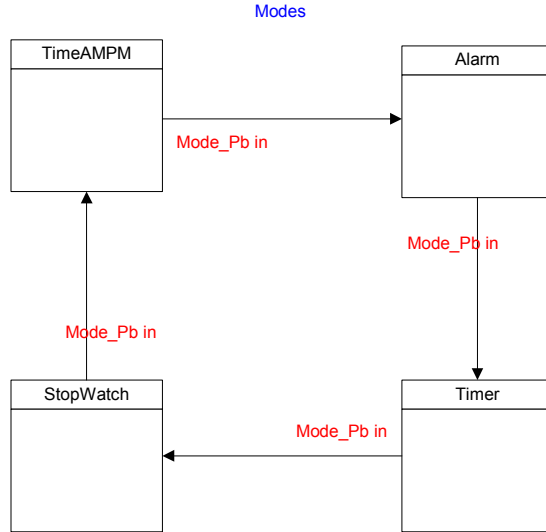
A screen transition chart is a diagram that displays states and transitions among states. Each state is presented as a rectangle, and whatever graphic objects are attached to its modes appear in the rectangle. The transitions are presented as arrows with accompanying text that describes each transition (for example, “Pushbutton1 in”). Graphic representation of the object that triggers a transition may also appear in the chart (depending on settings made in the State Chart tool).

Transitions are made in the Object Layout, Mode Tree, and State Chart tools. The transitions are important because they cause the application to change states. They determine how the prototype functions. For example, a transition from an Off state to an On state can be caused by pressing a button or operating a switch.

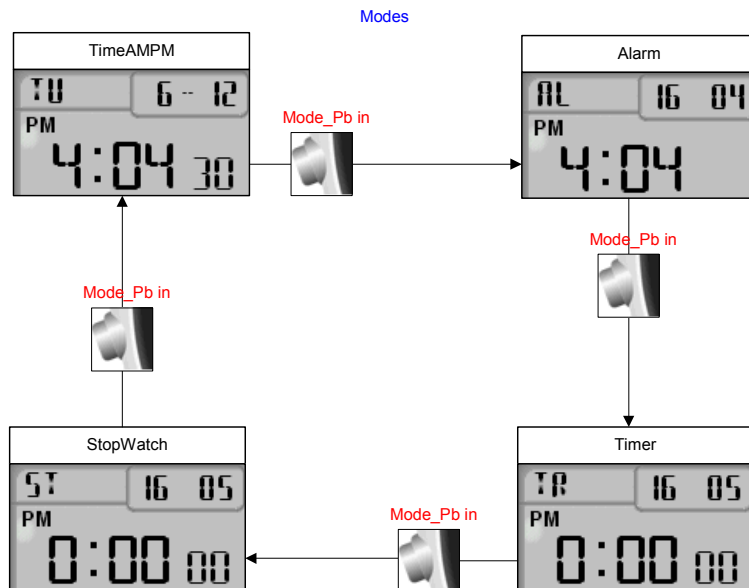
To create a screen transition chart, RapidPLUS Xpress takes a state chart created in the State Chart tool, generates it into a screen transition chart, and exports it to Microsoft Visio (2000 and higher). In Visio, the screen transition chart can be manipulated using the e-SIM-supplied Visio stencil, *esimstd.vss*.

The appearance of the screen transition charts can be enhanced by attaching graphics to the modes and using options in the State Chart tool. The illustrations on the next page show two charts in Visio, one basic and the other enhanced.

Example of a basic screen transition chart

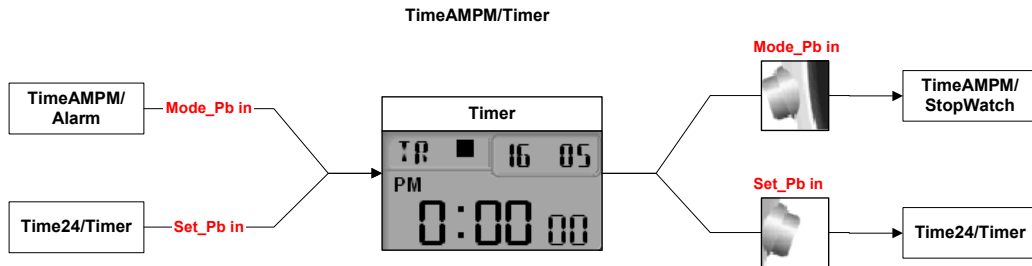


Example of an enhanced screen transition chart



A transition chart can also be generated for states that do not have child states. This type of chart, a single-state screen transition chart, shows the transitions to and from the state:

Example of a single-state screen transition chart



DESIGNING RAPIDPLUS XPRESS PROTOTYPES FOR RAPIDPLUS APPLICATIONS

RapidPLUS Xpress is the ideal tool for designing prototypes of products that will be developed in RapidPLUS, e-SIM's software package for building **functional** product prototypes.

e-SIM offers several products for producing product prototypes. The basic product, called RapidPLUS DOC, is used for building simulations and generating documentation for user interface specifications and test procedures. Another product, called RapidPLUS CODE, enables generation of C code from the RapidPLUS application. The generated C code can be integrated into the actual product. A third product, called RapidPLUS Web Studio, enables generation of Java code from the RapidPLUS application to produce simulations that can be viewed in a Web browser.

The RapidPLUS Xpress application supplies the information needed—and some building blocks—to begin building the RapidPLUS application. The RapidPLUS Xpress application can be opened in any of the RapidPLUS products, and then be modified and enhanced.

Tutorial 1—Building a Basic Xpress Application

In this tutorial, you will launch RapidPLUS Xpress and then build a very simple cell phone in RapidPLUS Xpress and create a screen transition chart from the cell phone application. The tutorial is divided into five stages:

Stage 1: Defining the System's States Using Modes

You define four states for the cell phone: off, powering on, on, and powering off.

Stage 2: Defining Background Graphics

You define a background bitmap that will appear in the screen transition chart as the background for all four states.

Stage 3: Designing the Screen Elements

You design the elements that appear on each state's screen. For example, the screen in the "powering on" state will present a Welcome message; the screen in the "powering off" state will present a Good-bye message.

Stage 4: Attaching Screen Elements to Modes

In the previous stage, you designed the screen elements; in this stage you enable them to appear in the screen transition chart.

Stage 5: Defining Transitions

You define transitions that cause the system to move from state to state. The tutorial should take 1½–2 hours.

Enjoy learning RapidPLUS Xpress!

LAUNCHING RAPIDPLUS XPRESS

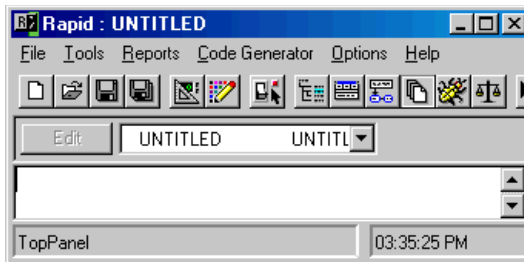
The following instructions present how to launch the stand-alone RapidPLUS Xpress program and how to launch it from within RapidPLUS.

To launch the stand-alone RapidPLUS Xpress program:

1. From the Windows Start menu, choose Programs|Rapid 7.0|RapidPLUS Xpress. The names on your system may be different depending on your choices during installation.
2. RapidPLUS Xpress opens. Proceed to the next section, “The RapidPLUS Xpress Tools.”

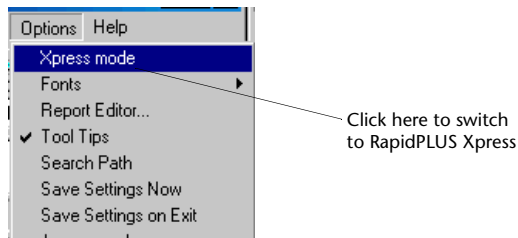
To launch RapidPLUS Xpress from within RapidPLUS:

1. From the Windows Start menu, choose Programs|Rapid 7.0|RapidPLUS. The names on your system may be different depending on your choices during installation. Three tool windows open.
2. Click in the small window in the upper-left corner titled Rapid: UNTITLED to make it the active window:



This window, which is referred to as the Application Manager, is used for opening the various tools in RapidPLUS and for setting general options.


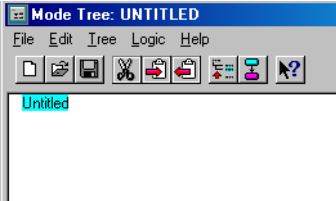

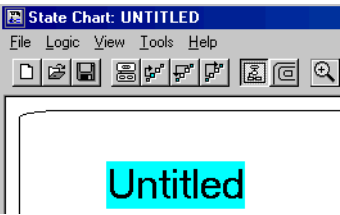

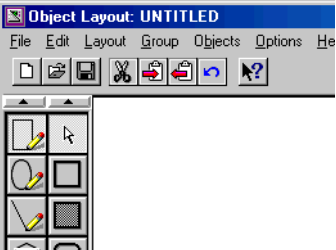
3. Choose Options|Xpress mode:



The screen configuration changes and the three main tools that are used in RapidPLUS Xpress open. Also the Rapid window's toolbar changes so that only the tools and options relevant to RapidPLUS Xpress are available.

The RapidPLUS Xpress Tools

The main tools in RapidPLUS Xpress are presented in the following table. When their windows are closed, you can open them from the Rapid window using the toolbar buttons shown below.

BUTTON	TOOL	PURPOSE
 Ctrl+T		The Mode Tree tool is used for organizing a system's states and determining each state's appearance.
 Ctrl+Shift+C		The State Chart tool graphically presents a system's states and the transitions among states.
 Ctrl+A		The Object Layout tool is used to add the objects and images that appear in the application and in the transition charts.

Another tool, the Prototyper, will be used at the end of the tutorial to run and test the application.

- ❖ *RECOMMENDATION: Don't close any of the tool windows or move them from their default positions while performing this tutorial.*

STAGE 1: DEFINING THE SYSTEM'S STATES USING MODES

To begin developing a RapidPLUS Xpress application for a system, you must first consider the system's functionality. In RapidPLUS Xpress, a system's functionality is defined in terms of **modes**. Modes are the distinct units of functionality that describe the system's behavior.

For example, a cell phone can be on or off; therefore, On and Off are both modes of the cell phone. The behavior of On mode can be described by additional modes such as Idle, Place a Call, Talk, End a Call, and Menu Functions. The behavior of Menu Functions mode can be described by additional modes such as Main Menu and Phone Book Menu. Phone Book Menu can be further described by modes such as Add Entry, Delete Entry, and so on.

In the first stage of developing a RapidPLUS Xpress application, you list the system's modes. Then you build these modes in the Mode Tree tool. A mode tree is a hierarchical representation of a system's behavior. It is used to organize a system's structure logically and efficiently.

A snapshot of a system at a given time is called a **state**. A state can capture the system:

- When only one mode is active, such as when the mode Off is active.
- When many modes are active, such as when the modes On—Menu Functions—Phone Book Menu—Add Entry are all active. Each of these modes presents different information about the On state.

In this tutorial, you will focus on states that capture the behavior of single modes. In other words, the state and the mode will be equivalent. In the second tutorial, you will work with states that capture the functionality of more than one mode.

Developing States Using RapidPLUS Xpress Modes

Although a real cell phone has many modes, at this stage in the tutorial you will focus on its four most basic modes: Off, Powering_On, On, Powering_Off. Using these modes, you will build a mode tree in the Mode Tree tool.

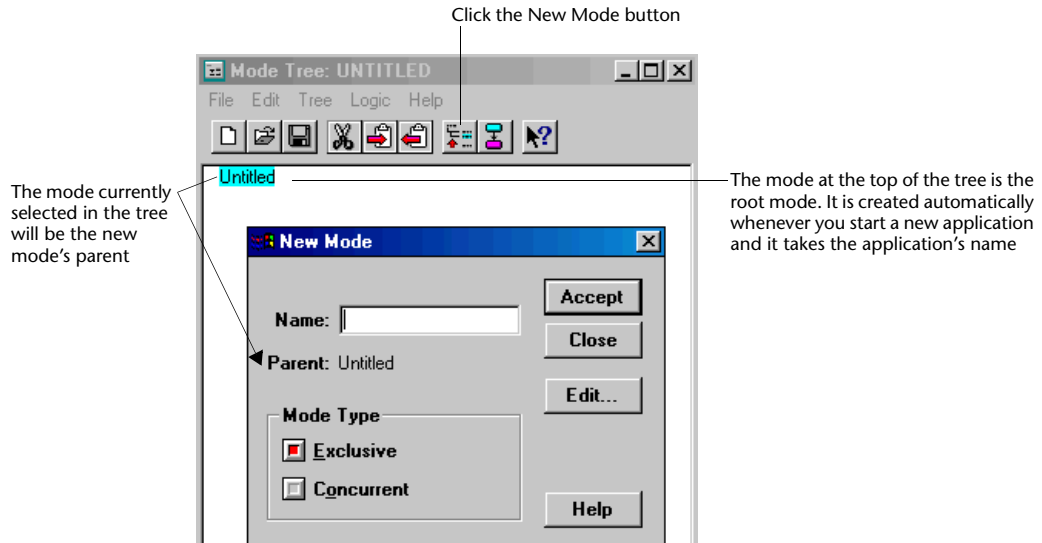
You'll start with "Off" because it is the default mode of the cell phone.

To add modes:

1. Click in the Mode Tree to make it the active window.
2. Choose Tree|New Mode, or click the New Mode button; the New Mode dialog box opens:



Ctrl+W



3. In the Name box, type **Off** and click **Accept**.

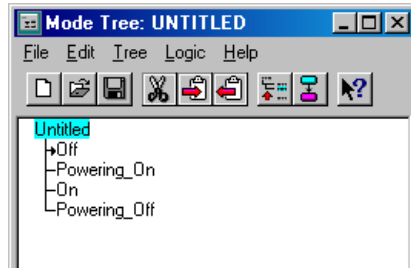
Off mode is added to the tree as a child of Untitled, which refers to the current application that has not yet been saved. The New Mode dialog box remains open.

Look at the State Chart tool. When you added Off to the tree, it was also added to the State Chart, which displays the modes graphically. Because Off is a child of Untitled, it is nested within Untitled.

4. In the Name box, type **Powering_On** and click **Accept**.
5. Type **On** and click **Accept**.

6. Type **Powering_Off** and click , then click .

The completed tree looks like:



Saving the Application

After you have completed each stage in the tutorial, save the application. **You can perform the Save operation from any of the windows.**

To save the application:

1. Choose File|Save, or click the Save button; the Save As dialog box opens.
2. Save the application to the \\Rapid70\Tutorials\RapidPLUSXpress folder and name it **CellPhone1**.



Ctrl+S

STAGE 2: DEFINING BACKGROUND GRAPHICS

In this stage, you will add a graphic of the cell phone—an imported bitmap—to provide realism for the prototype. You do not need the full graphic for the transition charts, so you will add a separate Rapid graphic—a frame object—to present the cell phone's screen. In order for the frame object to appear in the transition charts, you must attach it to the upper-most mode, the root mode.

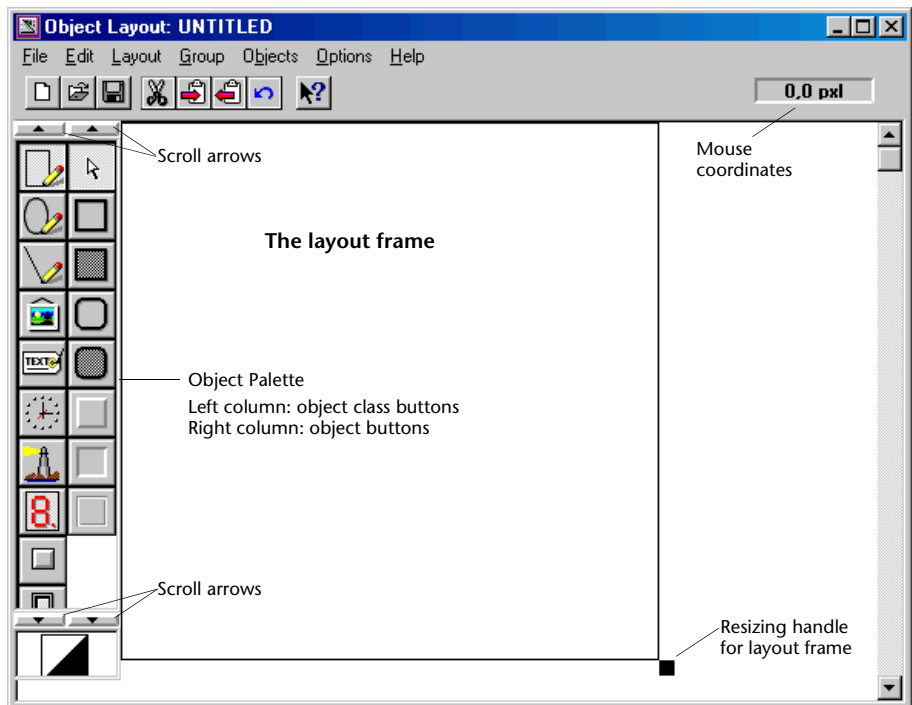
Graphic objects are added to an application in the Object Layout tool. They provide a visual representation of the system's user interface (such as switches, pushbuttons, lamps, imported images).

Adding Graphics to the Application

You will add the following two graphics:

- A bitmap of the cell phone that you will import into RapidPLUS Xpress. This bitmap will add realism and usability to the application.
- A frame object that will be the background for the states in screen transition charts.

Before you begin, take a look at the Object Layout window. Notice the Object Palette on the left side of the window. You will use it to select most of the objects that you will place on the layout frame.

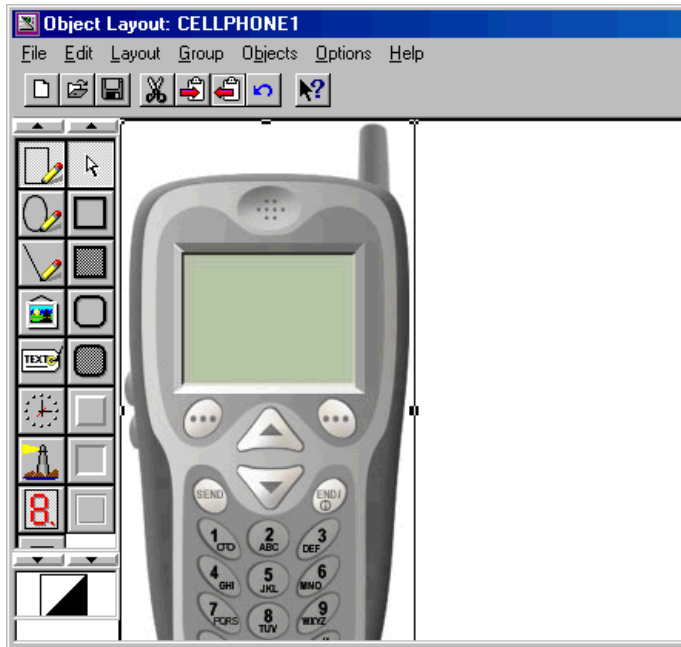


Adding the Cell Phone Bitmap Object

To add the cell phone bitmap object:

1. Click in the Object Layout to make it the active window.
2. Choose File|Import Bitmaps; the Import Bitmap File dialog box opens.

3. From the \\Rapid70\Tutorials\RapidPLUSXpress folder, select “PhoneImage.jpg,” then click .
4. Another dialog box opens with image management options; click . The cursor changes into crosshairs (+) for placing the bitmap.
5. Click in the layout frame; the cell phone bitmap appears.
6. Drag the bitmap object as far left and up as it will go. The Object Layout should look similar to this illustration:



Adding a Background Frame Object

You will add a frame object on top of the cell phone's display screen. It will provide the background for the states in transition charts and ensure that the size of the states will be uniform.

To add the background frame:

1. In the Object Palette, click the Frame class button at the top of the left column:



In the right column, the top frame object (below the pointer), Square Corners Frame, is selected. This is the object that you will use.

2. Place the crosshairs in the left corner of the cell phone's screen and click; a small frame is added on top of the screen.
3. Drag the lower-right resizing handle until the entire screen is enclosed. You may want to resize the frame object to include a bit more than the screen as shown here:



Attaching the Frame Object to the Mode Tree

In order for the frame object to appear as the background for states in the screen transition charts, you must attach it to the root mode, which bears the application name, CellPhone1.

To attach the frame object:

1. In the Mode Tree window, make sure that CellPhone1 is selected.
2. In the Object Layout window, right-click the frame object that you just added to open the popup menu and choose the option, Attach to Mode CellPhone1.

When the frame object is attached, RapidPLUS assigns it a name, "Frame1." The mode tree looks like this illustration:



Attached objects appear in red and in brackets. When the object is selected in the Object Layout, it is enclosed in a rectangle in the Mode Tree (as shown here)

3. Save your application.

Viewing the Background Frames in the Screen Transition Chart

Now that you have attached the background frame, you can create a screen transition chart to see how the frame adds a consistent background for all the modes.

❖ *NOTE: Microsoft® Visio® 2000 (or higher) must be installed to create the chart.*

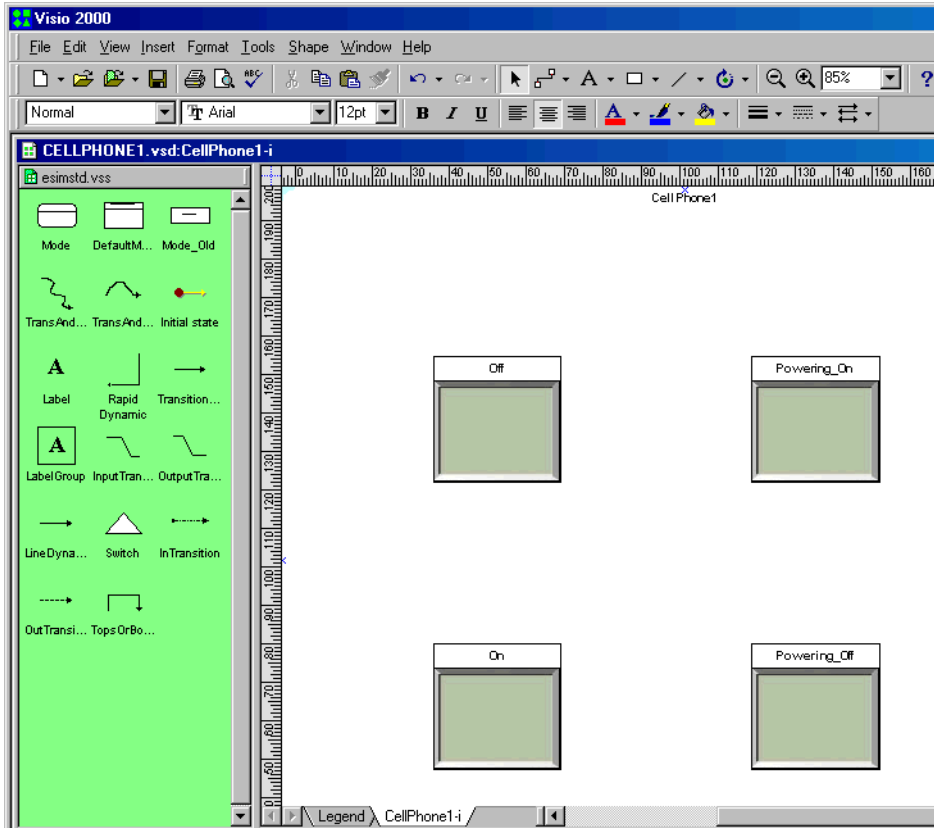
To create a screen transition chart:

1. Click in the State Chart to make it the active window.
2. Choose Tools|Export Transition Chart, or click the Transition Chart button; the chart is exported to Microsoft Visio.



Ctrl+Shift+T

Visio opens, first displaying a Legend page and then the chart. Your transition chart should look similar to this illustration:



Notice that the transition chart uses the transparent frame object as its shape for the states.

When you have finished viewing the transition chart, minimize the Visio window.

STAGE 3: DESIGNING THE SCREEN ELEMENTS

In this stage, you design how the system appears when each state is active. For example, when “Off” is active, the screen will be blank; when “Powering_On” is active, a graphic or message will be displayed on the screen; when “On” is active, a different graphic will be displayed; and so on.

The screen elements are added to the application in the Object Layout. They comprise both graphic and nongraphic objects. As mentioned in Stage 2, graphic objects provide a visual representation of the system’s user interface. Nongraphic objects (such as timers and .wav files) are not seen in the application, but maintain a very important role in the behavior of the system.

In this stage, you will add both graphic and nongraphic objects. After you have added the objects in the Object Layout, you will attach them to modes in the Mode Tree.

Designing the Screen for Off Mode

Because the screen is blank in Off mode, you do not add screen elements to it.

Designing the Screen for Powering On Mode

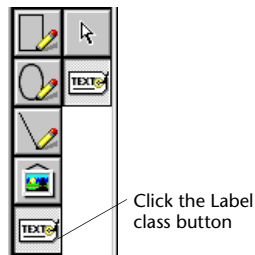
In this screen, “Welcome!” will be displayed. Also a single beep will sound.

Adding the “Welcome” Message using a Label Object

You will use a label object to hold the Welcome message.

To add the text:

1. Click in the Object Layout to make it the active window.
2. In the Object Palette, click the Label class button in the left column:



The right column displays the Label button, which is selected.

3. Place the crosshairs in the middle of the cell phone screen and click; the Label dialog box opens.
4. In the Text box, overtype “Text” with **Welcome!**.
5. Change the font by clicking **Font...** to open the Windows Font dialog box. The font we used is MS San Serif, Bold, 14 pt. Click **OK**.
6. Click **OK**; the text is added where you placed the crosshairs. Drag the label object to the appropriate position.

The screen should look similar to this illustration:



Naming the Label Object in the Parameter Pane

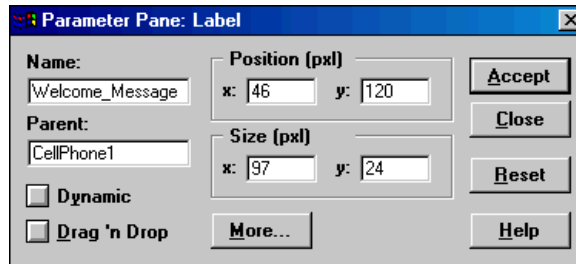
The Parameter Pane is a dialog box used by all graphic objects added to an application. It is used to set four parameters: object name, parent object, position, and size.

When you attached the frame object to the mode tree in Stage 2, Rapid assigned it a generic name, “Frame1.” The same would happen when you attach this label object—it would be assigned the name “Label1.”

Instead of using the default name, you will give a more meaningful name.

To name the label object:

1. Double-click the label object on the screen; the Parameter Pane opens.
2. Type **Welcome_Message** in the Name box:



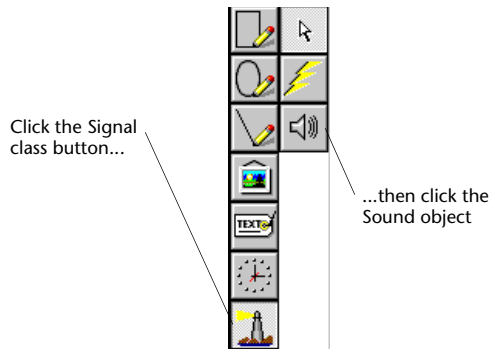
3. Click **Accept**, then click **Close**.

Adding the Beep Sound

The beep will be added using a nongraphic object, the Sound object. When a nongraphic object is added in the Object Palette, it is not placed directly on the Layout frame. Instead, a dialog box opens in which you can change the default name and access additional features via a second dialog box.

To add the beep:

1. In the Object Palette, click the Signal class button in the left column and the Sound object button in the right column:



2. The first dialog box opens. There is no need to change any parameters; however, to hear the beep, click **More...**; the second dialog box opens.
3. Click **Test**; the beep sounds. Click **OK** to close the dialog box.

- Click to close the first dialog box.

Look in the upper-left corner of the Layout frame; you should see the outline of a box that is selected:



The selected box encloses an icon, the Nongraphic objects icon, which appears when a nongraphic object has been added. Currently, it is hidden behind the cell phone bitmap object.

Help! I clicked somewhere else and can't see the hidden icon!

Don't worry. This situation is easy to fix.

1. Select the cell phone bitmap object.
2. Place the cursor in the upper-left corner where the icon is located.
3. Press the Tab key; the Nongraphic objects icon will be selected.

To see the hidden nongraphic objects icon:

1. Choose Layout|Order|Bring to Front. The icon appears in front of the bitmap object:



2. Double-click the icon; the Nongraphic Objects dialog box opens, listing all nongraphic objects that have been added to the application. Click .

Designing the Screen for On Mode

In this screen, the e-SIM logo will be displayed. You will add the logo in the same way you added the cell phone bitmap object. You will then make it transparent so that the color of the screen shows through it.

To add the e-SIM logo:

1. In the Object Layout, choose File|Import Bitmaps.
2. In the Import Bitmap File dialog box, select *logo.bmp*, then click .
3. In the next dialog box, click .
4. Place the crosshairs in the cell phone screen above “Welcome!” and click.
5. Reposition the logo as necessary. It should look similar to this illustration:



To make the logo bitmap transparent:

1. With the logo selected, choose Edit|Colors; the Colors Edit dialog box opens.
2. Select the Transparent check box and click .

Designing the Screen for Powering Off Mode



In this screen, “Good-bye” will be displayed.

To add the “Good-bye” text using a label object:



1. In the Object Palette, click the Label class button in the left column; the Label button is selected in the right column.
2. Place the crosshairs in the empty space next to the cell phone and click.
3. In the Label dialog box, overtype “Text” with **Good-bye**. The font you selected for the “Welcome!” will automatically be used. Click .

The text is added where you placed the crosshairs. For now, leave it where it is. When it is time to attach this label object to Powering_Off mode, you’ll position it correctly on the cell phone screen.

4. Double-click this object to open the Parameter Pane. Name the object **Good_Bye_Message** and click , then click .
5. Save the application.

STAGE 4: ATTACHING SCREEN ELEMENTS TO MODES

In this stage, you attach the graphic and nongraphic objects you added in the Object Layout to the appropriate modes in the Mode Tree.

Attaching Screen Elements to Powering_On Mode

You will attach the “Welcome!” message and beep to this mode.

To attach the screen elements:

1. Click in the Mode Tree to make it the active window.
2. Select Powering_On in the mode tree and right-click; the popup menu opens.
3. Choose Attach Object|Welcome_Message.
4. Right-click Powering_On again, and choose Attach Object|Sound1.

Attaching Screen Elements to On Mode

You will attach the e-SIM logo to this mode.

To attach the screen elements:

1. Select On in the mode tree and right-click.
2. Choose Attach Object|logo.

Attaching Screen Elements to Powering_Off Mode

You will attach the “Good-bye” message to this mode.

You may have noticed that when you moved from Powering_On mode, the Welcome label object disappeared from the Layout frame. Now that it is attached, you can only see it when Powering_On mode is selected.

Before you attach the Good-bye message, you will reposition it on the screen.

To attach the screen elements:

1. In the Object Layout, select the Good-bye message label object and drag it to its new position on the cell phone screen, approximately where the Welcome message was positioned. Don't deselect it.
2. In the Mode Tree, select Powering_Off and right-click.
3. Choose Attach Good_Bye_Message.

The mode tree looks like this illustration:



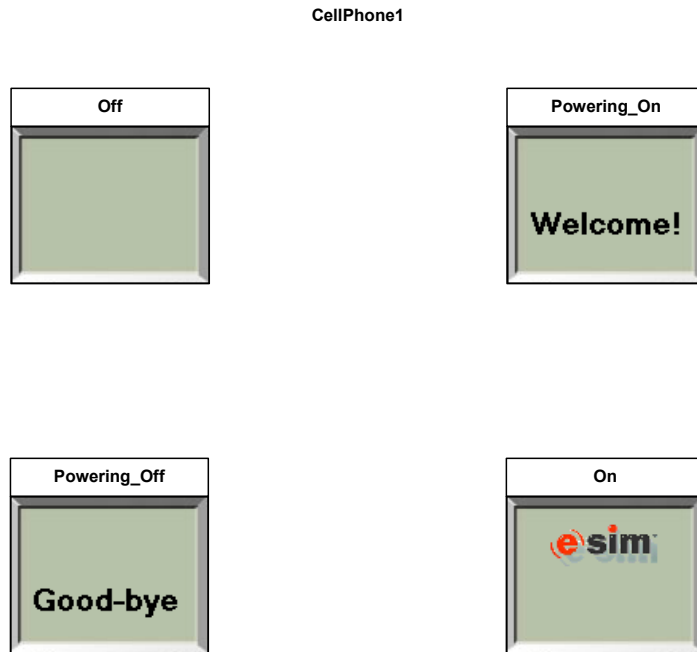
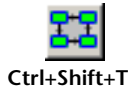
4. Save the application.

Viewing the Screens in the Screen Transition Chart

You can check how the screens look in the screen transition chart by re-exporting the application to Microsoft Visio.

To re-export the transition chart:

1. Click in the State Chart to make it the active window and select the CellPhone1 state.
2. Click the Transition Chart button; the chart is re-exported.
3. Maximize Visio. Your transition chart should look similar to this illustration:



By the way, if you want to see how the frame object adds consistency, temporarily remove it from the transition charts. To do so, right-click Frame1 in the mode tree and choose Deactivated, then regenerate the transition chart.

When you have finished viewing the transition chart, minimize the Visio window. Right-click Frame1 again and choose Deactivated so that the frame will once again appear in the screen transition charts.

STAGE 5: DEFINING TRANSITIONS

In this stage, you define transitions. A **transition** is a move from one mode to another that occurs when a specified trigger takes place. A **trigger** is an event of a specified RapidPLUS Xpress object. For example, the transition from Off mode to Powering_On mode will occur when an On/Off button is pressed. In screen transition charts, transitions appear as arrows going to and from the designated screens. Transitions are also important when the Rapid application is run, because the transitions are the triggers that cause the application to change states.

To define transitions, you add objects that will be used to trigger the transitions, then you make transitions using these objects' events.

Defining the Transition from CellPhone1 to Off Mode

In the Mode Tree, notice that there is an arrow pointing to Off mode. This arrow means that Off is the default child of its parent mode, CellPhone1.

Why You Don't Need to Define This Transition: a Look at Exclusive Modes

Take a look at the dialog box you used to add Off mode. To do so, select Off in the mode tree and choose Tree|Edit Mode. Notice that the Mode Type is Exclusive. An exclusive mode is a mode that cannot be active when a sibling mode is active. Exclusive modes are used for modes of the system that cannot run simultaneously. For example, when the cell phone is off, it cannot be on; when it is powering on, it cannot be powering off. (Be sure to close the dialog box when you have finished looking at it.)

When you add exclusive modes, the first child mode added becomes the default mode and is marked by an arrow. The default designation means that when the application is run in the Prototyper and a transition is made to the parent mode so that it becomes active, its default child becomes active as well. Therefore, there is no need to define a transition between the parent and its default exclusive child.

Defining the Transition from Off Mode to Powering_On Mode

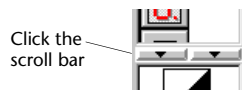
As mentioned earlier, the transition from Off mode to Powering_On mode will be triggered when an On/Off button is pressed. You will define this transition by adding a pushbutton object in the Object Layout and then using it to make the transition.

Adding the On/Off Pushbutton

The pushbutton object will appear on top of the End/Power key on the cell phone bitmap and will be made transparent so that the bitmap's End/Power key shows through the pushbutton object.

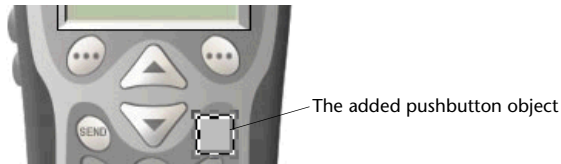
To add the pushbutton:

1. Click in the Object Layout to make it the active window.
2. In the Object Palette, use the scroll bar in the left column...



... to scroll down to the Pushbutton class button and click it. In the right column, the top pushbutton object, Flat Pushbutton, is selected. This is the object that you will use.

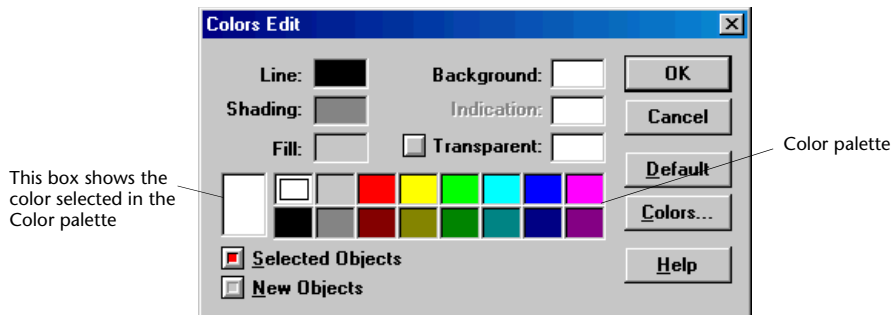
3. Place the crosshairs at the upper-left corner of the End/Power key and drag until the entire key is enclosed. When you release the mouse button, the pushbutton will hide the key.



To make the pushbutton transparent:

1. Choose Edit|Colors to open the Colors Edit dialog box.

Notice that the colors selected in the Line, Shading, Fill, and Background boxes are all different. Then notice that there is only one color box next to the Transparent check box:



- You must make the color elements the same color so they will be transparent. Next to the Color palette is a box that displays the selected color, white. Click the Line, Shading, and Fill color boxes to make them white.
- Select the Transparent check box and click . The transparent pushbutton object looks like the one in this illustration:



- Double-click this object to open the Parameter Pane. Name the object **End_Power_Key** and click , then click .

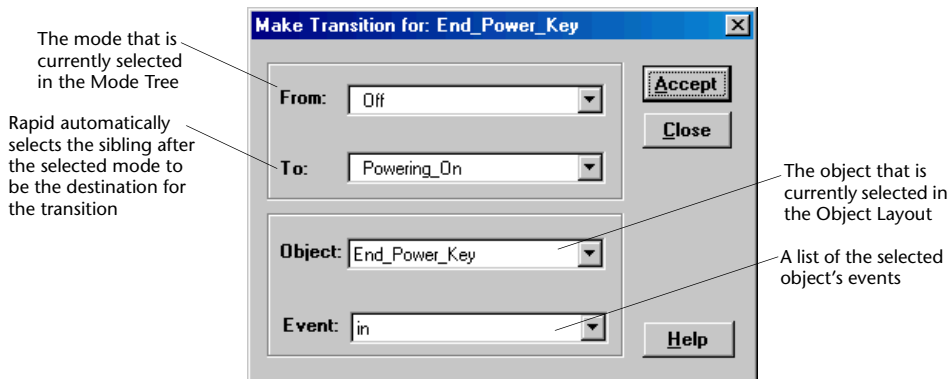
❖ *NOTE: Graphic objects, such as pushbuttons, that are used in transitions do not need to be attached to the mode tree.*

Making the Transition from Off Mode to Powering_On Mode

Transitions can be made from the Object Layout, Mode Tree, or State Chart. An efficient way to make a transition is to: (i) select the mode that is the source of the transition in the Mode Tree; then (ii) select the object that will trigger the transition in the Object Layout.

To make the transition:

- In the Mode Tree, select Off.
- In the Object Layout, select End_Power_Key (if not already selected), right-click and choose Make Transition; the Make Transition dialog box opens:



3. In the Event box, click the arrow. The pushbutton object has three events associated with it: in, out, and longPress. Select **longPress**.
4. Click **Accept** (and don't close the dialog box).
In the State Chart, notice that an arrow was drawn, which indicates the presence of a transition from Off to Powering_On.

Defining the Transition from On Mode to Powering_Off Mode

Because the End/Power key is used to power on *and* power off the cell phone, the transition from On to Powering_Off is similar to the one you just created.

To make the transition:

1. In the From box, click the arrow and select "On."
Notice that Powering_Off is now selected in the To box.
2. The selected object and event are correct; click **Accept**, then click **Close**.
In the State Chart, notice that another arrow was drawn, this time from On to Powering_Off. Click the arrowheads; transition information is presented in the Information pane below the chart.

Deleting incorrect transitions

You can delete incorrect transitions in the State Chart.

To delete a transition:

1. In the State Chart, select the transition arrow that you want to delete.
 2. Choose Logic|Delete Transition or press the Delete key.
-
-

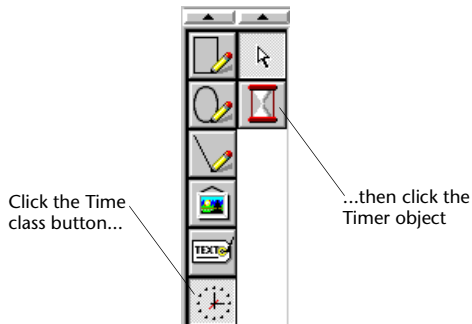
Defining the Transition from Powering_On Mode to On Mode

When a cell phone moves from Powering_On to On, no user action is involved; this transition is automatic. To define this transition, you add a timer nongraphic object in the Object Layout, attach it to Powering_On in the Mode Tree, and make the transition in the State Chart.

Adding the Timer Object

To add the timer:

1. Click in the Object Layout to make it the active window.
2. In the Object Palette, click the Time class button in the left column; click the Timer object in the right column:



3. The first dialog box opens; click .
4. In the second dialog box, change the initial value from 10 to 1.

This value means that the timer will count down 1 second, then trigger the transition.

5. Click twice to close both dialog boxes.

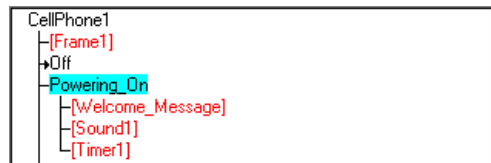
Attaching the Timer Object

Although you did not need to attach the pushbutton to the mode tree, you must attach the timer. Attaching the timer to a mode causes the timer to restart each time the mode is activated. After the timer is restarted, the transition is triggered.

To attach the timer:



1. In the Mode Tree, select “Powering_On.”
2. In the Object Layout, double-click the Nongraphic objects icon; the Nongraphic Objects dialog box opens.
3. Select “Timer Timer1” in the list.
4. Click **Attach To Mode** located at the bottom of the dialog box. Timer1 is added to the list of attachments for Powering_On:



Don't close the Nongraphic Objects dialog box.

Making the Transition from Powering_On Mode to On Mode

You will make this transition from the Nongraphic Objects dialog box.

To make the transition:

1. In the Nongraphic Objects dialog box, click **Make Transition...** to open the Make Transition dialog box.

All of the values in the dialog box are correct:

From: Powering_On

To: On

Object: Timer1

Event: tick (Timer1's only event)

2. Click **Accept** (and don't close the dialog box).

Defining the Transition from Powering_Off Mode to Off Mode

The timer tick event is also used to move from Powering_Off to Off.

To make the transition:

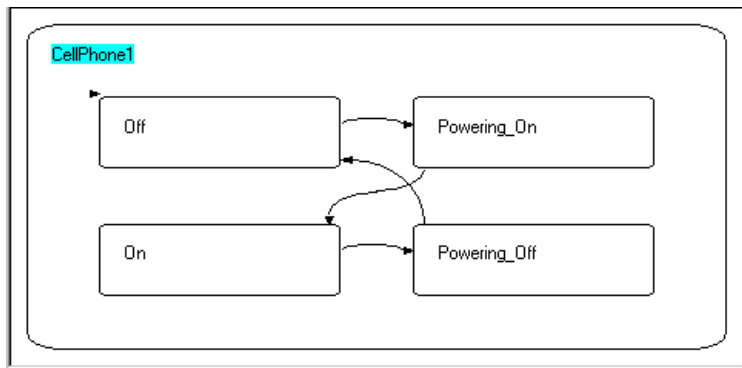
1. In the From box, click the arrow and select **Powering_Off**.
Rapid automatically updates the contents of the To box with the next sibling mode which is Off.
2. The selected event, tick, is correct so click **Accept**, then click **Close**.
3. Close the Nongraphic Objects dialog box.

To attach the timer:

1. In the Mode Tree, right-click Powering_Off.
2. From the popup menu, choose Attach Object|Timer1.
3. Save the application.

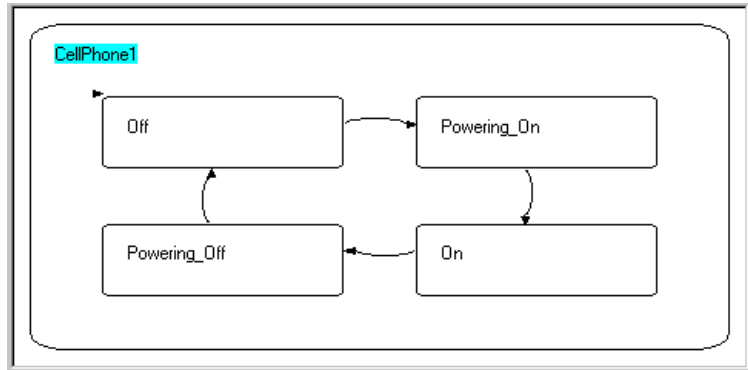
Designing the Charts

Take a look at the transitions in the State Chart. As you can see, two of the arrows cross each other. This is because the routing method used in Rapid to lay out states is left-to-right routing. The following illustration shows how left-to-right routing works:



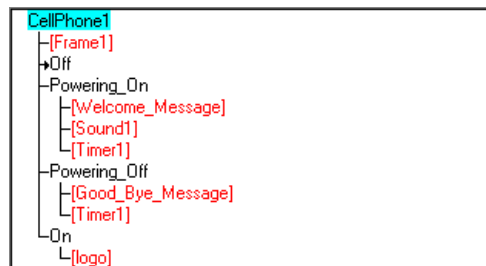
In Rapid simulations, left-to-right routing works fine; however, when this state chart is exported to Visio as a screen transition chart, the overlapping transition lines will remain and may make the chart confusing.

You can change the routing type to continuous by rearranging the states in the Mode Tree:



To rearrange the modes:

1. Click in the Mode Tree to make it the active window.
2. If Powering_Off is not already selected, then select it.
3. Drag it **above** On and release the mouse button. Powering_Off becomes the third sibling and On becomes the fourth:



4. Save the application.

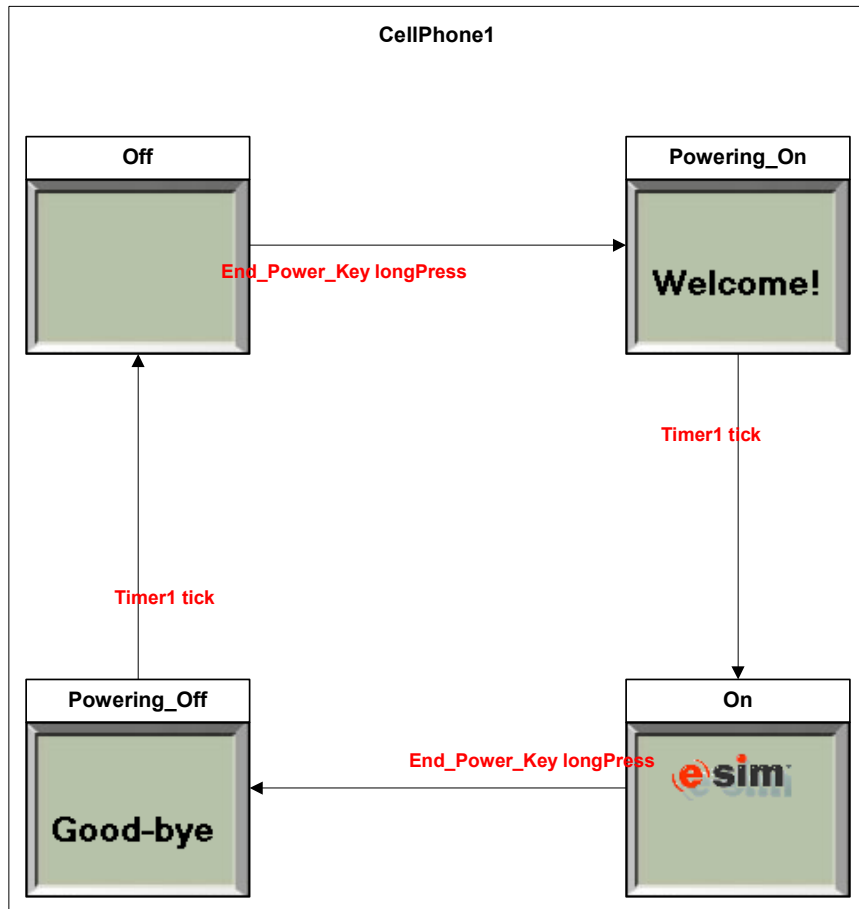
Viewing the Screens in the Screen Transition Chart

Now that you have added the transitions, you can export the chart again. The Mode Tree window is already active; you can export the chart from there.

To re-export the transition chart:

1. In the Mode Tree, choose File|Export Transition Chart; the chart is re-exported to Visio.

The final transition chart should look similar to this illustration:



RUNNING THE APPLICATION IN THE PROTOTYPER

Now that you have added the transitions, you can run the application in the Prototyper tool and see how Rapid moves from state to state.

To run the application:

1. Click in the Application Manager to make it the active window.
2. Choose Tools|Prototyper, or click the Prototyper button.



Ctrl+R

The Prototyper window opens *empty* because you only see the application when you run it.

3. In the Prototyper, click the Start (Play) button; the cell phone is displayed in its Off state:



While the Prototyper runs, the active modes are highlighted in the Mode Tree so you can check that each mode is being entered and exited. (This feature is called “Trace” and is located in the Prototyper’s Options

menu.) The modes being traced are highlighted in gray; the mode that is highlighted in cyan is the mode that was last selected in the Mode Tree.

4. Click-and-hold the End/Power key. The Welcome message appears and the beep is sounded. Then the Welcome message disappears and is replaced by the e-SIM logo.

Powering_On was briefly highlighted and now On mode is highlighted.

5. Click and hold the End/Power key again. The e-SIM logo disappears, the Good-bye message appears briefly, and then the display goes blank.

Powering_Off was briefly highlighted and now Off mode is highlighted.

If your application did not run as described above, open *Tutorial1_Completed.rpd* (located in \\Rapid70\Tutorials\RapidPLUS-Xpress) and compare it to your application.

6. Close the Prototyper window.

Congratulations!

You have completed the first of two tutorials about RapidPLUS Xpress. Although your screen transition chart is very basic and the application doesn't do much, you have learned the main features of RapidPLUS Xpress.

Now you can move on to designing a more complex system.

Before you go on...

SAVING SYSTEM SETTINGS

You can continue with the second tutorial or close RapidPLUS Xpress and perform the second tutorial later. Either way, now is a good time to save settings that you set during the tutorial (RapidPLUS opening in Xpress mode and font settings). The settings are saved to the *rapidx.ini* file.

To save system settings:

1. Click in the Application Manager to make it the active window.
2. Choose Options|Save Settings on Exit.

Tutorial 2—Expanding the Basic Application

In this tutorial, you will start with the cell phone application that you built in Chapter 1 and develop it into a more realistic application. Tutorial 2 is divided into five stages:

Stage 1: Defining the System's Modes

In this stage, you define many more modes for the cell phone including dialing a phone number, sending a call, and entering information in a phone book. These types of modes are grouped under a main Operation mode. Two modes that operate in the background, a battery monitor and a backlight, are grouped under a main System mode.

Stage 2: Defining the Battery Functionality

In this stage, you define a battery icon that always appears on the cell phone screen and indicates when the battery is full and when it is low. To achieve this effect, you add two graphics, one for a full battery and one for an empty battery, and attach them to the appropriate modes. You then create transitions from full battery mode to low battery mode.

Stage 3: Defining the Backlight Functionality

In this stage, you define a backlight that makes the screen appear illuminated whenever a key is pressed. The backlight goes off after five seconds of inactivity. To achieve this effect, you add a frame to simulate the illuminated screen and attach it to the appropriate mode. You then create transitions that cause the screen to illuminate.

Stage 4: Defining The Functionality for Ready Mode

In this stage, you define several screens: (i) the screen that appears when the cell phone is in an idle state awaiting user input; (ii) screens that show phone numbers being dialed; and (iii) screens that show phone numbers being cleared.

Stage 5: Defining the Functionality for Call_Control Mode

In this stage, you define several more screens: (i) the screen that appears when a phone number has been sent; (ii) the screen that appears for an incoming call; (iii) the screen that appears when the user is talking; and (iv) the screen that appears when a call is ended.

This tutorial does not fully develop the application, rather it touches on the main aspects of building a realistic RapidPLUS Xpress application. You will learn:

- To develop mode hierarchy efficiently.
- To choose the types of modes according to system functionality.
- To work with Rapid objects so that they provide optimal visual effects in screen transition charts.
- To develop transitions among modes that provide the necessary functionality.
- To customize screen transition charts in RapidPLUS Xpress.

The fully developed application, *CellPhone2_Completed.rpd*, is provided in the \\Rapid70\\Tutorials\\RapidPLUSXpress folder. After you have built the second tutorial, open the completed application to see how it was built.

Tutorial 2 should take 1½—2 hours.

Enjoy discovering RapidPLUS Xpress's capabilities!

STARTING THE TUTORIAL

This application is based on *CellPhone1.rpd*, the application you built in the previous tutorial.

To start:

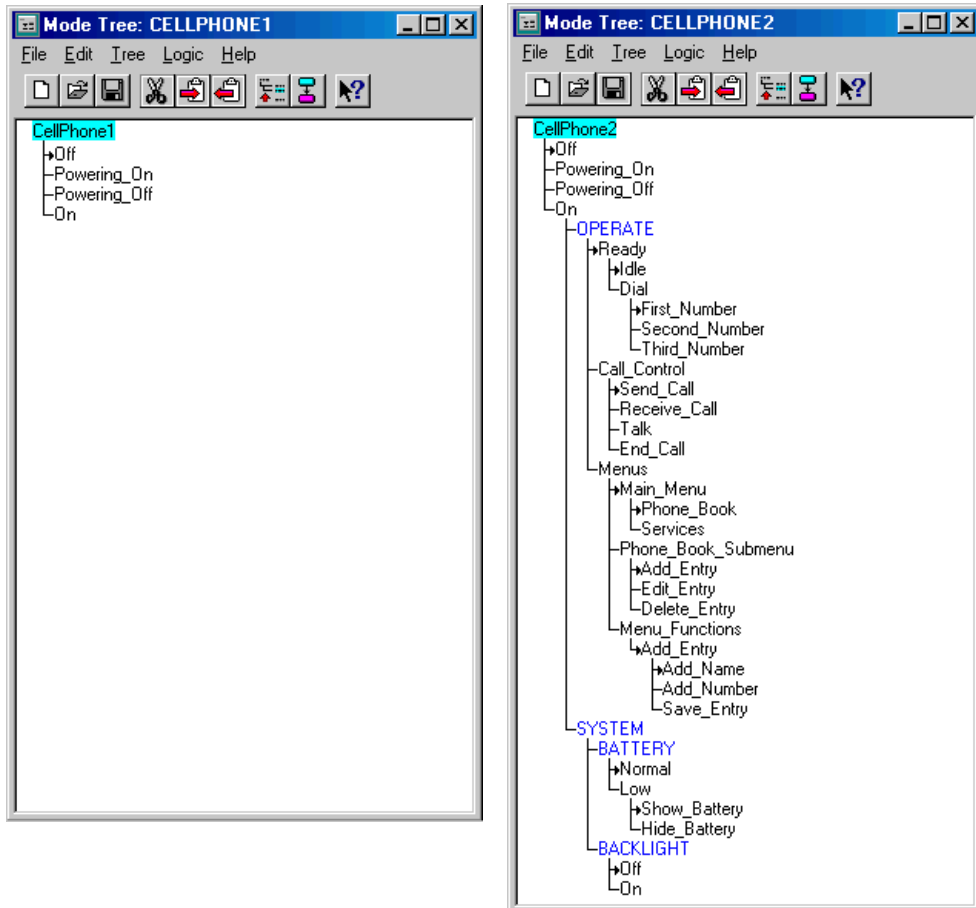
1. If Rapid is closed, launch it now. If you saved system settings in Tutorial 1, RapidPLUS will open in Xpress mode. If not, choose Options|Xpress mode.
2. In the Application Manager, choose File, then select “CellPhone1” from the list of most recently used files. The path for this file should be \\Rapid70\Tutorials\RapidPLUSXpress.
3. Choose File|Save As and save the application with the name **CellPhone2** in the same folder as CellPhone1.

STAGE 1: DEFINING THE SYSTEM’S MODES

A real cell phone may contain dozens or hundreds of modes, depending on the phone’s functionality and complexity. In the first stage of this tutorial, you will add 33 modes to the RapidPLUS Xpress application you built in Tutorial 1. That number may sound large, but it is not a difficult number to manage. The strength of the Mode Tree tool lies in its hierarchical nature, so you will be able to organize these modes logically and efficiently.

Before you begin, look on the next page at the completed trees for each application. The original application, CellPhone1, has four modes (its attachments are hidden); the next application, CellPhone2, has 37 modes.

Examine the hierarchy of both trees. The tree in CellPhone1 has only two generations: the parent (CellPhone1, the root mode) and its children. The tree in CellPhone2 has multiple generations. The root mode (CellPhone2) has children, grandchildren, etc.



Using Concurrent Modes

Before you build the second tree, you must understand the difference between exclusive and concurrent mode types. As you learned in Tutorial 1, exclusive modes are used for sibling modes that cannot be active simultaneously. For example, when the cell phone is on, it can't be off, and so on.

Sometimes there are sibling modes that can run simultaneously because they are functionally independent of each other. For example, the battery is being used while the cell phone is being operated; the backlight lights up when a key is pressed. These types are called *concurrent* modes. You determine the type when you add modes to the tree.

Adding Modes to the Tree

You will add modes logically, working down through the generations.

Adding Modes to On

On mode will have two concurrent child modes: Operate and System. Operate comprises all modes that require user input. System comprises modes that occur automatically.

To add modes to On:



Ctrl+W

1. In the Mode Tree, select “On” in the tree.
 2. Choose Tree|New Mode, or click the New Mode button to open the New Mode dialog box.
 3. In the Name box, type **OPERATE**. (We suggest using all caps for concurrent modes to differentiate them from exclusive modes.)
 4. Select the Concurrent mode type, then click .
- ❖ *NOTE: If you forget to select the Concurrent type, OPERATE is saved as an Exclusive type. Although you can edit a mode's name, you cannot edit its type. If you assign the wrong type, you must delete the mode and start again.*
5. In the Name box, type **SYSTEM** and click .

Notice that concurrent modes appear in blue.

Adding Modes to OPERATE

OPERATE mode will have three exclusive child modes: Ready, Call_Control, and Menus. Ready is the mode in which the cell phone waits for user input. Call_Control is the mode that manages call functionality. Menus is the mode that manages menu navigation and menu input.

To add modes to OPERATE:

1. Select “OPERATE” in the tree. Notice in the New Mode dialog box that the mode type switches back to Exclusive. Exclusive is the default type and the one you need for the next branch in the tree.
2. In the Name box, type **Ready** and click .
3. Type **Call_Control** and click .
4. Type **Menus** and click .

Adding Modes to Ready

Ready mode will have two exclusive children: Idle and Dial. Idle presents the screen that appears when the cell phone is awaiting user input. Idle does not have children. Dial comprises modes that display numbers as they are dialed.

To add modes to Ready:

1. Select “Ready” in the tree.
2. In the Name box, type **Idle** and click .
3. Type **Dial** and click .

Adding Modes to Dial

Dial mode will have three exclusive child modes: First_Number, Second_Number, and Third_Number. Each of these modes will present the screen as it looks when numbers are entered. None of them have children.

To add modes to Dial:

1. Select “Dial” in the tree.
2. In the Name box, type **First_Number** and click .
3. Type **Second_Number** and click .
4. Type **Third_Number** and click .

Adding Modes to Call_Control

By now you have added many modes so you no longer need explicit instructions. For Call_Control mode, add the following exclusive modes:

1. **Send_Call**: presents the screen that appears after the Send key has been pressed.
2. **Receive_Call**: presents the screen that appears when a call is incoming.
3. **Talk**: presents the screen that appears when a call has been answered.
4. **End_Call**: presents the screen that appears when a call is ended.

None of these modes have children.

Adding Modes to Menus

For the Menus branch, add the following modes, taking care to preserve the hierarchy.

To Menus, add three exclusive children:

1. **Main_Menu**: presents the cell phone's main menu options.
2. **Phone_Book_Submenu**: presents the phone book's menu options.
3. **Menu_Functions**: presents one menu function.

To Main_Menu, add two exclusive children:

1. **Phone_Book**
2. **Services**

Both of these modes present menu options. Usually a cell phone has many options in its main menu. To keep the tutorial short, only two options are presented. Neither of these modes has children.

To Phone_Book_Submenu, add three exclusive children:

1. **Add_Entry**
2. **Edit_Entry**
3. **Delete_Entry**

These modes present menu options for performing work on phone book entries. None of these modes have children.

To Menu_Functions, add one exclusive child:

1. **Add_Entry**

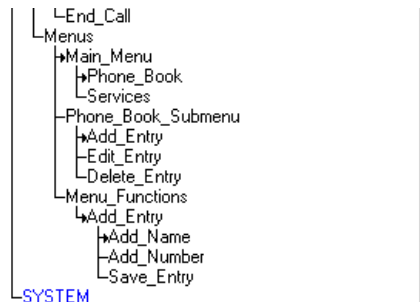
Usually a cell phone has many menu functions. To keep the tutorial short, only one menu function is presented. Add_Entry presents screens for entering data in phone book entries.

To `Add_Entry`, add three exclusive children:

1. `Add_Name`
2. `Add_Number`
3. `Save_Entry`

The first two modes present data being entered in a phone book entry. The third mode presents the screen that appears when the entry has been saved. None of these modes have children.

The completed Menus branch should look like this illustration:



Adding Modes to SYSTEM

Before you add modes to the `SYSTEM` branch, you can collapse the `OPERATE` branch to give more room in the tree and help you focus on the upcoming branches.

To collapse the `OPERATE` branch:

1. Select “`OPERATE`.”
2. Choose `Tree|Collapse`.

For the `SYSTEM` branch, add the following modes, taking care to preserve the hierarchy.

To `SYSTEM`, add two *concurrent* children:

1. **BATTERY**: comprises modes that display the battery icon.
2. **BACKLIGHT**: comprises modes that display the backlight.

To BATTERY, add two exclusive children:

1. **Normal**: presents the full battery icon. This mode has no children.
2. **Low**: comprises modes that present the low battery icon.

To Low, add two exclusive children:

1. **Show_Battery**: presents the low battery icon.
2. **Hide_Battery**: hides the low battery icon.

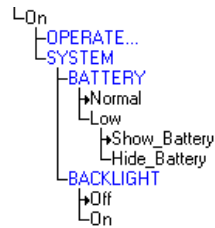
Using these modes will cause the low battery icon to appear to blink.
Neither of these modes has children.

To BACKLIGHT, add two exclusive children:

1. **Off**: presents the default screen which is not illuminated.
2. **On**: presents the screen illuminated.

Neither of these modes has children.

When you have finished building the tree, close the New Mode dialog box.
The completed SYSTEM branch should look like this illustration:

**Stage 1 is completed.**

Compare your tree with the illustration on p. 42 to ascertain that it is correct.
To see the entire tree, double-click OPERATE to expand it.

Be sure to save your application.

STAGE 2: DEFINING THE BATTERY FUNCTIONALITY

In this stage, you will define a battery icon that always appears on the cell phone screen—when the cell phone is on— and indicates whether the battery is full or low.

To achieve this effect, you add two graphics, one for a full battery and one for an empty battery. You will attach the full battery graphic to the mode BATTERY/Normal and the empty battery graphic to the mode BATTERY/Low/Show_Battery. You will create three transitions: one from Normal mode to Low mode and two within Low mode's subtree to make the empty graphic blink.

Adding the Battery Graphics

You can add both graphics in one Import file operation.

To add the battery graphics:

(Before you add the graphics, check in the Mode Tree that either On mode or OPERATE mode is selected.)

1. In the Object Layout, choose File|Import Bitmaps.
2. In the Import Bitmap File dialog box, select both “EmptyBattery.bmp” and “FullBattery.bmp” located in the \\Rapid70\Tutorials\RapidPLUS-Xpress folder and click .
3. In the “Import Bitmaps from: 2 files” dialog box, click .
4. Place the crosshairs in the screen's upper-right corner (above the e-SIM logo) and click; both graphics are added to the layout frame.
5. Double-click the top bitmap to open the Parameter Pane. Notice that the name of the bitmap file has become the name of this new bitmap object. Close the Parameter Pane.
6. Shift-click the other bitmap so that both battery graphics are selected. If you have trouble selecting both bitmaps, you can use the Edit|Select from List command to select EmptyBattery and FullBattery from a list of application objects.

- Choose Layout|Align|Top and then choose Layout|Align|Right to place them in the identical position. If they are not exactly where you want them to appear, nudge them into place using the Shift+arrow keys.



Keep the battery graphics selected.

- Now, make both graphics transparent. To do so, choose Edit|Colors to open the Colors dialog box, select the Transparent check box to make the transparent color white, and click .
- Click outside the cell phone to deselect both bitmaps.

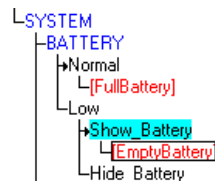
Attaching the Battery Graphics to the Battery Modes

You will attach FullBattery to the BATTERY/Normal mode and EmptyBattery to the BATTERY/Low/Show_Battery mode. There is no need to attach any bitmaps to the Hide_Battery mode because it will be blank to simulate the blink effect with Show_Battery mode.

To attach the graphics to the modes:

- In the Mode Tree, right-click BATTERY/Normal and choose Attach Object|Bitmap|FullBattery.
- Right-click BATTERY/Low/Show_Battery and choose Attach Object|Bitmap|EmptyBattery.

The tree should look like this illustration:



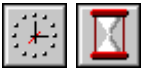
Defining Transitions for the Battery

When the cell phone is powered on, the battery starts running. As time passes, the battery empties and eventually gets low. You will use a timer object to cause the battery to empty.

Defining the Transition from Normal Mode to Low Mode

You will define this transition using a timer object.

To add the timer:



1. In the Object Layout, click the Time class button in the left column of the Object Palette and the Timer object in the right column.
2. In the dialog box, name the object **One_Min_Tmr**, then click .
3. In the second dialog box, change the initial value from 10 sec to **1 min**.
4. Click twice to close both dialog boxes.

Tips on naming objects used in transitions

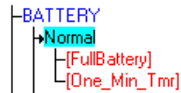
Look at the suggested name for the new timer object: `One_Min_Tmr`. It could have been `One_Minute_Timer`, `Battery_Timer`, or even just `Timer2`. Here are points to keep in mind when you name transition objects.

- Transition object names appear in transition charts together with their events. For example, “`One_Min_Tmr tick`.” The longer the name, the more space it takes in the chart, which can lead to collisions with other transitions. Therefore, you should keep names as short as possible.
 - Transition names should be descriptive. `Timer2` is okay, but when an application has numerous timers, it is hard to remember their attributes. `Battery_Timer` is also a descriptive name, but it doesn’t mention the time setting. If the time setting appears in the object’s name, it is easier to reuse the object.
 - Why not “1-min timer”? Numbers cannot be used as the first character in an object name. Dashes and spaces also cannot be used and will be substituted with an underscore (`_`).
 - You could also use the name `OneMnTmr`; fewer characters, yet descriptive. Just make sure that readers of your transition charts will understand your naming conventions.
-
-

To attach the timer:

1. In the Mode Tree, right-click BATTERY/Normal and choose Attach Object|Time|One_Min_Tmr.

Attaching the timer will cause it to restart each time Normal mode is activated. Normal mode is completed:

**To make the transition from Normal mode to Low mode:**

1. Right-click BATTERY/Normal and choose Make Transition.
2. In the Make Transition dialog box, select:
To: **Low**
Object: **One_Min_Tmr**
Event: **tick**
3. Click , then click .


Defining Transitions that Makes the Empty Battery Icon Blink




You can make the empty battery icon blink by using another timer object that transitions back and forth between Show_Battery and Hide_Battery modes.

**To add another timer object:**

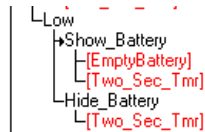
1. Double-click the Nongraphics icon to open the Nongraphic Objects dialog box.
2. In the list, select "Timer1," then click .
3. In the Duplicate dialog box, rename the object **Two_Sec_Tmr** and click .
4. The Two_Sec_Tmr timer object is selected in the list. Click and change the initial value from 1 sec to 2 sec.
5. Click . The new timer object is still selected.
Keep the Nongraphic Objects dialog box open.

To make the transitions:

1. Make the first transition: in the Nongraphic Objects dialog box, click .
2. In the Make Transition dialog box, select:
From: **Show_Battery**
To: **Hide_Battery**

The selected object and event, `Two_Sec_Tmr` and tick, are entered by default.
3. Click  (and don't close the dialog box).
4. Make the second transition: in the Make Transition dialog box, select:
From: **Hide_Battery**
To: **Show_Battery**
5. Click , then click .
6. Close the Nongraphic Objects dialog box.
7. In the Mode Tree, attach `Two_Sec_Tmr` to both `Show_Battery` and `Hide_Battery` modes.

The Low mode branch is complete:

**Testing the Application in the Prototyper**

Now that you have added the battery functionality, test it before you build the transition chart.



Ctrl+R

To run the application:

1. In the Application Manager, choose Tools|Prototyper, or click the Prototyper button.
2. In the Prototyper, click the Start (Play) button.

3. Power on the cell phone with a long press on the End/Power key. Notice that the full battery icon is displayed:



4. Wait one minute for the empty battery icon to start blinking.
5. Power off the cell phone, then close the Prototyper window.

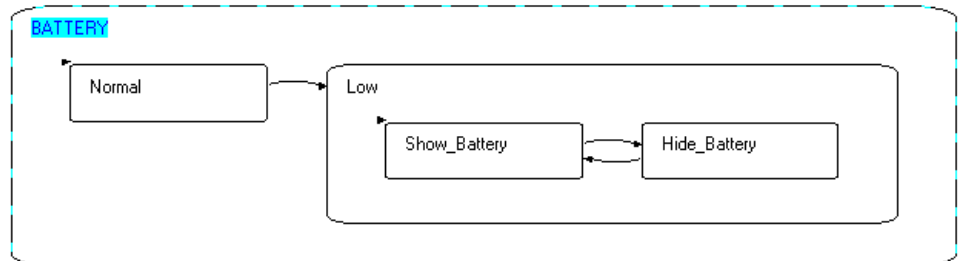
Viewing the State Chart



Ctrl+F

1. In the Mode Tree, click the Battery mode.
2. Click in the State Chart and look at the chart. If you can't see all of the modes, you can make the chart fit in the space by choosing View|Fit, or clicking the Fit button.

The chart should look similar to the following illustration. If you see a different view, double-click BATTERY in the State Chart or try the various View options.

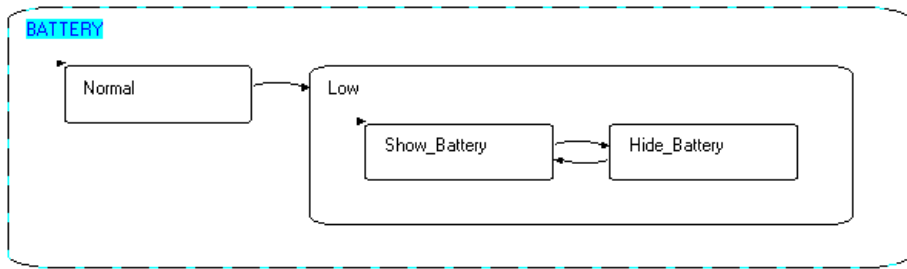


Viewing the Transition Charts

You will export transition charts for three states: Battery, Normal, and Low. Use the following instructions for each chart.

To export the charts:

1. In the State Chart, double-click the state for which you want to export a transition chart so that it appears as the top state. For example, in the following state chart, Battery is the top state; therefore, the exported transition chart will show Battery and its two children.

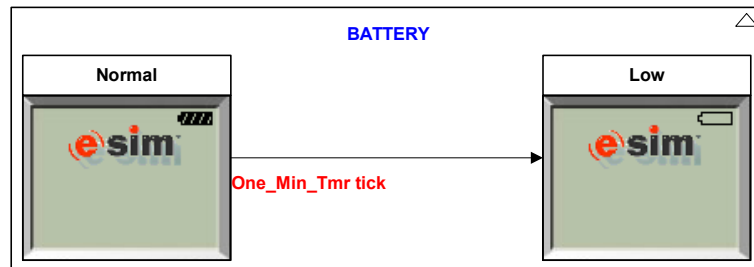


Ctrl+Shift+T

2. Choose Tools|Export Transition Chart, or click the Transition Chart button; Visio opens displaying the chart.

Transition Chart for Battery state

The transition chart should look similar to the following illustration:



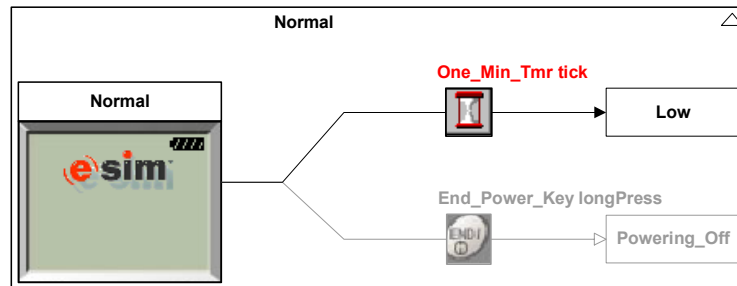
In Normal state, the full battery is displayed. In Low state, the empty battery is displayed. Low/Show_Battery is the default exclusive child of Low; therefore, it is active and its screen appears in the chart.

A one-minute timer causes the transition from Normal to Low.

Transition Chart for Battery/Normal State

Normal is a state with *no* child states, and its chart is referred to as a **single-state screen transition chart**. This type of chart is different from a screen transition chart, which is the chart type for a state that *has* child states.

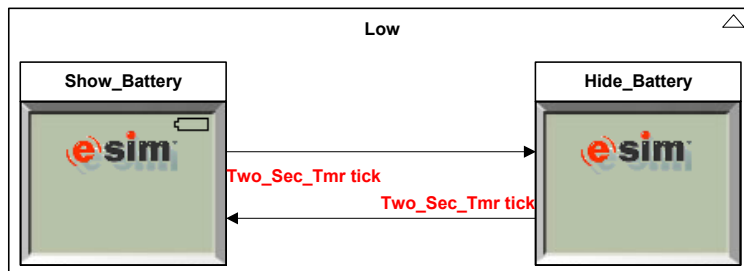
A single-state screen transition chart is “state-centric,” only showing transitions to and from the selected state. The transition chart for Normal state should look similar to the following illustration:



The “grayed” transition with an unfilled arrow indicates a transition from an ancestor mode, in this case from its great-grandparent, On mode.

Transition Chart for Battery/Low State

The transition chart should look similar to the following illustration:



In Show_Battery state, the empty battery is displayed; in Hide_Battery state, no battery graphic is displayed. The two-second timer that transitions between the two states causes the blink effect when the application is run in the Prototyper.

Stage 2 is completed. Be sure to save your application.

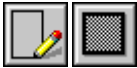
STAGE 3: DEFINING THE BACKLIGHT FUNCTIONALITY

In this stage, you will define a screen that will give the appearance of a backlight. The backlight illuminates each time a key is pressed and goes off after five seconds of inactivity.

To achieve the backlight effect, you will attach a frame object to the Backlight/On mode that has a lighter color than the original screen. You will create transitions to and from the backlight's Off and On modes to turn the backlight off and on.

Adding the “Illuminated” Frame Object

To add the frame object:



1. In the Object Layout, click the Frame class button in the left column of the Object Palette and the Square Corners Filled Frame in the right column.
2. Add a frame that is smaller than the size of the screen so that you can see the logo. You will resize it later.



The battery icon may or may not show depending on the mode that is selected in the Mode Tree.


3. Make the frame's color lighter than the screen color. To do so, open the Colors Edit dialog box (Edit|Colors) and click .
4. In the Colors dialog box, select a color. We used the RGB values of 212, 247, 96.
5. Click , then click . The color appears in the Selected Color box next to the palette.

- Click the Line and Fill color boxes to make them this color, then click



- Because the frame is filled, if you resize it to fit the screen, it will cover all the graphics; therefore you must send it behind the e-SIM logo.

With the filled frame selected, choose Layout|Order|"Send to Back of"; the Order Objects dialog box opens.

- Alt-click the logo to add it to the dialog box, then click .

- Resize the frame to fill the screen:



Attaching the Illuminated Frame to Backlight/On Mode

To attach the filled frame object:

- In the Mode Tree, right-click Backlight/On mode.
- Choose Attach Object|Frame|SquareCornersFilledFrame. (When it is attached, it will be renamed Frame2.)

The Backlight branch looks like this illustration:



Defining the Transitions for the Backlight

When the cell phone is in idle state, the backlight is off. When any key is pressed, the backlight illuminates. If no other key is pressed, the backlight goes off after five seconds. In this step, you will create transitions from all the keys so that the backlight will illuminate when any of them is pressed.

Adding the Pushbuttons

You will copy the End_Power_Key pushbutton object to create the other pushbuttons because you have already defined its size and transparency (in Tutorial 1).



Ctrl+C



Ctrl+V

To add the Send pushbutton:

1. In the Object Layout, select “End_Power_Key” and copy it. To copy it, choose Edit|Copy, or click the Copy button.
2. Choose Edit|Paste Object; the Paste Conflicts: Duplicate Elements dialog box opens.
3. Click **Keep both**; the dialog box closes and the cursor changes to crosshairs.
4. Place the crosshairs at the upper-left corner of the Send key and click. If the transparent pushbutton doesn’t cover the Send key, reposition it until it does.
5. Double-click the new pushbutton object to open the Parameter Pane and rename it **Send_Key**. Click **Accept** (leave the Parameter Pane open).

To add the soft key pushbuttons:

1. With Send_Key selected, Shift-click End_Power_Key to select it too.
2. Copy and paste these objects; the Paste Conflicts: Duplicate Elements dialog box opens.
3. Click **Keep both** twice.
4. Place the crosshairs at the upper-left corner of the left soft key and click. If the new pushbuttons don’t cover the soft keys correctly, reposition them:



5. Deselect both pushbuttons, then select the pushbutton over the left soft key.

6. In the Parameter Pane, rename it **Left_Soft_Key** and click .
7. Select the pushbutton over the right soft key and rename it **Right_Soft_Key**. Be sure to click .

To add a general pushbutton for the numeric keys:

Now you will add one general pushbutton for all the numeric keys. If these keys aren't visible on the layout frame, scroll down until they are visible.

1. Copy and paste one of the transparent pushbuttons.
2. In the Paste Conflicts dialog box, click .
3. Place the crosshairs over the number 1 key and click. Resize this new pushbutton until the entire keypad is enclosed:



4. Rename this new pushbutton **Numeric_Keys** and click , then click .






❖ *NOTE: You can create pushbuttons that use different bitmaps for the In and Out positions. For instructions, refer to the online Help file in the "Adding realism to pushbutton and switch positions" topic.*

Defining the Transitions from Backlight/Off to Backlight/On

You will make five transitions, one transition for each pushbutton object. Each time that a pushbutton is pressed in, the backlight will go on.

To define the transitions:

1. In the Mode Tree, right-click Backlight/Off and choose Make Transition. In the Make Transition dialog box, From: Off and To: On are correct.
2. Make the first transition; select:
Object: **End_Power_Key**
Event: **in**
and click .

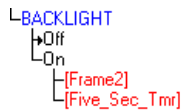
3. Make the second transition; select:
Object: **Send_Key**
Event: **in**
and click .
4. Make the third transition; select:
Object: **Left_Soft_Key**
Event: **in**
and click .
5. Make the fourth transition; select:
Object: **Right_Soft_Key**
Event: **in**
and click .
6. Make the fifth transition; select:
Object: **Numeric_Keys**
Event: **in**
and click , then click .

Defining the Transition from Backlight/On to Backlight/Off

You will define this transition using a timer object so that the backlight will go off after five seconds of inactivity.

To define the transition:

1. In the Object Layout, add a timer object. Name it **Five_Sec_Tmr** and set its initial value to **5 sec**.
2. Attach **Five_Sec_Tmr** to **Backlight/On** mode:



3. Make the following transition:
From: **On**
To: **Off**
Object: **Five_Sec_Timer**
Event: **tick**

Testing the Application in the Prototyper

Now that you have added the backlight functionality, test it before you build the screen transition chart.



To run the application:

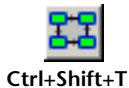
1. Open the Prototyper and click the Start button.
2. Power on the cell phone. Notice that the screen's default color is displayed.
3. Click any of the keys on the cell phone. The screen's color immediately changes to the lighter backlight color. After five seconds, the screen displays the default color.
4. Click several keys to see how the backlight "switches" on and off.

Did you notice that the timer starts counting when the first key is pressed in and turns off after five seconds, even though you pressed more keys?

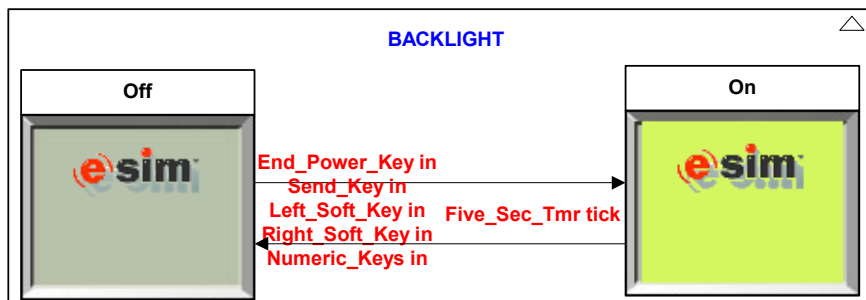
This functionality is fine for the screen transition charts; however, if you want the timer to restart each time you press a numeric key, you can add a transition from On mode to itself that is triggered each time a numeric key is pressed.

5. Power off the cell phone, then close the Prototyper window.

Viewing the Screen Transition Chart



Export the transition chart for Backlight. The chart should look similar to this illustration:



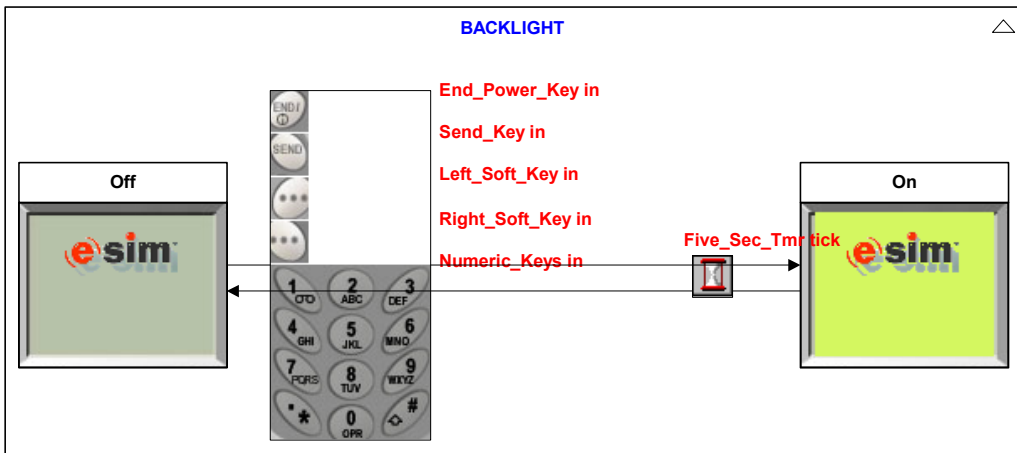
Modifying Options for the Screen Transition Chart

You can modify the screen transition chart's default settings to display the transitions as images.

To modify chart options:

1. In the State Chart, choose Tools|Options; the Chart Options dialog box opens.
2. Select the Screen Transition tab.
3. In the Information Shown as Images box, select Transition Triggers. This option will display images of the pushbuttons.
4. Click to close the dialog box.
5. Save this setting by using the Save Settings Now command from the Options menu in the Application Manager.

Export the transition chart for Backlight state again. The chart should look similar to this illustration:



Stage 3 is completed. Be sure to save your application.

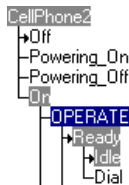
STAGE 4: DEFINING THE FUNCTIONALITY FOR READY MODE

In this stage, you define the functionality for Ready mode. Ready mode itself does not have attachments or transitions; it simply acts as an organizational mode in the OPERATE branch of the tree.

Ready mode has two child modes:

- Idle: presents the screen that appears when the cell phone is awaiting user input.
- Dial: comprises modes that display numbers as they are dialed.

Idle mode will always be the first screen that the user sees after powering on the cell phone. Why? Take a look at the following illustration. It shows the tree—with the attachments hidden—as the application is running in the Prototyper. A transition was just made from Powering_On to On:



When On mode becomes active, its child modes, OPERATE and SYSTEM, become active. Because OPERATE is active, its default child, Ready, becomes active. And when Ready is active, its default child, Idle, becomes active.

Designing the Screen for Idle Mode

In Idle mode, the screen will display the logo, the time, and the Menu option for opening the Main menu:



Moving the e-SIM Logo

Currently, the e-SIM logo is attached to On mode, which means that the logo will appear in every screen in the On branch. Because the logo only needs to appear when the application is idling, you should move it to Idle mode.

❖ *NOTE: The reason that you wouldn't move the logo to Ready mode is because doing so would cause the logo to remain on screen when numbers are being dialed.*

To move the logo:

1. In the Mode Tree, select the attachment “logo” under On mode.
2. Drag it to Idle mode, making sure that Idle is highlighted before you release the mouse button.

Defining a Time Display

In Idle mode, the cell phone will display the current time.

To add a time display:



1. In the Object Layout, click the Label class button in the left column of the Object Palette; the Label button is already selected in the right column.
2. Place the crosshairs under the e-SIM logo and click.
3. In the Text box, overwrite “Text” with **10 : 15**.
4. In the Font dialog box, change the font to Arial, bold, 10 pt.
5. Click twice.

- Double-click the label object to open the Parameter Pane and name the label object **Time_Label**. Click . Keep the Parameter Pane open.
- Reposition the time label object as necessary.

To attach the time label:

- In the Mode Tree, right-click Idle and choose Attach Time_Label.

Defining a Menu Option

In Idle mode, the cell phone will display the option, Menu.



To add the menu option:

- In the Object Layout, the Label class button should still be selected in the left column of the Object Palette. Click the Label button in the right column.

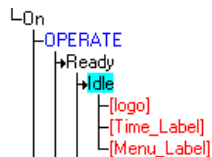
(Why wasn't the Label button still selected? Because an object button is deselected after the object has been added to the layout frame.)

- Place the crosshairs on the screen above the left soft key and click.
- In the Text box, overtype "Text" with **Menu** and click .
- In the Parameter Pane, name the label object **Menu_Label** and click .
- Reposition the menu label object as necessary.

To attach the menu label:

- In the Mode Tree, right-click Idle and choose Attach Menu_Label.

The Idle branch is complete:



You have finished designing the screen for Idle mode.


Designing the Screens for Dial Mode

Dial has three children: `First_Number`, `Second_Number`, and `Third_Number`. Each of these child modes will present a screen that displays numbers as they are dialed. Dial mode will present a Clear option for clearing numbers entered on the screen.

Adding a Clear Option

All cell phones have a Clear option that enables the user to either clear the last number entered or clear the entire phone number. The Clear option should be available from all of Dial mode's child modes; therefore, you will add the clear label to Dial.

To add and attach the Clear label:

1. Add another label object, positioning it above the right soft key.
2. In the Text box, add the text **Clear**.
3. In the Parameter Pane, name the object **Clear_Label**.
4. Align the Clear label with the Menu label. To do so, select `Menu_Label` and copy its y-position in the Parameter Pane. Select `Clear_Label` and paste the copied value into its y-position.
5. Click .
6. In the Mode Tree, attach `Clear_Label` to Dial mode.

In the Object Layout, notice that the logo, time, and menu option do not appear. They are part of Idle mode and are not needed when a phone number is being dialed:



Designing the Screens for Dialed Numbers

In Dial mode, you will simulate the dialing of numbers. The actual numbers dialed are not important in the screen transition charts; the appearance and placement of the dialed numbers are important.

In each child mode, you will add another number to the display. The numbers are displayed in label objects.

To add the first number label:

1. In the Object Layout, click the Label button in the right column of the Object Palette, then place the crosshairs on the screen above the Clear label and click.
2. In the Text box, overwrite “Text” with **4**.
3. Click to open the Font dialog box, then change the font size to **14** and click . Click again.
4. In the Parameter Pane, name the object **First_Num_Label** and click .
5. Reposition the label object so that the number, 4, appears over the “r” in Clear:



6. Attach First_Num_Label to First_Number mode.

To add the second number label:

1. Select “First_Num_Label” on the layout frame, then copy and paste it.
2. In the Paste Conflicts dialog box, click .

- Place the crosshairs on the layout frame and click:

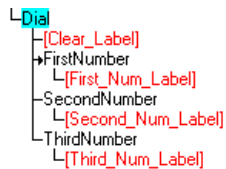


- In the Parameter Pane, rename the object **Second_Num_Label** and click .
- Click . Type **45** and click .
- Reposition the object so that it is aligned with First_Num_Label. To do so, select both of the number labels. Choose Layout|Align|Right, then choose Layout|Align|Bottom.
- Attach Second_Num_Label to Second_Number mode.

To add the third number label:

- Repeat the procedure that you just performed, naming the object **Third_Num_Label** and assigning the text **456**.
- Align Third_Num_Label with Second_Num_Label.
- In the Mode Tree, attach Third_Num_Label to Third_Number mode.

The Dial branch is complete:



Defining Transitions for Ready Mode

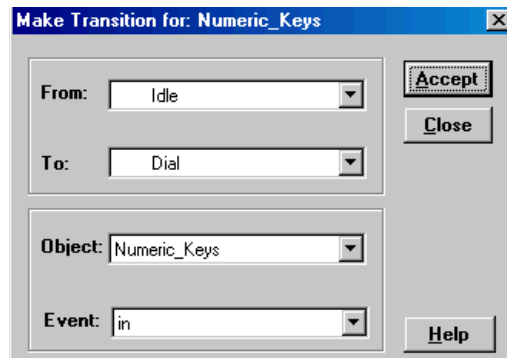
In Stage 3, you added the pushbutton objects needed to make the transitions in this stage; you do not have to add any more transition objects.

Defining the Dialing Transitions

Pressing numeric keys repeatedly will cause transitions from Idle to Dial/First_Number to Second_Number to Third_Number. No matter which key is pressed, the display will be the same: 4→45→456.

To make the transition from Idle to Dial/First_Number:

1. In the Mode Tree, select “Idle.”
2. In the Object Layout, right-click Numeric_Keys (the numeric keypad object) and choose Make Transition:



All of the values are correct so click **Accept**. Keep the dialog box open.

To make the transition from First_Number to Second_Number:

1. In the dialog box, click the arrow in the From box and select **First_Number**. The To value changes to Second_Number and the object and event remain the same.
2. Click **Accept**.

To make the transition from Second_Number to Third_Number:

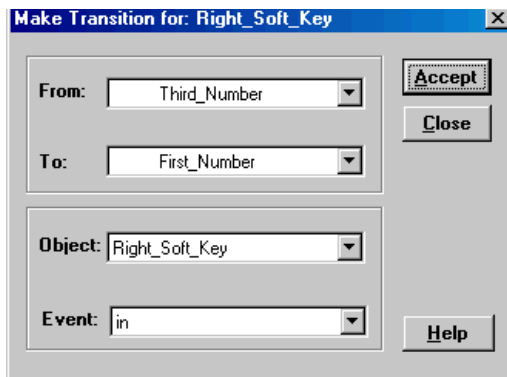
1. Click the arrow in the From box and select **Second_Number**. The To value changes to Third_Number and the object and event remain the same.
2. Click **Accept**, then click **Close**.

Defining the Clearing Transitions

Pressing the right soft key (Clear) repeatedly will cause transitions from Third_Number to Second_Number to First_Number to Idle. A long press on the right soft key will cause a transition from any of Dial's child modes all the way back to Idle.

To make the transition from Third_Number back to Second_Number:

1. In the Mode Tree, select "Third_Number."
2. In the Object Layout, right-click Right_Soft_Key pushbutton and choose Make Transition:



3. Change the To value to **Second_Number** and click **Accept**.



To make the transition from Second_Number back to First_Number:

1. Click the arrow in the From box and select **Second_Number**.
2. Change the To value to **First_Number** and click **Accept**.

To make the transition from First_Number back to Idle:

1. Click the arrow in the From box and select **First_Number**.
2. Change the To value to **Idle** and click **Accept**, then click **Close**.

To make the transition from Dial's child modes back to Idle:

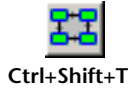
1. In the Mode Tree, right-click "Dial" and choose Make Transition.
The values:
From: Dial
To: Idle
Object: Right_Soft_Key
are correct.
2. Change the Event value to **longPress**.
3. Click , then click .

Testing the Application in the Prototyper

Now that you have added the Idle and Dial functionalities, test the application before you build the screen transition chart.

**To run the application:**

1. Open the Prototyper and click the Start button.
2. Power on the cell phone.
3. Click any of the number keys on the keypad.
One by one, the three numbers are added to the display.
4. Use the Clear soft key to test the Clearing functionality.
5. Notice the functioning of the battery icon and backlight.
6. When you have finished testing the application, power off the cell phone and close the Prototyper.



Viewing the Screen Transition Charts

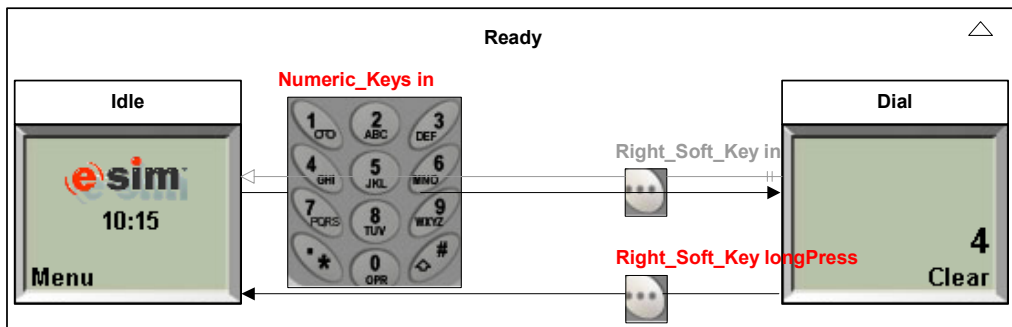
Export the transition charts for Ready and Dial. The charts should look similar to the next two illustrations:

Screen Transition Chart for Ready State

In Idle state, the idle screen is displayed. Pressing any numeric key transitions to Dial state, in which Dial and its default child, First_Number, are active.

Notice the transitions from Dial state back to Idle state. The transition arrow named “Right_Soft_Key longPress” appears as a solid black line and arrow; this arrow type signifies that the source for the transition is Dial state.

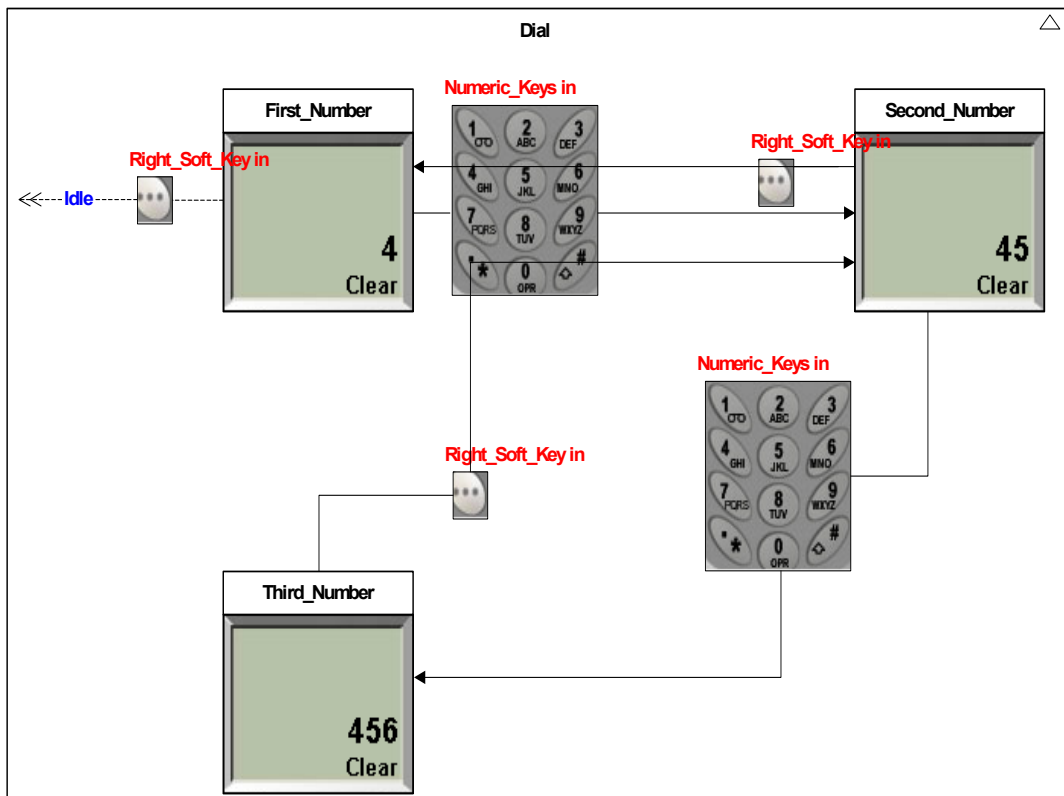
The other transition arrow, “Right_Soft_Key in,” appears as a gray line with an unfilled arrow and two small lines (⊥); this arrow type signifies that the source for the transition is one of Dial state’s children, in this case, First_Number.



Transition Chart for Dial State

This chart illustrates the transitions among Dial's children. Pressing in any numeric key transitions from First_Number to Second_Number to Third_Number. Pressing in the right soft key (Clear) transitions back through the numbers.

The dashed arrow that points back to Idle state indicates that there is a transition to a state that is not a child of Dial.



Stage 4 is completed. Be sure to save your application.

STAGE 5: DEFINING THE FUNCTIONALITY FOR CALL_CONTROL

In this stage, you define the functionality for Call_Control mode. Call_Control itself does not have attachments or transitions; like the Ready branch, it acts as an organizational mode in the OPERATE branch of the tree.

Call_Control has four child modes: Send_Call, Receive_Call, Talk, and End_Call. Send_Call presents the screen that appears after the Send key has been pressed. Receive_Call presents the screen that appears when a call is incoming. Talk presents the screen that appears when a call has been answered. End_Call presents the screen that appears when a call is ended.

Defining the Screen for Send_Call Mode

In Send_Call mode, the screen will display the message “Calling...” and the phone number that was dialed:



Generally, phone numbers contain more than three digits. For design purposes, three digits are enough to show the design concept.

Designing the Screen

You will use the label object that is attached to Third_Number mode for the phone number in the Send_Call display. You will then duplicate this label object to create a label object that holds the text “Calling...”. The Duplicate operation copies a graphic object (but not to the Clipboard) and pastes it on the layout frame with a new name.

To add the dialed number to Send_Call mode:

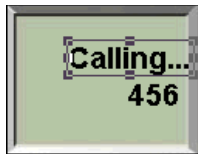
1. In the Mode Tree, select Dial/Third Number. Its screen is displayed in the Object Layout.
2. In the Object Layout, select the “456” display (Third_Num_Label).
3. In the Mode Tree, right-click Send_Call and choose Attach Third_Num_Label.

To add the Calling... label:

1. In the Object Layout, Ctrl+click-and-drag Third_Num_Label to duplicate it. When you release the mouse button, a copy of Third_Num_Label is added to the screen:



2. Double-click the new label object to open the Parameter Pane and rename it **Calling_Message**. Click .
3. Click . In the Label dialog box, type **Calling...** and click . The screen looks similar to this illustration:



4. Right-click the Calling_Message label object and choose Attach to Mode Send_Call.
 5. Center the two label objects in the screen. To do so, select them and the frame object (Frame1). Choose Layout|Center|Horizontal.
- ❖ *NOTE: The label object, Third_Num_Label, is now attached to two modes. Its position in each mode is retained, that is, in Third_Number mode it appears in the lower-right side of the screen and in Send_Call mode it appears in the center.*

Making the Transition from Dial/Third_Number to Send_Call

You will use the transparent pushbutton over the Send key for the transition.

To make the transition:

1. In the Object Layout, select the pushbutton object named Send_Key.
2. In the Mode Tree, right-click Third_Number mode and choose Make Transition.

3. For the To: value in the Make Transition dialog box, select “Send_Call.”
The other values:
From: Third_Number
Object: Send_Key
Event: in
are correct.
4. Click , then click .

Defining the Screen for Receive_Call Mode

In Receive_Call mode, the screen will display the message “Incoming Call!”:



You will design the screen for Receive_Call mode; however, you will not make transitions to it because the current application does not have a mechanism for sending calls to the cell phone. The completed application, *Tutorial2_Completed.rpd*, does implement this mechanism and you can learn how it was done by looking at that application.

Designing the Screen

You will use a label object for the Incoming Call message.

To define the screen:

1. In the Mode Tree, select Receive_Call mode.
2. In the Object Layout, add a label object. In its Label dialog box, change “Text” to **Incoming Call!** and redefine the font size to be **10 pt**.
3. In the Parameter Pane, name this object **Incoming_Call_Message**.
4. Position this object in the center of the screen using the Layout|Center|Horizontal and Layout|Center|Vertical commands.
5. Attach this object to Receive_Call mode.

Defining the Screen for Talk Mode

In Talk mode, the screen will display the time as it elapses:



The transition from Send_Call will be triggered by a timer. In the completed application, *Tutorial2_Completed.rpd*, the trigger will be a mechanism that simulates the call being answered.

Designing the Screen for Talk Mode

You will duplicate the Incoming Call label to create the display.

To define the screen:

1. Select the label object, Incoming_Call_Message, and duplicate it (Ctrl+click-and-drag).
2. In the Parameter Pane, name this new object **Time_Elapsed_Message**.
3. Click **More...**. In the Label dialog box, change “Incoming Call!” to **00:00:43**.
4. Attach this object to Talk mode.
5. Position this object in the center of the screen.

Making the Transition from Send_Call to Talk

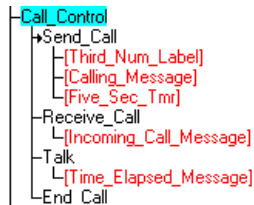
You will use the five second timer (Five_Sec_Tmr) for the transition.

To make the transition:



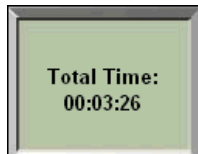
1. In the Mode Tree, select “Send_Call” mode.
2. In the Object Layout, double-click the Nongraphic Objects icon to open the Nongraphic Objects dialog box.
3. Select Five_Sec_Tmr in the list, then click **Attach To Mode** located at the bottom of the dialog box.
4. Click **Make Transition...**.

5. For the To: value in the Make Transition dialog box, select “Talk.”
The other values:
From: Send_Call
Object: Five_Sec_Tmr
Event: tick
are correct.
6. Click **Accept**, then click **Close**.
7. Close the Nongraphic Objects dialog box.
8. The completed Send_Call, Receive_Call, and Talk branches look like this illustration:



Defining the Screen for End_Call Mode

In End_Call mode, the screen will display the total time that elapsed during the duration of the call:



Designing the Screen

You will duplicate the 00:00:43 label object in Talk mode to create the total time elapsed label object. You will then duplicate this new label object to create a label for the “Total Time:” text.

To design the screen:

1. In the Mode Tree, select Talk.
2. In the Object Layout, duplicate the Time_Elapsed_Message object.

3. In the Parameter Pane, name this new object **Total_Time_Elapsed_Message**.
4. In its Label dialog box, change “00:00:43” to **00:03:26**.
5. Attach this object to End_Call mode.
6. Duplicate the Total_Time_Elapsed_Message object.
7. In the Parameter Pane, name this new object **Total_Time_Message**.
8. In its Label dialog box, change “Text” to **Total Time:.**
9. Attach this object to End_Call mode:



10. Position these objects in the center of the screen.

Making the Transitions



You will make two transitions:

- A transition from Talk to End_Call using the transparent pushbutton over the End/Power key.
- A transition from End_Call to Idle using the five second timer.

To make the transition from Talk to End_Call:

1. In the Object Layout, select the End_Power_Key pushbutton object.
2. In the Mode Tree, right-click Talk mode and choose Make Transition. In the Make Transition dialog box, all of the values are correct:
 - From: Talk
 - To: End_Call
 - Object: End_Power_Key
 - Event: in
3. Click , then click .

To make the transition from End_Call to Idle:

1. In the Mode Tree, right-click End_Call mode and choose Make Transition.
2. Make the transition: select:
To: **Idle**
Object: **Five_Sec_Tmr**
Event: **tick**
3. Click , then click .
4. Attach Five_Sec_Tmr to End_Call mode. To do so, Ctrl+click Five_Sec_Tmr, which is attached to Send_Call mode, and drag it to End_Call mode.
5. The completed End_Call branch looks like this illustration:

**Testing the Application in the Prototyper**

Now that you have added the Call_Control functionality, test the application before you build the screen transition chart.

**To run the application:**

1. Open the Prototyper and click the Start button.
2. Power on the cell phone.
3. Dial three numbers, then send the call.
4. Click the End/Power key to end the call.
5. When you have finished testing the application, close the Prototyper.

If your application didn't run as described above, open the supplied *Tutorial2_Completed.rpd* (located in \\Rapid70\Tutorials\RapidPLUSXpress) and compare it with your application.

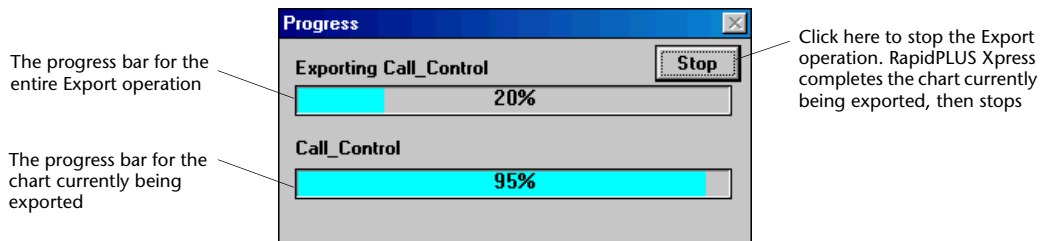
Viewing the Screen Transition Charts

You will now export the transition charts for Call_Control and its four children. There is a setting in the Chart Options dialog box that enables you to export the screen transition chart for a state and all the screen transition charts for its children.

To export transition charts for Call_Control and its children:

1. In the State Chart, choose Tools|Options to open the Chart Options dialog box and click the Screen Transition tab.
2. Select the Export Subtree Charts option, then click to close the dialog box.
3. Make Call_Control the top state, then export its transition charts.

The Progress dialog box indicates the status of the Export operation:



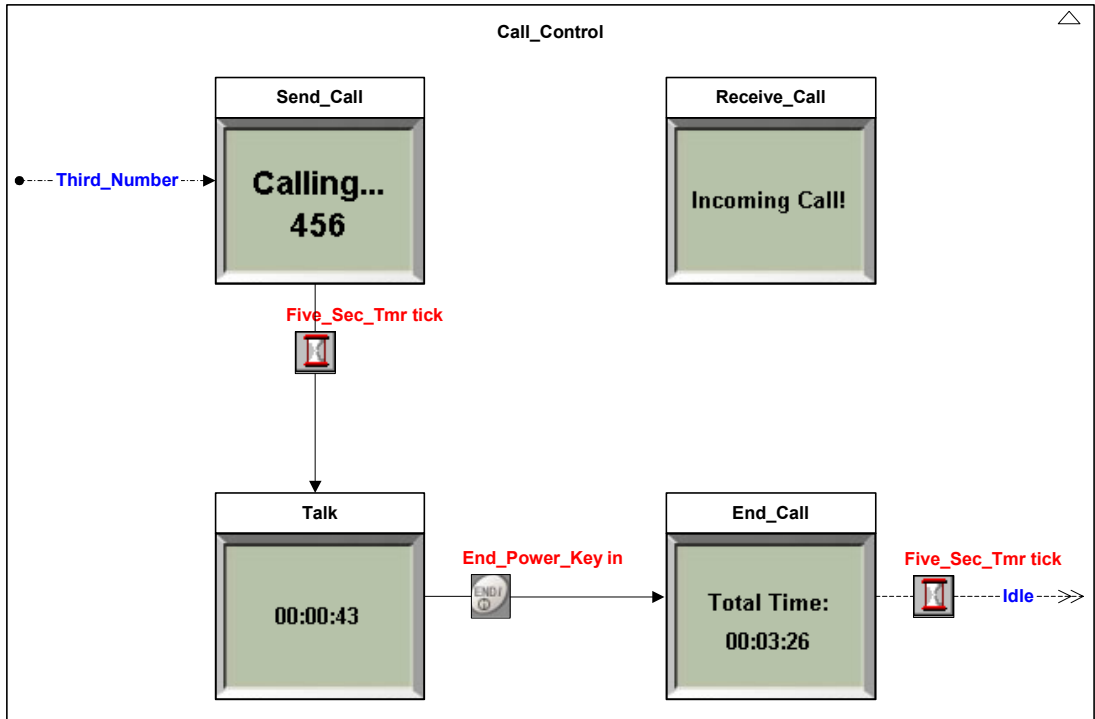
❖ *NOTE: RapidPLUS Xpress uses the Clipboard during the Export process. When a single chart is being exported, the process is quick and the use of the Clipboard is insignificant. However, when several charts are being exported, the process takes longer and the Clipboard will be unavailable in other Windows® programs until the Export operation is completed.*

The screen transition charts should look similar to the illustrations on the following pages.

Screen Transition Chart for Call_Control State

The following chart shows all the screens for Call_Control's children.

The dashed arrow that points to Send_Call indicates that there is a transition from Third_Number, a state outside of Call_Control. The dashed arrow exiting End_Call indicates that there is a transition to Idle, a state outside of Call_Control.

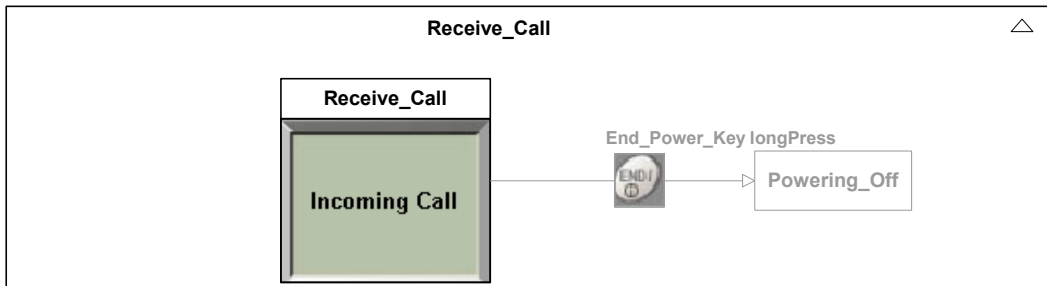
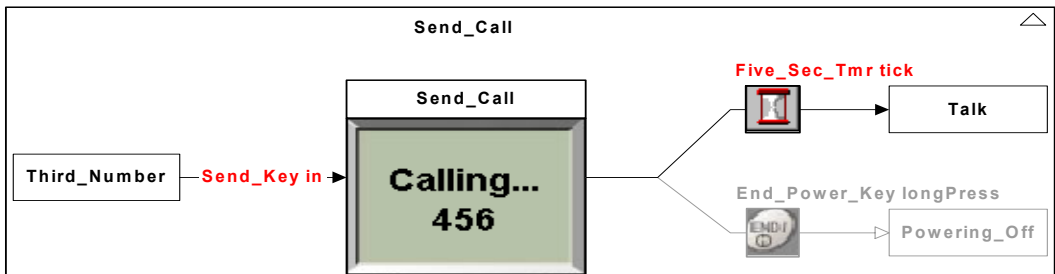


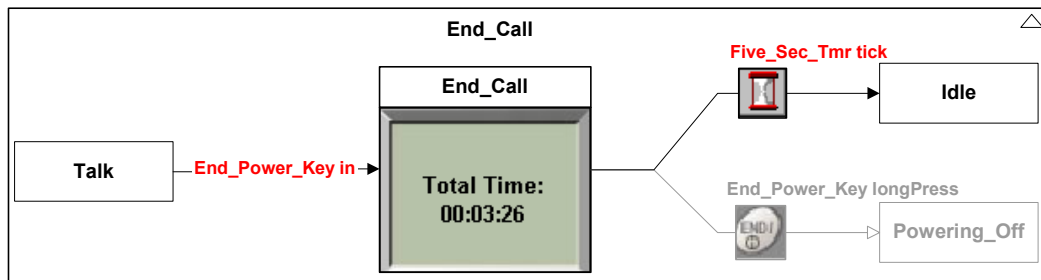
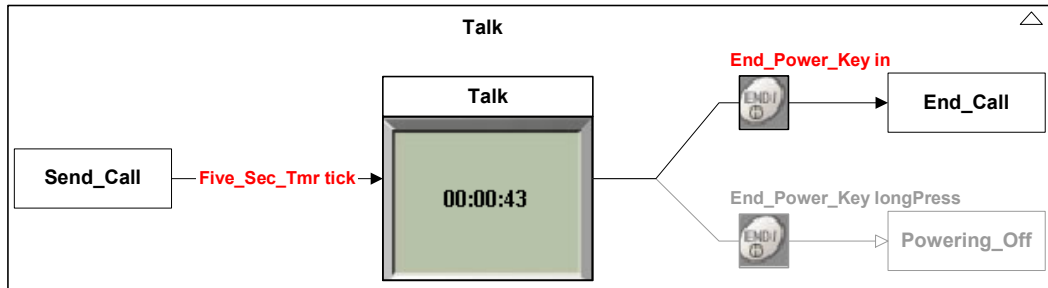
Screen Transition Charts for Call_Control's Child States

Call_Control's children do not have children; therefore, single-state screen transition charts are produced for each of them.

In each illustration, notice that there are images for the transition triggers' graphic objects exiting the states, but not entering the states. The entry and exit states are currently displayed as simple rectangles, but they can be displayed as images (by opening the Chart Options dialog box, Single State Transition tab, and selecting the options, Entry States and Exit States, in the Information Shown as Images group).

Notice the arrow in the upper-right corner of each chart. This arrow is a hyperlink to the parent's chart. To go to the parent's chart from a child's chart, right-click this arrow to open a popup menu and select the parent.





Stage 5 is completed. Be sure to save your application.

Congratulations!

You have completed the second of two tutorials about RapidPLUS Xpress. You have learned:

- How to work with mode hierarchy to design a logical and efficient RapidPLUS Xpress application.
- How to modify the appearance of screen transition charts.
- How to work with Rapid objects, attachments, and transitions.

Now you can move on to designing your own RapidPLUS Xpress applications. But before you do, open *CellPhone2_Completed.rpd*, the fully developed application, located in the \\Rapid70\\Tutorials\\RapidPLUSXpress folder. Examine how the Menus branch was developed. Also, examine the mechanism that was used to simulate the answering of phone calls.

Working in RapidPLUS Xpress

The purposes for building a RapidPLUS Xpress application are two-fold:

1. To build an application that presents a product's design and functionality.
2. To create screen transition charts from the application that document the product's design and functionality.

This chapter explains how to build applications and create screen transition charts. Information about screen transition charts is new in RapidPLUS Xpress and is completely documented in this manual. Information about building applications is presented in existing RapidPLUS documentation and is therefore not presented here. References to other documentation are given where appropriate.

This chapter presents:

- The workflow for building a RapidPLUS Xpress application.
- The RapidPLUS Xpress library of objects.
- How to define state images for the screen transition chart.
- How to work with transitions.
- How to create screen transition charts.
- Team development of RapidPLUS Xpress applications.

Before you begin working with RapidPLUS Xpress, we strongly recommend that you perform the tutorials presented in Chapters 2 and 3.

WORKFLOW FOR DEVELOPING AN APPLICATION

There are several stages in developing RapidPLUS Xpress applications. The development doesn't have to be followed in the exact order described below, but must include each stage:

Stage 1: Defining the Product's Functionality

A product's functionality is defined through **modes**. Modes are the distinct units of functionality that describe the system's behavior. For example, a cell phone can be on or off; therefore, On and Off are both modes of the cell phone application. The behavior of On mode can be described by additional modes such as Place a Call, Talk, and End a Call.

The product's functionality is organized logically and efficiently using a tree design in the Mode Tree tool. For details about designing and building a mode tree, see "Stage 1: Defining the System's States Using Modes" on p. 12 and "Stage 1: Defining the System's Modes" on p. 41. Information about modes is available in the online Help file in the "Working in the Mode Tree" topic.

Stage 2: Designing the Product's User Interface

A product's user interface comprises graphic and nongraphic elements that create the look of the product. The user interface is created in the Object Layout tool. For details about designing the user interface, see "Stage 2: Defining Background Graphics" on p. 14 and "Stage 3: Designing the Screen Elements" on p. 20. Information about graphic and nongraphic elements is available in the online Help file in the "Working in the Object Layout" topic.

This stage is usually performed in conjunction with the next stage.

Stage 3: Defining the Product's States

A **state** comprises one or more modes. It is a snapshot of a product at a given time. The appearance of each state is created by attaching graphic objects to the state's mode(s). The states are then used in the screen images that appear in the screen transition charts.

As mentioned above, Stages 2 and 3 are usually performed together. Graphic objects are added in the Object Layout tool and are then attached to the modes in the Mode Tree tool. For details about defining product states, see "Defining States" on p. 89.

Stage 4: Making Transitions Among Modes

A **transition** is a move from one mode to another that occurs because a specified event takes place. For example, the transition from Off mode to On mode will occur when an On/Off button is pressed. Transitions are important when the RapidPLUS Xpress application is run, because the transitions cause the application to move from mode to mode.

For details about making transitions, see “Working With Transitions” on p. 97.

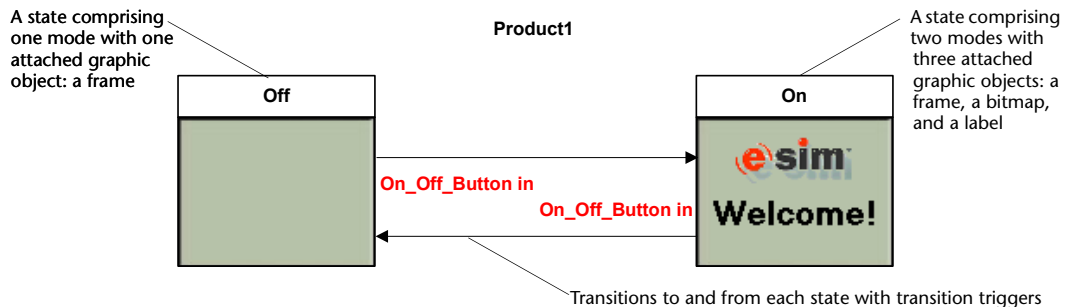
Stage 5: Testing the Application

The application can be tested to ensure that the transitions move the application from mode to mode. Testing takes place in the tool called the Prototyper. For details about running and testing an application, see “Running the Application in the Prototyper” on p. 37. Information about the Prototyper is available in the online Help file in the “Working in the Prototyper” topic.

Stage 6: Creating Screen Transition Charts

A screen transition chart is a diagram that displays screen images representing states and transitions among states. Screen transition charts are used to document product design concepts. They are generated in RapidPLUS Xpress and exported to Microsoft® Visio® (2000 and higher).















































The look of a screen transition chart is determined by two factors: the graphic objects that are attached to the various modes that make up each state and the Chart design options that are defined in the State Chart tool. A basic screen transition chart looks like this illustration:



For details about creating screen transition charts, see “Creating Screen Transition Charts” on p. 100. Information about the State Chart tool is available in the online Help file in the “Working in the State Chart” topic.

RAPIDPLUS XPRESS OBJECTS

RapidPLUS Xpress provides a library of objects that are particularly useful for designing product prototypes. Each object has its design and predefined logic. The library of objects includes the following objects, organized by object class in the Object Palette:

Class:	 Frames	 Circles	 Lines	 Bitmap
Objects:	 Square Corners  Square Corners Filled  Round Corners  Round Corners Filled  Out  In  Outline	 Circle  Filled Circle  Ellipse  Filled Ellipse	 Solid  Dashed  Dotted	 Bitmap
<hr/>				
Class:	 Label	 Time	 Signal	 Displays
Objects:	 Label	 Timer	 Event  Sound	 Text Display
<hr/>				
Class:	 Pushbuttons	 Switches	 Multimedia	 Animated
Objects:	 Flat  Square  Round	 Rotary  Vertical Toggle  Horizontal Toggle  Rocker  Vertical Slider  Horizontal Slider  Cross  Stepper  Joystick	 Wave Audio	 Animated

In addition to these objects, graphic files can be imported into the application to add realism (for example, a graphic of a cell phone and specialized buttons). After a graphic file has been imported, it is treated as a RapidPLUS Xpress bitmap object. There are also two system objects, Mouse and Keypad, that cannot be used in stand-alone RapidPLUS Xpress.

DEFINING STATES

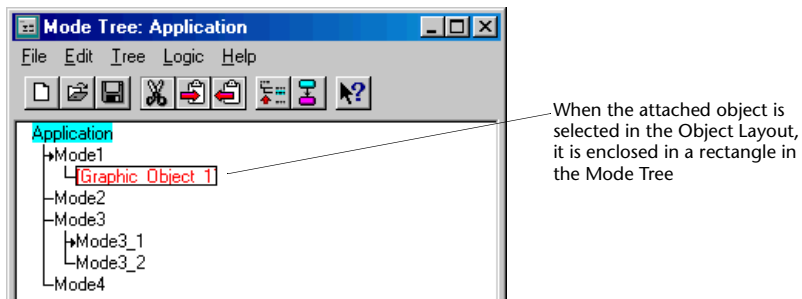
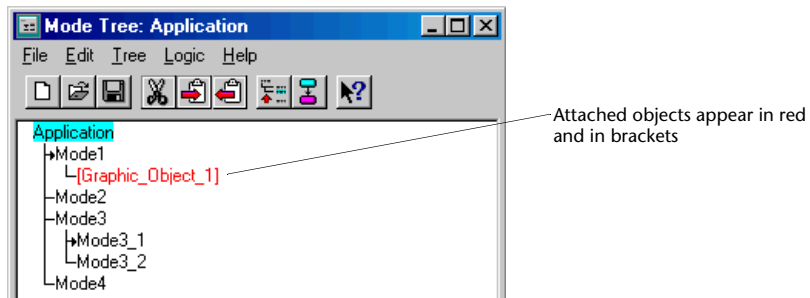
A **state** comprises one or more modes. It is a snapshot of a system at a given time. The appearance of each state is created by attaching graphic objects to the state's mode(s). The states are then used in the screen images that appear in the screen transition charts. Also, by attaching graphic objects to the state's modes, you can see in the Object Layout how each mode will appear when the application is run in the Prototyper.

Besides attaching graphic objects to modes, you can attach nongraphic objects (such as timers and .wav files), which are not seen in the application, but play an important role in the application's behavior.

The steps for defining a state are:

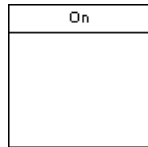
1. Add the appropriate objects in the Object Layout. For instructions about adding objects, refer to the topics "Adding graphic objects" and "Adding nongraphic objects" in the online Help.
2. Attach objects to the appropriate modes, as explained later in this section.

When an object is attached to a mode, the mode tree looks similar to these illustrations:



Attaching Graphic Objects to the Mode Tree

Graphic objects are attached to the mode tree to create the appearance of screens in transition charts. Without attaching images and/or graphic objects, states in screen transition charts are simple rectangles:



A typical state in a screen transition chart that has no attached graphic objects

With attached images and graphic objects, states in transition charts present how the product's screens will look in that state:



A typical state in a screen transition chart that has attached graphic objects.

The attached objects are a bitmap for the e-SIM logo, two label objects (for the time display and Menu option), and a transparent frame object (that shows part of the cell phone bitmap on which it was placed)

❖ *NOTE: Both of the states shown above comprise several modes. For example, the frame object is attached to the root mode, the e-SIM logo is attached to the On mode, and the labels are attached to the Idle mode. For details about the mode structure, see p. 63.*

Objects can be attached from the Object Layout and Mode Tree tools, using several Attach commands—either from the menu or pop-up menus. Attach commands are located in the following menus:

TOOL	MENU	COMMAND NAME
Object Layout	Edit	Attach to Mode: <name of mode selected in Mode Tree>
Mode Tree	Tree	Attach <name of object selected in Object Layout> Attach Selected Objects Attach Object <object class> <object name>

A graphic object can be attached to more than one mode. And it can be positioned in a different location in the Object Layout for each mode.

To attach a single graphic object:

1. In the Object Layout, select the object that will be attached.
2. In the Mode Tree, select the mode to which the object will be attached and right-click to open the popup menu.
3. Choose Attach|<name of selected object>.

To attach more than one graphic object:

1. In the Object Layout, select the objects that will be attached.
2. In the Mode Tree, select the mode to which the objects will be attached and right-click to open the popup menu.
3. Choose Attach Selected Objects.

To attach an object exclusively from the Mode Tree:

1. In the Mode Tree, select the mode to which the object will be attached and right-click to open the popup menu.
2. Choose Attach Object|<object class>|<object name>.

To attach objects from the Object Layout:

1. In the Mode Tree, select the mode to which the object(s) will be attached.
2. In the Object Layout, select the object(s) that will be attached and right-click to open the popup menu.
3. Choose Attach to Mode <name of selected mode>.

Attaching Nongraphic Objects to the Mode Tree

Nongraphic objects are attached to the mode tree to add effects (for example, sounds from *.wav* files or the sound object). Sounds will be heard when the application runs in the Prototyper; however, no reference is made to them in the screen transition charts.

Nongraphic objects can be attached from the Object Layout and Mode Tree tools.

To attach a nongraphic object from the Object Layout:

1. In the Mode Tree, select the mode to which the object will be attached.
2. In the Object Layout, open the Nongraphic Objects dialog box and select the nongraphic object from the list.
3. Click the Attach to Mode button located at the bottom of the dialog box; the object is attached to the mode.
4. Close the dialog box.

To attach a nongraphic object from the Mode Tree:

1. In the Mode Tree, select the mode that the object will be attached to and right-click to open the popup menu.
2. Choose Attach Object|<object class>|<object name>.

Behavior of Attached Objects

Attached objects affect an application and generated screen transition charts in several ways:

- **Attached graphic objects:** determine the look of states in screen transition charts. They also determine the look of an application when it runs in the Prototyper. Each attached graphic object appears when the mode to which it is attached is active and disappears when the mode is no longer active.
- **Attached nongraphic objects:** affect the behavior of the application when it runs in the Prototyper. They do not appear in screen transition charts. Nongraphic objects used in transitions are attached to enable the transitions to occur.

Certain objects must be attached to modes in order for them to function properly when the application is run in the Prototyper. These objects are:

OBJECT	BEHAVIOR WHEN ATTACHED TO A MODE
<i>Animation</i>	Runs through its frames one time when the mode is entered. In screen transition charts, only the first frame is displayed.
<i>Event</i>	The event occurs when the mode is entered.
<i>Sound</i>	The sound occurs when the mode is entered.
<i>Timer</i>	Timers used in transitions must be attached. Attaching the timer causes it to restart each time the mode is entered. When the mode is exited, the timer stops and its counter is reset to its initial value.
<i>WaveAudio</i>	The linked <i>.wav</i> file plays—from the beginning of the file—when the mode is entered. If the file is still playing when the mode is exited, the file stops playing.

Working with Attachments

After you have attached objects to the mode tree, you may want to manipulate them in the Mode Tree and/or Object Layout tools. This section explains how to work with attachments.

Moving Attachments

You can move an attachment from one mode to another.

To move an attachment:

1. Select the attachment that you want to move.
2. Drag it to the other mode, making sure that the mode is highlighted before you release the mouse button.

Duplicating Attachments

You can easily duplicate an attachment and attach it to a second mode. The Duplicate operation copies the attachment (but not to the Clipboard) and attaches it to the selected mode.

To duplicate an attachment:

1. Select the attachment that you want to duplicate.
2. Ctrl+drag it to the second mode, making sure that the mode is highlighted before you release the mouse button.

Cutting, Copying, and Pasting Attachments

Attachments can be cut, copied, and pasted using the standard Cut, Copy, and Paste commands.

Hiding attachments in a mode tree

You can hide all of a mode tree's attachments. This capability is useful when you want to view or work with the mode tree without seeing the attachments.

To hide all objects attached to the mode tree:

1. In the Mode Tree, choose Tree|Hide Attachments.

A check mark appears next to this command in the Tree menu.

To view the hidden attachments:

1. Choose Tree|Hide Attachments again.
-
-

Selecting Attached Graphic Objects

In the Mode Tree, all graphic objects that have been attached to a mode can be selected in a single operation using the Select Attachments command.

When this command is used, the graphic attachments of the selected mode and its descendants are selected in the Object Layout. This operation is cumulative; that is, previously selected attachments are not deselected when the attachments of another mode are selected.

To select attached graphic objects:

1. In the Mode Tree, select a mode.
2. Choose Edit|Select Attachments. All of the mode's graphic objects, and the graphic objects of its descendant, are selected in the Object Layout. In the Mode Tree, each selected object is enclosed in a rectangle.
3. Repeat steps 1 and 2 for all the modes whose graphic attachments you want to select.

To cancel the Select Attachments operation:

1. In the Object Layout, click an area outside of the selected graphic objects.

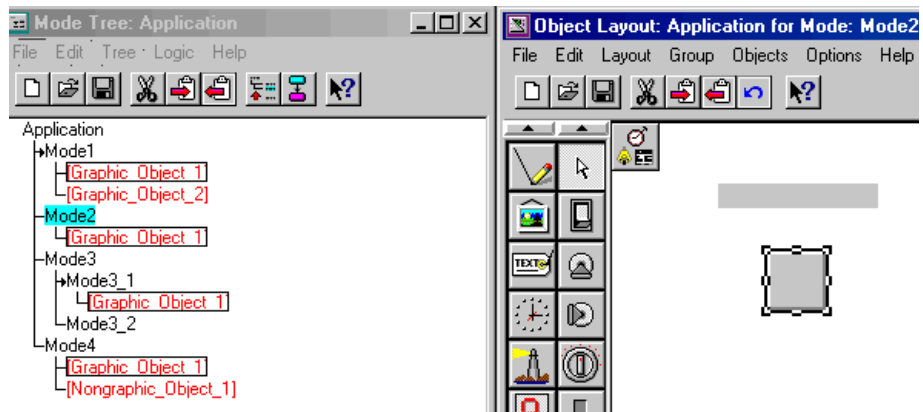
Selecting All Instances of Attached Graphic Objects

From the Object Layout, you can select a graphic object(s) and then use the Select All Attachments command to select all instances of its attachment in the Mode Tree. This command is useful when you want to make changes to all instances of an attached object(s).

To select all instances of an attached graphic object:

1. In the Object Layout, select the object.
2. Choose Edit|Select All Attachments.
SHORTCUT: Ctrl+click the object.

The attachment is selected in each of the modes to which the object is attached.



Example of a graphic object's attachments selected via the Select All Attachments command

Deactivating Attached Objects in a Mode Tree

You can deactivate objects—graphic and nongraphic—that are attached to a mode tree. A deactivated graphic object will not appear in the Object Layout, the Prototyper, or screen transition charts; a deactivated nongraphic object will not function in the Prototyper.

To deactivate an object attached to the mode tree:

1. In the Mode Tree, select the object you want to deactivate.
2. Choose Tree|Deactivated. A check mark appears next to this command in the Tree menu and the deactivated object appears in gray.

You can also select the Deactivated command from the object's popup menu.

To reactivate the object:

1. Select the deactivated object in the tree.
2. Choose Tree|Deactivated again.

Deactivating Ancestor Attachments

There will be times that you want a graphic object to appear in all descendant modes (children, grandchildren, and so on), so you will attach the object to the parent mode. There will also be times that you want the attached object to appear in all descendant modes except one or two.

A command, Deactivate Ancestor Attachment, deactivates the attached object in specific descendant modes. When this command is used, the ancestor attachment will appear in all of its descendant modes, except in the modes that have the attachment deactivated.

To deactivate an ancestor attachment:

1. In the Mode Tree, right-click the mode for which you want to deactivate the ancestor attachment.
2. Choose Deactivate Ancestor Attachment|<attachment name>.

The deactivated object is attached to the selected mode and appears in gray to indicate that it is deactivated.

To reactivate the ancestor attachment in the descendant mode:

1. Select the deactivated attachment (the one that appears in gray), then delete it.

The original attachment in the ancestor mode is not affected.

Example

The Menu_Options mode (parent) has numerous child modes that require a label object displaying the word “Menu”; however, two of the child modes require a label object displaying the word “Select”. The Menu label object is attached to Menu_Options mode and deactivated in the two child modes that do not require it.

WORKING WITH TRANSITIONS

A **transition** is a move from one mode to another that occurs when a specified trigger takes place. A **trigger** is an event of a specified RapidPLUS Xpress object. For example, the transition from Off mode to On mode will occur when an On/Off button is pressed.

Transitions are important when the Rapid application is run, because the transitions cause the application to move from mode to mode. Transitions are important in screen transition charts because they document how the product moves from one state to another.

A transition can have more than one trigger. For example, the transition from Place_a_Call mode to End_the_Call mode can be triggered by pressing the End button on the cell phone, by pressing the End button on a second phone, or by powering off the cell phone.

To define transitions, you add objects that will be used to trigger the transitions, then you make transitions using these objects’ events. Typical objects used in transitions are pushbuttons, switches, and timers.

A transition has four elements:

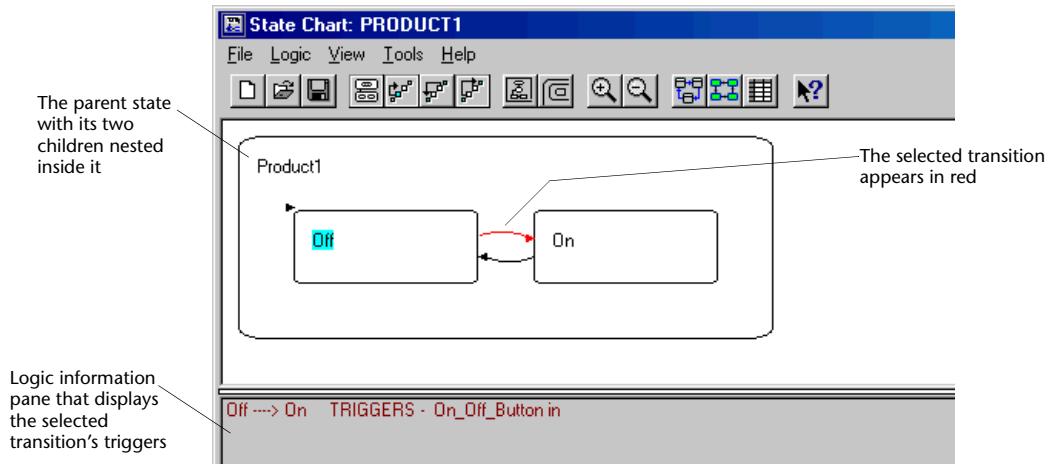
- A source mode (example: Off mode)
- A destination mode (example: On mode)
- An object (example: Pushbutton1)
- One of the selected object’s events (example: longPress)

The steps for defining a transition are:

1. Add the appropriate object in the Object Layout. For instructions about adding objects, refer to the topics “Adding graphic objects” and “Adding nongraphic objects” in the online Help.

2. Make a transition from one mode to another using one of the object's events. The transition is made in the Make Transition dialog box (see p. 99 for instructions).

When a transition is made, the State Chart looks similar to this illustration:



This application has two transitions. When a transition is selected in the chart, the arrow turns red. Transition details are displayed in the Logic Information pane below the chart (if View|Display Logic Info is selected).

Tips on naming objects used in transitions

Here are points to keep in mind when you name transition objects.

- Transition object names appear in transition charts together with their events. For example, "One_Min_Tmr tick." The longer the name, the more space it takes in the chart, which can lead to collisions with other transitions. Therefore, you should keep names as short as possible.
- Transition names should be descriptive. Timer2 is okay, but when an application has numerous timers, it is hard to remember their attributes. Battery_Timer is also a descriptive name, but it doesn't mention the time setting. If the time setting appears in the object's name, it is easier to reuse the object.
- Why not "1-min timer"? Numbers cannot be used as the first character in an object name. Dashes and spaces also cannot be used and will be substituted with an underscore (_).

- You could also use the name OneMnTmr; fewer characters, yet descriptive. Just make sure that readers of your transition charts will understand your naming conventions.

Making Transitions

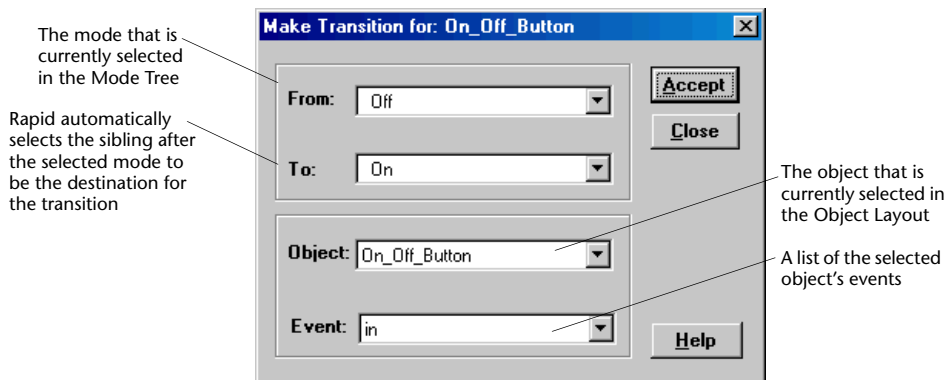
Transitions can be made from the Object Layout, Mode Tree, or State Chart using the Make Transition command—either from the menus or popup menus. This command is located in the following menus:

TOOL	MENU	COMMAND NAME
Object Layout	Edit	Make Transition
Mode Tree	Logic	Make Transition
State Chart	Logic	Make Transition

To make a transition using a graphic object:

An efficient way to make a transition is to use the following procedure:

- In the Mode Tree, select the mode that is the source of the transition.
- In the Object Layout, select the object that will trigger the transition.
- Select the Make Transition command (Ctrl+M); the Make Transition dialog box opens:



4. Make sure that the From, To and Object values are correct, then select the event value.
5. Click **Accept**, then click **Close**.

To make a transition using a nongraphic object:

1. In the Mode Tree, select the mode that is the source of the transition.
2. In the Object Layout, open the Nongraphic Objects dialog box (by double-clicking the Nongraphic Objects icon).
3. In the list of nongraphic objects, select the object that will trigger the transition.
4. Click **Make Transition...** located at the bottom of the dialog box; the Make Transition dialog box opens.

Deleting Transitions

You can delete transitions from the State Chart.

To delete a transition:

1. In the State Chart, select the transition that you want to delete; when selected, the transition appears in red.
2. Choose Logic|Delete Transition or press the Delete key. You can also select the Delete Transition command from the popup menu.

CREATING SCREEN TRANSITION CHARTS

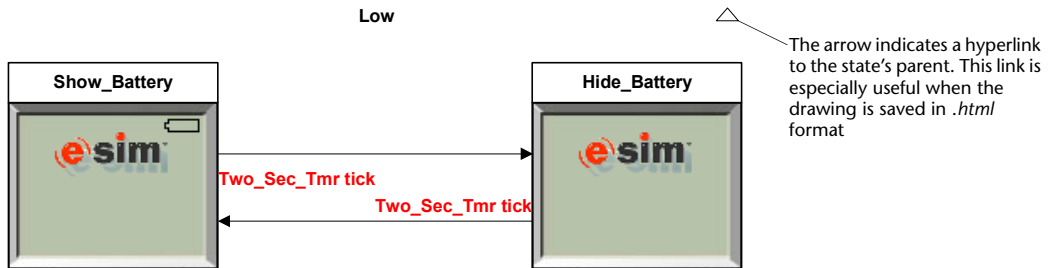
RapidPLUS Xpress produces two types of screen transition charts:

- Screen transition charts
- Single-state screen transition charts

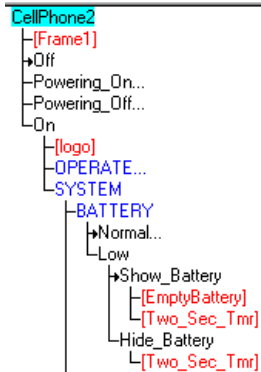
This section presents both types of charts, how to modify them, and how to export them. This section also presents information about the charts in Microsoft® Visio®.

Screen Transition Charts

A screen transition chart is generated for a state that has children. It shows transitions among the state's children. Here is an example of a screen transition chart for a state (Low) that has two children:



Compare the two child states. The screen image that appears in both of them is based on a transparent frame object placed on a bitmap of a cell phone's screen. The e-SIM logo is a bitmap that was attached to an ancestor mode. The difference in the appearance of the two states is that Show_Battery state has a battery bitmap attached to its mode and Hide_Battery state does not. Here is the corresponding mode tree:



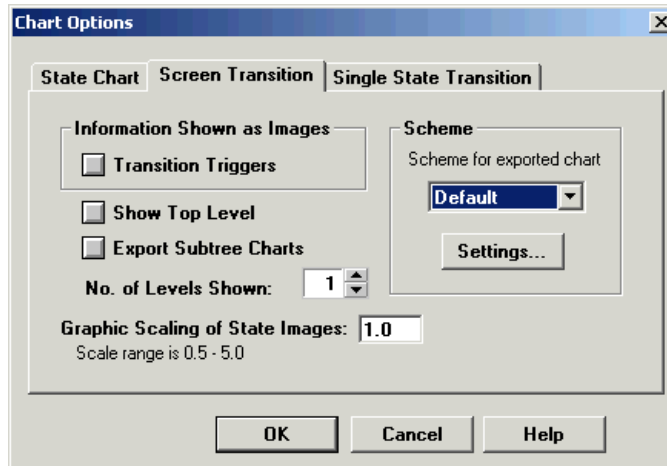
The same object (Two_Sec_Tmr) and event (tick) are used to trigger transitions between the two modes.

This chart was exported using the default settings of the Chart Options dialog box. These settings are explained in the following section.

Modifying the Appearance of Screen Transition Charts

To change the format of a screen transition chart:


1. In the State Chart, choose Tools|Options; the Chart Options dialog box opens.
2. Select the Screen Transition tab. The default settings are:



The dialog box options are described in the following table:

Options for Screen Transition Charts

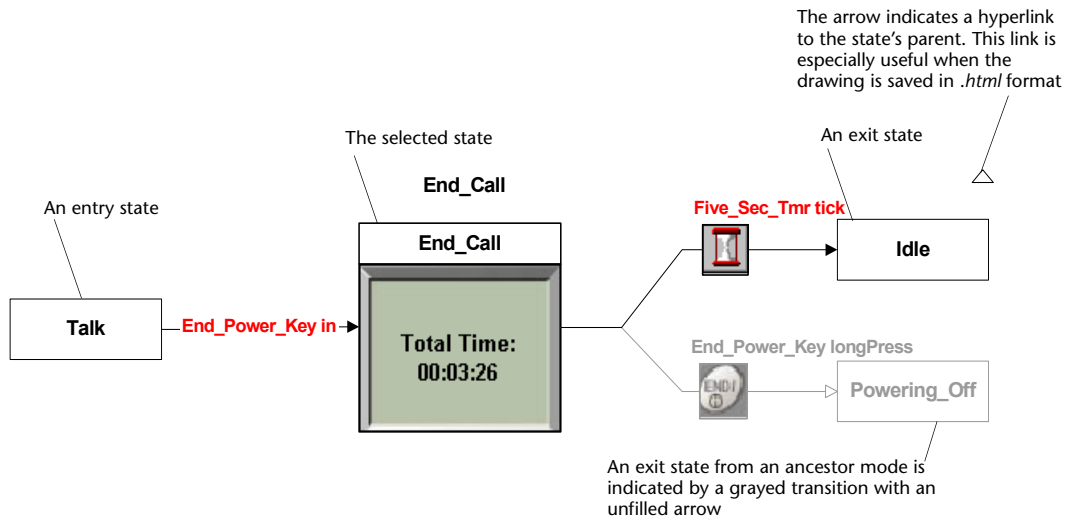
OPTION	DESCRIPTION
<i>Information Shown as Images: Transition Triggers</i>	<p>When not checked, each transition trigger (the object and its triggering event) is displayed as text only. The font for this text is set via the Label Font dialog box.</p> <p>When checked, the object in each transition trigger is displayed as a graphic image.</p> <p>Default: not checked</p> <p>❖ <i>NOTE: This option is not available when the "No. of Levels Shown" is set to 2.</i></p>

OPTION	DESCRIPTION
<i>Show Top Level</i>	<p>When checked, the selected state is shown enclosed in a border. When not checked, the selected state is not enclosed.</p> <p>Default: not checked</p>
<i>Export Subtree Charts</i>	<p>When not checked, a single screen transition chart is exported for the selected state.</p> <p>When checked, transition charts (screen and/or single-state) are generated for the selected state and ALL its descendants (children, grandchildren, great-grandchildren, etc.).</p> <p>Default: not checked</p>
<i>No. of Levels Shown</i>	<p>The number of levels that will appear in the screen transition chart, either 1 or 2.</p> <p>A setting of one level shows the selected state and its children. A setting of two levels shows the selected state, its children, and grandchildren.</p> <p>Default: 1</p> <p>❖ <i>NOTE: When the setting is 2, the "Transition Triggers Shown as Images" option is unavailable.</i></p>
<i>Graphic Scaling of State Images</i>	<p>The scaling factor for the bitmap representation of states. 1.0 means 1:1, or the same size as in the Object Layout window; .5 reduces the image by half; 2.0 means 2:1, or twice the size of the object image in the Object Layout; and so on.</p> <p>Range: 0.5 to 5.0</p> <p>Default: 1.0</p>
<i>Scheme</i>	<p>The Scheme list presents predefined schemes that determine the appearance of textual elements in charts. You can use the supplied schemes or create your own by selecting <New> from the list and clicking . All supplied and new schemes are part of a RapidPLUS stencil, <i>schemes.vss</i>. For details about creating new schemes, see "Creating New Chart Schemes" on p. 107.</p>

Single-State Transition Charts

A single-state transition chart is the transition chart that Rapid generates for a state that does not have children. It shows transitions to and from the selected state. Transitions to the selected state come from “entry states” and transitions from the selected state go to “exit states.”

Here is an example of a single-state transition chart:



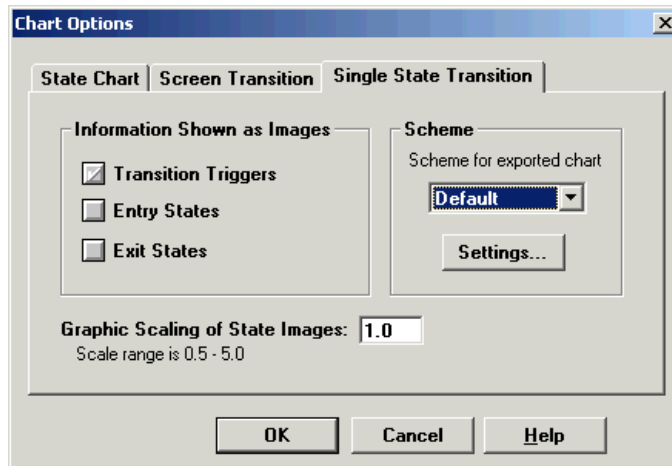
Notice that the entry and exit states appear as simple rectangles. Also notice that the transition from **Talk** mode to **End_Call** mode appears as text only, whereas the transitions from **End_Call** mode appear with images for the transition objects. The appearance of entry states, exit states, and transitions is set in the Chart Options dialog box.

The chart shown above was created using the default settings. These settings are explained in the following section.

Modifying the Appearance of Single-State Transition Charts

To change the format of a single-state transition chart:


1. In the State Chart, choose Tools|Options; the Chart Options dialog box opens.
2. Select the Single-State Transition tab. The default settings are:



The dialog box elements are described in the following table:

Options for Single-State Transition Charts

OPTION	DESCRIPTION
<i>Information Shown as Images: Transition Triggers</i>	<p>When not checked, each transition trigger (the object and its triggering event) is displayed as text only. The font for this text is set via the Label Font dialog box.</p> <p>When checked, the object in each transition trigger is displayed as a graphic image.</p> <p>Default: checked</p>





OPTION	DESCRIPTION
<i>Information Shown as Images: Entry States</i>	<p>When not checked, each entry state (a state that has a transition to the selected state) is displayed as a simple rectangle.</p> <p>When checked, each entry state is displayed as it was designed in the Mode Tree, that is, with all of its attachments.</p> <p>Default: not checked</p>
<i>Information Shown as Images: Exit States</i>	<p>When not checked, each exit state (a state that has a transition to it from the selected state) is displayed as a simple rectangle.</p> <p>When checked, each exit state is displayed as it was designed in the Mode Tree, that is, with all of its attachments.</p> <p>Default: not checked</p>
<i>Graphic Scaling of State Images</i>	<p>The scaling factor for the bitmap representation of states. 1.0 means 1:1, or the same size as in the Object Layout window; .5 reduces the image by half; 2.0 means 2:1, or twice the size of the object image in the Object Layout; and so on.</p> <p>Range: 0.5 to 5.0 Default: 1.0</p>
<i>Scheme</i>	<p>The Scheme list presents predefined schemes that define the appearance of textual elements in charts. You can use the supplied schemes or create your own by selecting <New> from the list and clicking . All supplied and new schemes are part of a RapidPLUS stencil, <i>schemes.vss</i>. For details about creating new schemes, see the following section.</p>

Creating New Chart Schemes

RapidPLUS provides a stencil named *schemes.vss* to which you can add additional scheme masters. This stencil contains several predefined masters; the master named “Default” is the basis for all predefined and new masters.

❖ *NOTE: Only modify the Default master if you want all subsequent masters to contain these modifications.*

To create a new chart scheme:

1. Open the Chart Options dialog box (Tools|Options) to either the Screen Transition or Single State Transition page.
2. In the Scheme list, click the arrow and select <New>, then click . Microsoft Visio opens to a new master named “Default.<consecutive number>”. The *schemes.vss* stencil is also opened, but is not in focus.
3. Modify any or all of the following elements:
 - Mode names
 - Transition text (default, history, and deep history)
 - Note text
 - Indirect transitions: text and line color(Although other elements can be modified, they will not be applied to the charts.)
4. Save the new scheme. To do so:
 - a. Close the master.
 - b. In the Update dialog box that opens, click .
 - c. In the *scheme.vss* stencil, right click the new master (named “Default.<number>”) and choose Master Properties.
 - d. Edit the Master’s name, then click .
 - e. Close *schemes.vss*. In the Save Changes dialog box, click .

The newly created scheme is added to the Scheme list in the Chart Options dialog box.

Exporting Transition Charts

Screen transition charts and single-state transition charts are exported from either the State Chart or the Mode Tree tools.

To export a transition chart from the State Chart:

1. In the State Chart, double-click the state for which you want to export a transition chart so that it appears as the top state.
2. Choose Tools|Export Transition Chart, or click the Transition Chart button.



Ctrl+Shift+T

To export a transition chart from the Mode Tree:

1. In the Mode Tree, select the state for which you want to export a transition chart.
2. Choose File|Export Transition Chart.

Microsoft Visio opens, first displaying a Legend page and then the transition chart.

❖ *NOTE: The Export Subtree Charts option in the Chart Options dialog box, Screen Transitions tab, enables you to export transition charts for the selected state and all of its descendant states.*

A Consideration When Exporting Transition Charts

RapidPLUS Xpress uses the Clipboard during the Export process. When a single chart is being exported, the process is quick and the use of the Clipboard is insignificant. However, when several charts are being exported, the process takes longer and the Clipboard will be unavailable in other Windows programs until the Export operation is completed.

Adding Notes to Screen Transition Charts

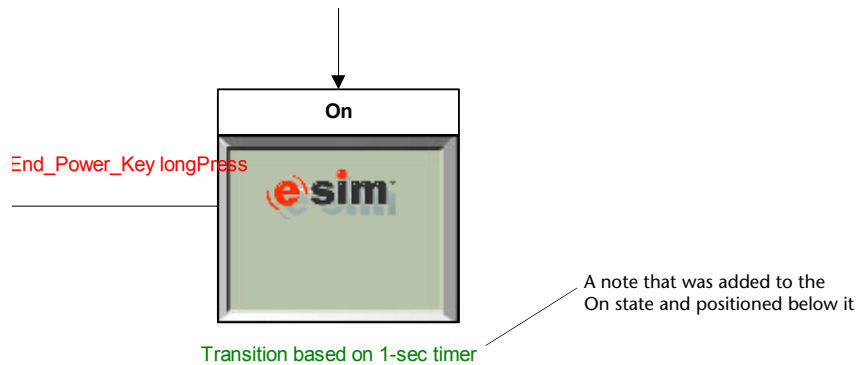
You can add notes in Rapid which will then be included in screen transition charts next to states and/or triggers. Notes are written and positioned in the small Rapid window.

Adding Notes to States

Notes can be positioned in six places: above the state, below it, to its left, to its right, as a footnote, or as its title. You specify the position when you add the note.

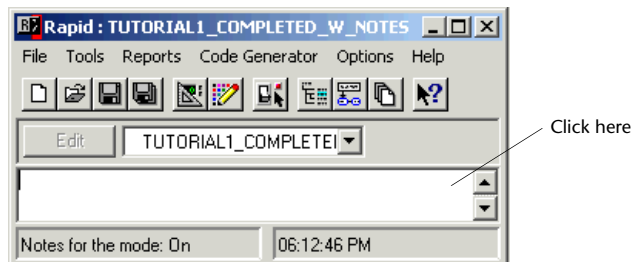
Example

This illustration was taken from a screen transition chart. It shows a state that has a note:



To add a note:

1. In the Mode Tree or State Chart, select the mode to which you want to add a note.
2. In the small Rapid window, click in the Notes area:



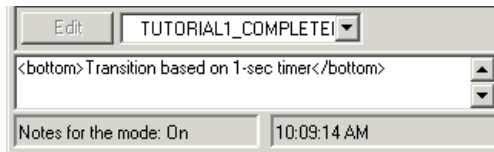
- Position the note. To do so, right click in the Notes area to access the popup menu and choose Tags. The submenu for the Tags command displays six positions: Title, Top, Left, Bottom, Right, and Footnote. Choose one of the positions.

For example, to create the note shown on the previous page, you would choose **Bottom**.

A special formatting tag is added, which is similar to an HTML tag. In this example, the following tag would be added:

```
<bottom></bottom>
```

- Insert the cursor between the opening and closing tags and type the note. For example:



When you export the screen transition chart, the note will appear in the selected position.

Information about notes in transition charts

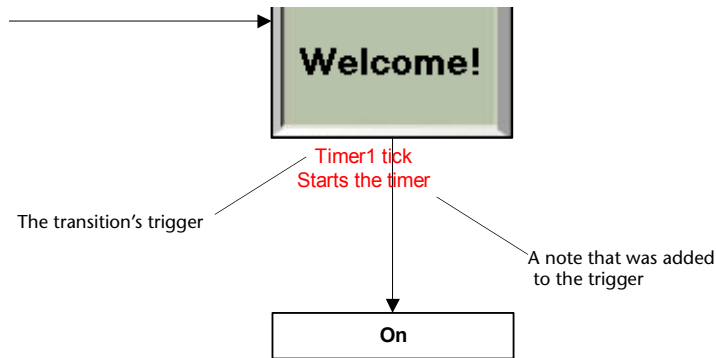
- The appearance (font, font color, etc.) of each note in Microsoft Visio is determined by the transition chart's Legend page. You can change the appearance of notes by changing their appearance on the Legend page. Changes are applied to transition charts that are exported *after* the changes were made.
 - ❖ *NOTE: On the Legend page, only the notes style can be changed.*
 - A state can contain any or all of the note types. In addition, a state can contain multiple tags of the same type.
 - Notes that do not appear inside tags are not exported.
 - If the Title tag is used, the note replaces the state's title. The note will also appear in outgoing transitions (transitions to modes outside the current transition chart) in place of the state's title.
 - If you want a note to contain angle brackets (< and >), you must type them twice (<< and >>).
-

Adding Notes to Transition Triggers

A trigger is the event of a specified RapidPLUS Xpress object that causes a transition to occur. In a screen transition chart, notes appear below the trigger.

Example

This illustration was taken from a screen transition chart. It shows a trigger that has a note:



To add a note to a trigger:

1. In the State Chart window, select the transition that contains the trigger to which you want to add a note. To select a transition, click its arrow.

When a transition is selected, it appears in red and all of its triggers are displayed in the Logic information pane at the bottom of the window. The top-most trigger is selected by default.

❖ *NOTE: If the Logic information pane is hidden, you can show it by choosing View|Display Logic Info.*

2. In the Logic information pane, select the relevant trigger.
3. In the small Rapid window, click in the Notes area (as shown on p. 109) and type the note.

When you export the screen transition chart, the note will appear.

WORKING WITH TRANSITION CHARTS IN MICROSOFT VISIO

When a transition chart is generated in RapidPLUS Xpress, it is exported to Microsoft® Visio® (2000 or higher); therefore Visio must be installed on your system.

The first time that a state's transition chart is generalized, Visio opens and displays a new *.vsd* drawing comprising two pages: the first page displays a legend and the second page displays the transition chart. The *.vsd* file is assigned a default name based on the name of the application and is placed in the folder in which the RapidPLUS Xpress application is located. The page displaying the transition chart is assigned a default name based on the name of the state.

Each time a transition chart is generated for that specific state, its page is overwritten. Each time a transition chart is generated for a different state in the application, a new page is added to the Visio drawing. When a state's screen transition chart is exported together with its subcharts, a separate page is added for each state's transition chart.

❖ *NOTE: If the RapidPLUS Xpress application is located in a folder on a server and you don't have permission to save to that folder, RapidPLUS Xpress places the .vsd file in your TEMP folder.*

Modifying Transition Charts

You can edit exported transition charts in Visio. However, if you edit a state's transition chart in Visio and then re-export the state's transition chart, your changes in Visio will be overwritten.

To prevent overwriting your changes, save the modified *.vsd* file with a different name.

Printing Transition Charts

The size of a Visio page is determined by the Export operation's output. RapidPLUS Xpress sets the Print Zoom to "Fit to 1 sheet across by 1 sheet down."

This setting ensures that each chart will fit onto one page when it is printed; however, it also means that the sizes of the screens may vary from chart to chart. If you have a printer that can print large pages, you can change the Print Zoom to "Adjust to 100%."

Distributing Transition Charts Over the Web

In Visio, you can save the screen transition charts to *.html* format for distribution over the Web. When RapidPLUS Xpress generates a transition chart, it adds hyperlinks to all of the transition chart's elements (states, transitions, etc.) Also, an arrow at the top of each transition chart is a hyperlink to the state's parent.

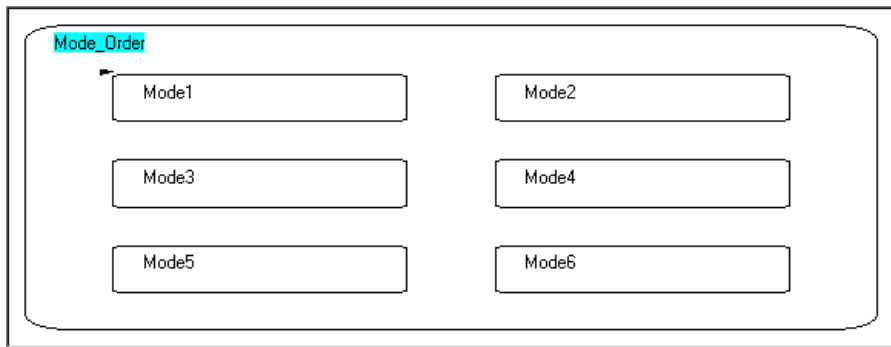
These hyperlinks enable you to navigate easily among all of the transition charts.

TIPS FOR BUILDING SUCCESSFUL APPLICATIONS

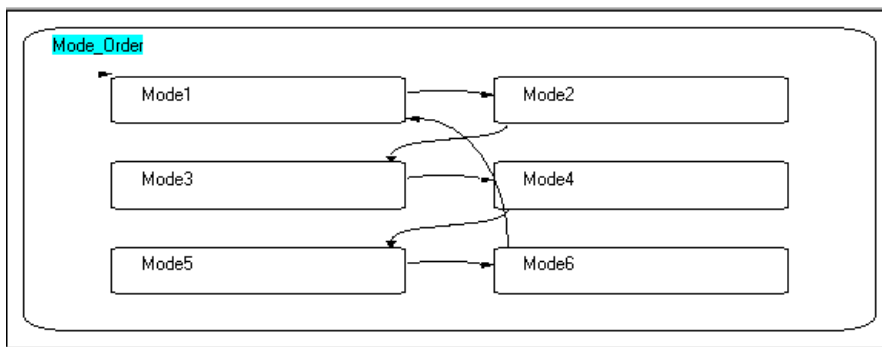
Below is additional information to assist you in building successful RapidPLUS Xpress applications. Most of the information applies to the stand-alone version of RapidPLUS Xpress; the last section applies to standard RapidPLUS programs.

Changing the Order of Modes

In the State Chart tool, child modes are nested within their parent mode. Within the parent mode, child modes are arranged in left-to-right order in columns of two as shown below:



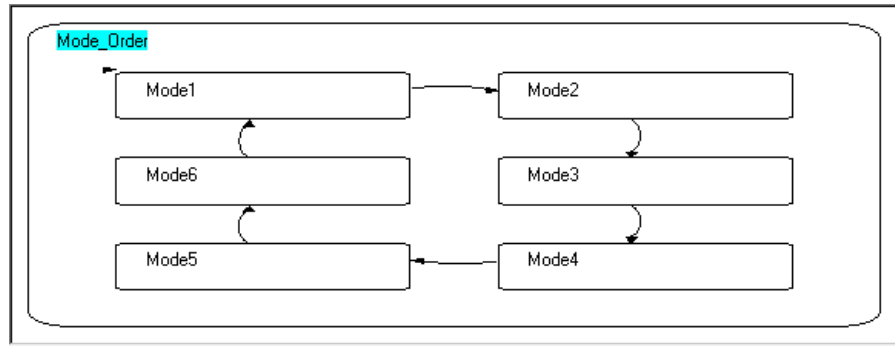
This layout (routing) method is known as left-to-right routing. When transitions are added, the state chart looks like:



In Rapid simulations, left-to-right routing works fine; however, when this chart is exported, the overlapping transition lines will remain and may make the screen transition chart confusing. You can change the routing by rearranging the modes in the Mode Tree. There are several ways to rearrange modes:

- Drag modes along the tree.
- Use the Alt + up/down keys.
- Use the Move command (Edit|Move).

For details about moving modes, refer to the “Moving Modes” topic in the online Help. In the following example, the six modes were rearranged in the Mode Tree to provide continuous routing:



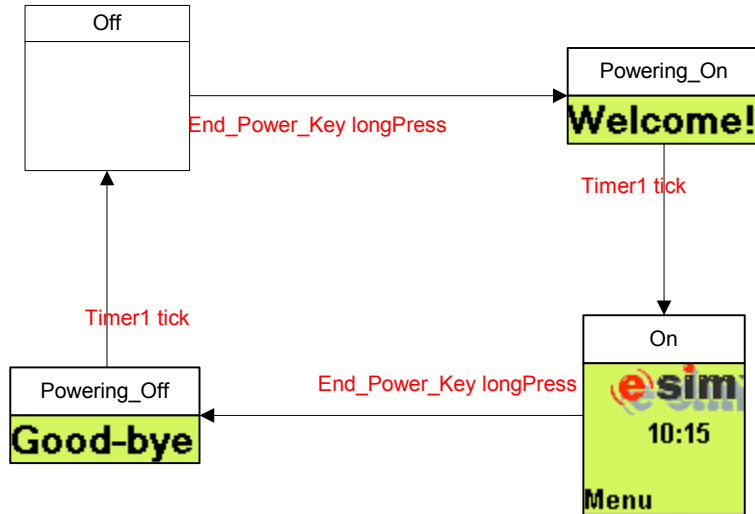
To see how rearranged modes affect a screen transition chart, see “Designing the Charts” on p. 34.

Building Uniform States in Screen Transition Charts

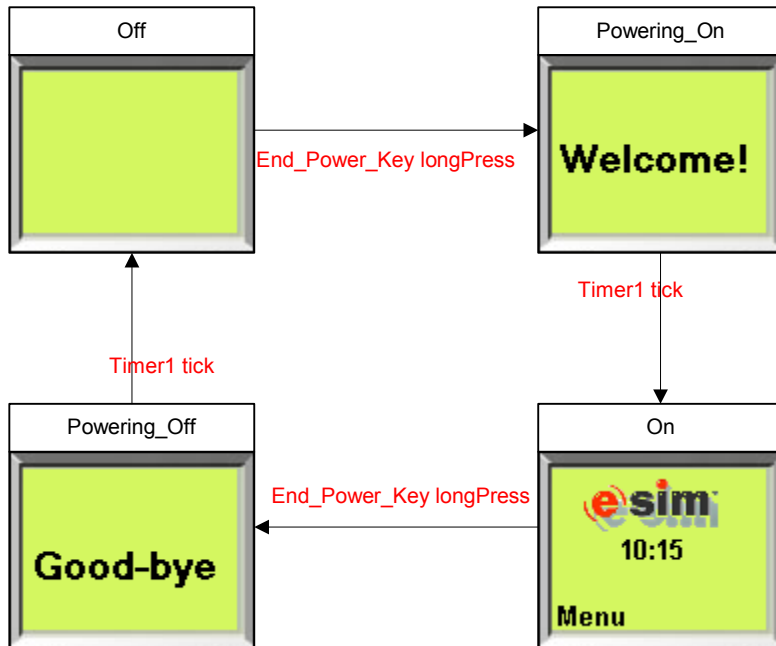
You can attach a frame object to the root mode to provide a background for the states in transition charts and ensure that the size of the states will be uniform. See “Adding a Background Frame Object” on p. 17 for an example. Without the attached frame object, the size of each state is determined by its attachments.

Compare the following two screen transition charts. The first one does not have a frame object attached to its root mode and the second one does. Notice how the attached frame object in the second chart provides a uniform look for the states. This uniformity not only provides legibility to the chart, but also shows how the screen will look when this state is achieved.

A sample chart without a background frame



A sample chart with a frame attached to the root mode



TEAM DEVELOPMENT OF RAPIDPLUS XPRESS APPLICATIONS

A team of designers can work together to develop an application. Team development is especially useful when a feature-rich application is being developed.

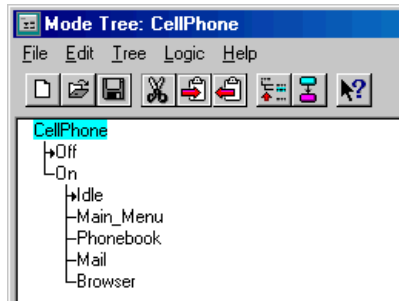
To assist you with team development, two methodologies are presented in this section: one methodology utilizes user objects, the other does not. A **user object** is a RapidPLUS application that is built specifically for nesting inside another RapidPLUS application (known as the parent application).

User objects can be re-used within the same application—and in other applications as well. Once built, they reduce the work of recreating functionality. A Rapid-supplied user object, *User_Object_Template.udo*, is the basis for all user objects added to the parent application.

Both methodologies rely on a master mode tree and common graphics.

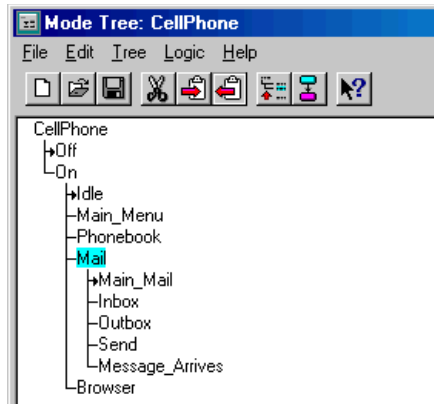
Developing an Application Without User Objects

1. The team leader designs a master mode tree. For example:



2. The team leader adds the objects that are common to the entire application, such as the bitmap of the cell phone and pushbuttons for the keypad keys.
3. The team leader gives a copy of the master application to each designer.
4. The team leader assigns each designer a mode to develop. For example, the designer in charge of the Mail feature will develop the Mail mode.

5. Each designer builds the assigned mode/branch making sure not to change the names of the modes or the structure of the master mode tree. For example, the designer in charge of the Mail feature might add the following child modes to Mail mode:



6. The designers give their completed applications to the team leader.
7. The team leader copies and pastes each mode/branch into the master application. For example, from the Mail application, copy the Mail mode; its attached objects and logic are also copied. In the master application (*CellPhone.rpd*), select On mode and paste the new Mail branch.
8. When a copied mode is pasted into the master mode tree, its attached objects and logic are pasted with it. For each duplicate bitmap object, the Paste Conflicts dialog box will open. Do not choose the Keep Both option; otherwise the master application will contain more than one version of the same bitmap.

We recommend that you choose the **Keep original** option. This option ensures that the common graphic objects were not changed by the designers.

Developing an Application With User Objects

When you develop an application with user objects, you must base all the user objects on the Rapid-supplied user object, *User_Object_Template.udo*.

This section explains how to work with the user object template, describes the interface elements in the user object template, provides a development methodology, and explains how to use a keypad user object.

Working with the User Object Template

The file, *User_Object_Template.udo* has a special interface that was designed to communicate with the parent application in which it is nested. The template contains:

- Two modes—Idle and Active.
- One graphic object that gives the user object its graphic representation in the Object Layout of the parent application.
- Two nongraphic objects—event objects—that trigger transitions between the user object's modes.
- Two exported functions that, when called by Rapid, activate the event objects.

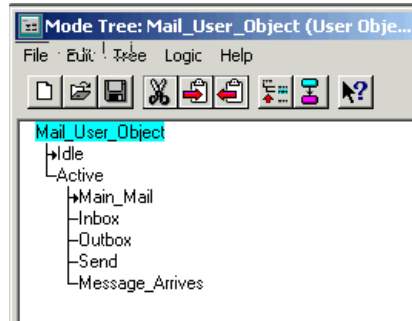
These interface elements are described in detail on p. 121.

Each time you build a new user object, you will start with the user object template.

To build a new user object:

1. Open *User_Object_Template.udo*. It is located in the \\Objects folder.
2. Choose File|Save As, select your working folder, and give the application a meaningful name. Notice that it is saved with the *.udo* extension.

3. Build the user object, making sure that changes or additions that you make will not adversely affect the parent application. All modes should be children of Active mode, as shown in the following example:



A sample mode tree of a user object

What Happens When the User Object is Active

The user object and its parent application communicate via exported functions and events. The user object template includes two exported functions: `xpress_start` and `xpress_stop`.

When the parent application runs and the mode to which the user object is attached becomes active, the following things occur:

- a. Rapid shows the user object.
- b. Rapid enters the user object's Idle mode.
- c. Rapid calls the `xpress_start` exported function. This function triggers an event, which then causes a transition in the user object to Active mode.
- d. Active mode contains user-defined modes (such as the five child modes shown in the illustration above). These modes are activated by internal events defined in the user object.
- e. When the mode to which the user object is attached in the parent application is exited, Rapid calls the `xpress_stop` function, which activates a transition to Idle mode in the user object.
- f. The user object is hidden.

The User Object Template's Interface Elements

TYPE	NAME	DESCRIPTION
Exported functions	xpress_start*	A function that is used in the parent application to trigger the event object, xpress_onStart_ev. This function is called when the mode to which the user object is attached is activated. Do not delete or modify this exported function.
	xpress_stop*	A function that is used to trigger the event object, xpress_onStop_ev. This function is called when Active mode is exited. Do not delete or modify this exported function.
Objects	xpress_onStart_ev*	A nongraphic event object that is used to trigger a transition from Idle mode to Active mode. Do not delete or modify this object.
	xpress_onStop_ev*	A nongraphic event object that is used to trigger a transition from Active mode to Idle mode. Do not delete or modify this object.
	Frame1	A named, frame object that can be used to attach the user object to a mode. This object can be replaced by any other graphic object.
Modes	Idle	An exclusive mode whose transition to Active mode occurs when the event object, xpress_start_ev, is triggered. Do not delete or modify this mode.
	Active	An exclusive mode whose transition to Idle mode occurs when the event object, xpress_stop_ev, is triggered. Do not delete or modify this mode.

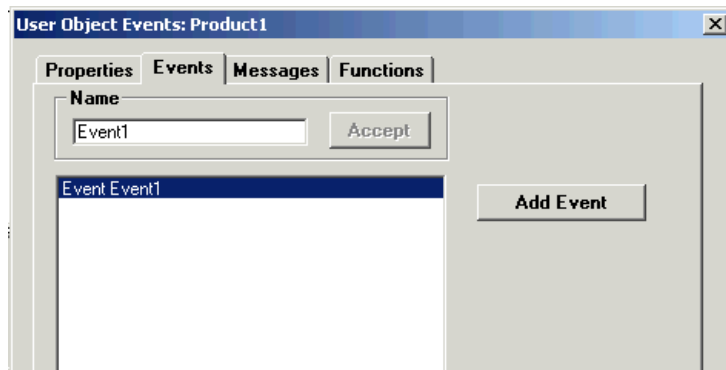
* The event objects and exported functions are used in logic that can only be accessed when the program is in standard RapidPLUS mode of operation. This logic should not be changed.

Adding Exported Events to the User Object

Exported events are used to trigger transitions in the parent application. The user object template does not contain predefined exported events. You must add them as necessary.

To add exported events:

1. Open your user object.
2. In the Object Layout, double-click the layout making sure that no object is selected. The Parameter Pane opens for the user object.
3. Click **More...**; the User Object Functions dialog box opens.
4. Click the Events tab.
5. Click **Add Event**. An event is added and assigned a default name:

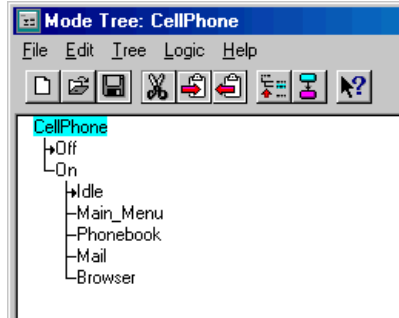


6. Change the default event name by overtyping it in the Name box, then click **Accept**.

Repeat steps 5 and 6 for each exported event that you want to add.

The Development Methodology

1. The team leader designs a master mode tree. For example:



2. The team leader determines which of the modes' functionalities will be based upon user objects.
3. The team leader assigns each designer a user object to develop. For example, the designer in charge of the Mail feature will develop the Mail user object.
4. Each designer builds the assigned user object based on the template user object (see instructions on p. 119).
5. The designers give their completed user objects to the team leader.
6. The team leader places the user objects in the same folder as the parent application, then adds them to the parent application using the Objects|Add User Object command.
7. The team leader attaches each user object to its appropriate location in the mode tree and makes transitions to and from the user object using its exported events.

Working with a Keypad User Object

An application for a mobile device usually requires a keypad. You can develop the keypad as a user object, which can then be added to the parent application and to all user objects that require keypad inputs. The keypad user object does not need to be based on *User_Object_Template.udo* because its functionality does not require the predefined interface elements (*xpress_start*, *xpress_stop*, etc.).

In order to add the keypad user object, be sure to save it to the same folder as the parent application and user objects. Do not attach the keypad user object to any modes.

In the parent application, the keypad user object must be at the bottom of the stack order so that each of the user object keypad will be active when the user object is active. To send the parent application's keypad user object to the bottom: select it, then choose Layout|Order|Send to Back.

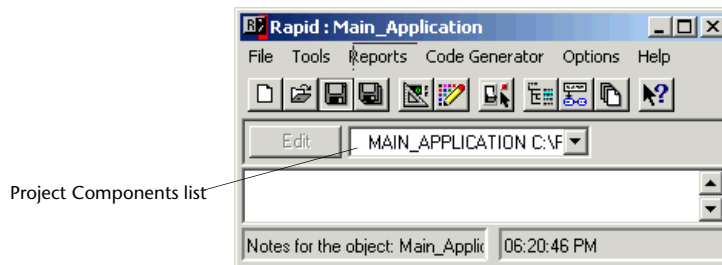
Refer to the Xpress example application located in the \\Examples folder to see how a keypad user object was developed and utilized.

Guidelines for Generating Screen Transition Charts from User Objects

- Be sure that the option, "Information Shown as Images: Transition Triggers," is not selected when generating a screen transition chart for the mode that contains the user object; otherwise, all of the user object's graphic objects will be shown.
- When generating transition charts for an application that contains user objects, charts are only generated for the parent application.

To generate screen transition charts for a user object:

1. You must first switch the context to the required user object. To do so, click in the Application Manager window. Notice the Project Components list that displays the name of the parent application:



2. Click the down arrow and select the relevant user object.
3. Generate screen transition charts as you would for any application.

Using Common Graphics During Team Development

Sometimes, several members of the development team will need to use the same graphics. In these cases, there should be one source for all of the graphics used in the application.

The simplest way to maintain control over common graphics is to add them to the master application before the team development stage starts. However, sometimes, common bitmaps need to be imported into more than one application. In this case, here is a suggested methodology.

To work with common graphics:

1. Store all of the graphic files in a single folder. This folder should be a subfolder of the folder that contains the master application (for example, in a subfolder named “Images”).
2. Bring the graphics into the Object Layout using the Import Bitmaps command (File|Import Bitmaps). Select the “Link to File” Image Management option in the Import Bitmaps dialog box.

Each bitmap imported into Rapid brings its name with it, thus ensuring that the bitmap has the same name in all the applications into which it is imported.

❖ *NOTE: Some characters are not legal in RapidPLUS Xpress and they will be replaced with an underscore (_).*

ADDING SUPPLEMENTAL LOGIC (USING RAPIDPLUS)

This section applies to owners of RapidPLUS + RapidPLUS Xpress. Before you add supplemental logic, we recommend that you be familiar with standard RapidPLUS methodology.

In the Xpress mode of operation, logic is added—behind the scenes—each time an object is attached to a mode or a transition is created. The RapidPLUS Xpress designer never has to write a line of logic, nor does the designer see the logic.

If the RapidPLUS developer wants to add more sophisticated logic, standard RapidPLUS mode can be used at any time.

To switch to the standard RapidPLUS mode of operation:

1. In the Application Manager, choose Options|Xpress mode to deselect it.

Adding Logic

In the standard RapidPLUS mode of operation, the logic entered via Xpress mode appears as read-only logic.

These read-only logic lines appear at the beginning of their appropriate sections; that is, entry activities appear at the top of the entry activities section and exit activities appear at the top of the exit activities section.

This behavior is designed to allow the RapidPLUS developer to override the logic automatically written by RapidPLUS Xpress in order to create the necessary behavior. The RapidPLUS developer must be careful when adding logic so that the read-only behavior is not invalidated.

Example of how supplemental logic can invalidate Xpress logic

The RapidPLUS Xpress designer attached Bitmap1 to ModeA. The RapidPLUS developer later writes a logic line in the entry activities of ModeA:

```
bitmap1 hide
```

Result: When the application is run in the Prototyper, the bitmap will not show when ModeA is entered, even though it is attached to the mode. For the RapidPLUS Xpress designer, this behavior may be confusing.

Using Logic That is Accessible in Standard RapidPLUS Only

For some objects, there are functions that can only be accessed in standard RapidPLUS mode of operation.

Examples

In RapidPLUS Xpress, you can create a transition using an event from a user object (**.udo*), but you cannot call an exported function of the user object when Rapid is set to Xpress mode of operation. For such functionality, you must switch to standard RapidPLUS mode of operation. The same rule applies to ActiveX objects and JavaBean objects.

The text display object is available from the Object Palette of RapidPLUS Xpress. In Xpress mode, you can use this object as an advanced label, setting its colors and text entry type. However, to change the text that appears in the text display during runtime or to manipulate the cursor, you must switch to standard RapidPLUS mode.

Index

A

adding

- additional logic 125
- background frame 17
- exported events 122
- imported bitmaps 15
- label objects 20
- logos 24
- modes 13, 43
- notes to states and transition triggers 109
- pushbutton objects 29
- sound objects 22
- timer objects 32
- transitions 28
- user objects 123

aligning objects 49, 68, 75

ancestor attachments

- deactivating 96

animation objects 93

Application Manager 10

applications

- running in the Prototyper 37
- saving 14
- team development 117–125
- workflow for developing 86

attaching objects to mode tree

- how to 18, 25, 90, 92
- timer objects 33

attachments

- behavior of 92
- cutting, copying, and pasting 94
- deactivating 27, 96
- deactivating ancestor attachments 96
- duplicating 94

- hiding 94
- moving in the mode tree 64, 93
- selecting all instances of 95
- selecting in the mode tree 94

B

background frame

- purpose 15, 17, 115

background graphics 14, 115

backlight effect 56

bitmaps

- importing 15
- making transparent 24

blink effect 51, 55

bringing objects to front 23

C

centering objects 75

changing

- font for label objects 21
- object colors 29, 56
- order of modes 35, 114

Chart Tool. *See* State Chart

charts

- screen transitions. *See* screen transition charts
- single state screen transitions. *See* single-state screen transition charts

collapsing branches in a mode tree 46

Colors Edit dialog box 29, 56

concurrent modes 42

copying objects 58

creating

- screen transition charts 18, 108
- transitions 30, 99
- user objects 119

D

- deactivating
 - ancestor attachments 96
 - attachments 27, 96
- default mode 28
- deleting transitions 31, 100
- duplicating
 - attachments 94
 - graphic objects 74
 - nongraphic objects 51

E

- editing
 - modes 28
 - object colors 29
- effects
 - backlight 56
 - blink 51
- esimstd.vss* 6
- event objects 93
- exclusive modes 28, 42
- expanding branches in a mode tree 47
- exported events 122
- exporting transition charts
 - for user objects 124
 - from the Mode Tree 36, 108
 - from the State Chart 18, 108
 - with subtree charts 81, 103

F

- fonts
 - changing for label objects 21
 - saving font settings in *rapidx.ini* 38

G

- graphic objects
 - attaching to mode tree 90–91
 - behavior when attached 92
 - definition 14
 - used in transitions 30

H

- hiding attachments 94

I

- images

- displaying triggers in transition charts 62
 - importing 15
 - used during team development 125
- importing bitmaps 15, 125

L

- label objects
 - adding to layout 20
 - changing font 21
- launching RapidPLUS Xpress 10
- legend page 19, 110
- logic
 - adding logic in standard RapidPLUS mode 125
 - read-only 126
- logic information pane 98, 111

M

- making transitions 30, 99
- Microsoft Visio 6, 18, 87, 108, 112
- mode of operation
 - standard RapidPLUS 125–126
 - Xpress mode 10
- Mode Tree tool
 - collapsing branches 46
 - deactivating attached objects 27
 - defined 86
 - examples of trees 42
 - expanding collapsed branches 47
 - exporting transition charts 36, 108
 - making transitions 30
 - purpose 4, 11–12
 - rearranging modes 35, 114
 - selecting attachments 95
- modes
 - adding 13, 43
 - attaching nongraphic objects 33
 - attaching objects 18, 25, 90–91
 - concurrent 42
 - default 28
 - definition 12, 86
 - editing 28
 - exclusive 28, 42
 - parent 13, 28
 - root 13

moving
 attachments 64, 93
 modes 35, 115
 objects on the layout 49

N

naming
 objects 21
 objects used in transitions 50, 98
 nongraphic objects
 attaching to mode tree 33, 92
 behavior when attached 92
 definition 20, 89
 duplicating 51
 making transitions with 100
 sound objects 22
 timers 33
 viewing in Nongraphic Objects dialog box 23

Nongraphic objects icon 23

notes

adding to states 109–110
 adding to transition triggers 111

O

Object Layout

adding frame objects 17
 adding label objects 20
 adding pushbutton objects 29
 attaching objects to mode tree 18, 90
 importing bitmaps 15
 making transitions 30
 purpose 3, 11
 Select All Attachments command 95
 sending objects to back of other objects 57
 window described 15

Object Palette 15, 88

objects

aligning 49, 68
 attaching to mode tree 18, 25, 90–92
 available in Object Palette 88
 bringing objects to front 23
 centering 75
 changing colors 56
 copying 58
 duplicating 74

graphic and nongraphic 20
 graphic, defined 14
 making colors transparent 29
 naming 21
 pasting 58
 selecting multiple objects 48

opening

applications 41
 Parameter Pane 22
 RapidPLUS Xpress 10
 RapidPLUS Xpress tools 11

P

Parameter Pane 21–22
 parent mode 13, 28
 pasting objects 58
 Prototyper
 overview 5
 running applications 37
 tracing active modes 37
 pushbutton objects
 adding to layout 29
 making transparent 29

R

RapidPLUS products 8
 RapidPLUS Xpress
 launching 10
 overview 1–8
 tools 3, 11
 workflow for developing applications 86

rapidx.ini 38

read-only logic 126

root mode 13

routing of states 34, 114

running applications in the Prototyper 37

S

saving

applications 14
 screen transition charts 112
 system settings 38

schemes in screen transition charts 103, 106–107
 screen transition charts

See also single-state screen transition charts
 creating 18

- screen transition charts (cont.)
 - designing 20, 34, 115
 - displaying triggers as images 62
 - distributing over the Web 113
 - examples 7, 27, 36, 54–55, 61–62, 72, 82, 101
 - exporting 18, 108
 - exporting with subtree charts 81, 103
 - generating from user objects 124
 - in Microsoft Visio 112
 - modifying appearance of 102
 - notes for states and triggers 109–111
 - overview 6, 87
 - printing 113
 - saving 112
 - schemes 107
 - transitions to ancestor mode 55
 - uniformity 115
- selecting
 - attachments 94–95
 - multiple objects 48
- sending objects to back of a specified object 57
- single-state screen transition charts
 - definition 55, 104
 - examples 8, 55, 83–84
 - exporting 108
 - modifying appearance of 105
- sound objects 22, 93
- standard RapidPLUS mode of operation 125–126
- State Chart
 - deleting transitions 31
 - exporting transition charts 18, 108
 - fitting chart to window 53
 - options for transition charts 62, 102
 - overview 4
 - purpose 11
- states
 - adding notes 109–110
 - definition 12, 89
 - designing for transition charts 20
 - sample state in a screen transition chart 90
- T**
- team development 117–125
 - using common graphics 125
 - with user objects 119
 - without user objects 117
- text display object 126
- timer objects
 - adding to layout 32
 - attaching to mode tree 33
 - used in transitions 93
- tracing application behavior 37
- transition charts. *See* screen transition charts
- transitions
 - definition 28, 97
 - deleting 31, 100
 - displaying as images in transition charts 62
 - graphic vs. nongraphic objects 30
 - how to make 30, 99–100
 - naming 50, 98
 - overview 6
- transparent colors
 - bitmap 24
 - pushbuttons 29
- triggers
 - adding notes 111
 - definition 28, 97
- U**
- user objects
 - adding exported events 122
 - building 119
 - definition 117
 - development methodology 123
 - generating screen transition charts 124
 - in standard RapidPLUS 126
 - keypad 123
- User_Object_Template.udo 119, 121
- V**
- viewing application behavior 37
- Visio *See* Microsoft Visio
- W**
- WaveAudio objects 93
- X**
- xpress_onStart_ev 121
- xpress_onStop_ev 121
- xpress_start 121
- xpress_stop 121