Opal®

Getting Started

Version 2.12
Contents

Chapter 1: Introduction

Welcome to Opal ......................................................... 1-1
  Who Should Read this Guide .................................... 1-2
  What You Need to Know ......................................... 1-2
  What this Guide Provides ........................................ 1-3
About Opal .................................................................. 1-4
  Modernizing Legacy Applications ............................. 1-4
  Opal Features ........................................................ 1-5
Conventions ............................................................... 1-9
Getting Help ............................................................. 1-11
  The Main Help Window ............................................ 1-11
  Context-Sensitive Help ............................................ 1-12
  Online User Guide .................................................. 1-12
  Internet Resources .................................................. 1-13
  Contacting Technical Support .................................... 1-13
Chapter 2: Before You Install

Opal Integrator ......................................................... 2-1
Opal Server ................................................................. 2-2
Opal Player ................................................................. 2-2
Installation Methods .................................................... 2-3
Opal Project Folders and Files ........................................ 2-4
Opal Objects ................................................................. 2-6
Naming Objects ........................................................... 2-6
Before You Integrate Host Applications ................................ 2-7
Before You Integrate Databases ........................................ 2-7

Chapter 3: Opal Integrator

Installing Opal Integrator .................................................. 3-1
  Opal Installation Tips .................................................. 3-2
  Attended Installation .................................................. 3-3
  Unattended Installation ................................................. 3-4
Using Online Documentation .......................................... 3-5
Setting Your Opal Preferences ......................................... 3-6

Chapter 4: Opal Server

Opal Server Overview .................................................. 4-1
Installing Opal Server .................................................. 4-3
  Attended Installation .................................................. 4-3
  Unattended Installation ................................................. 4-4
Uninstalling Opal Server ................................................ 4-6
Chapter 5: Opal Player

Opal Player Overview .......................................................... 5-1
Installing Opal Player ......................................................... 5-4
Atended Installation .......................................................... 5-5
Unattended Installation ....................................................... 5-8
Opal Player Proxy Server Setup ........................................ 5-9
Using Opal Player ............................................................... 5-11
Running Opal Internet Applications .................................. 5-11
Running Opal Network Applications ................................. 5-12
Active X and Internet Explorer 3.02 ................................. 5-14

Chapter 6: Opal Integrator Tours

Tour 1: Behaviors and Properties ........................................ 6-2
Creating a New Project ....................................................... 6-3
Setting Up the Page .......................................................... 6-8
Adding an Image and Arranging Objects ............................ 6-12
Adding Behaviors to Objects .............................................. 6-16
Tour 2: Database Integration ............................................... 6-21
Creating a New Project ....................................................... 6-22
Adding Database Fields ..................................................... 6-23
Adding Controls and Filtering Data .................................... 6-30
Inserting, Deleting, and Updating Database Records ............ 6-36
Tour 3: Host Integration ....................................................... 6-37
Creating a New Project and Adding Additional Pages .......... 6-38
Creating an Opal Host Session .......................................... 6-40
Adding Data and Controls to Pages ..................................... 6-52

Glossary

Index
Welcome to Opal

Welcome to Opal—the best way to modernize legacy applications while seamlessly integrating systems across multiple platforms and architectures. With Opal, you can apply a creative, easy-to-use, modern user interface and deploy applications to the Web—without changing proven business code. Opal increases the value of your existing applications by extending their useful life and streamlining business processes.

Opal is an integral part of Computer Associates’ Harmony family, a comprehensive information management strategy that allows organizations to exploit new technologies while integrating existing applications and data sources.

Opal has three basic components:

- **Opal Integrator** is the development tool you use to create your applications. Opal Integrator uses a drag-and-drop approach to integrate legacy applications and databases.

- **Opal Server** moves host and database communications off the client, making the application available to a network server, intranets, or the Internet.

- **Opal Player**, the run-time client for Opal, “plays” the published applications.
Welcome to Opal

Who Should Read this Guide

This book is intended for use by three groups of Opal users:

- **Application Developers**, who will design, test, publish, and deploy Opal projects
- **System Administrators**, who will configure Opal Server
- **System Analysts**, who will set up Opal Player

What You Need to Know

This book assumes you have used Microsoft Windows 95 or Windows NT.

- If you will use Opal Integrator to design applications in which you will integrate host or database information, you should have a background in the appropriate host and database systems. Some knowledge of multimedia file formats, such as .AVI and .WAV, and graphic formats such as .BMP, .GIF, .JPG, and .TIF is also helpful.

- If you will configure Opal Server for host and database communications, you should have a basic understanding of socket communications and the TCP/IP transport protocol. You will need to know whether your network is protected by a firewall or proxy server, and the IP addresses of Opal Integrator and Opal Player clients to allow access to Opal Server. You will also need to know how to install and configure a Windows NT service.

- If you will use Opal Player with either Microsoft Internet Explorer or Netscape Navigator, you must be familiar with the appropriate Web browser and ensure that the corresponding Opal browser plug-in is properly installed. If you will use the stand-alone Opal Player, you must ensure that it is properly installed. In all cases, Opal Player must be installed on each client computer.
What this Guide Provides

This book is designed to familiarize you with the features, concepts, and objectives of Opal. It is organized into three groups of chapters: Overview, Installation and Setup, and Touring. This book also includes a glossary of terms commonly used in this and other Opal documentation.

Overview

Chapter 1, “Introduction” provides an overview of this book and the features of Opal.

Installation and Setup

Chapter 2, “Before You Install” summarizes the system requirements you must meet before installing the Opal components on your computers.

Chapter 3, “Opal Integrator” provides an overview of Opal Integrator and the files and data structures created during installation.

Chapter 4, “Opal Server” provides an overview of Opal Server, its setup, and how it interacts with Opal clients.

Chapter 5, “Opal Player” provides an overview of Opal Player and how it works.

Touring

Chapter 6, “Opal Integrator Tours” guides you through a tour of Opal’s features. In the tour, you will use Opal objects and host and database integration to design a model Opal application.
About Opal

Modernizing Legacy Applications

Opal’s modern user interface employs animation, sound, video, and graphics to energize and simplify applications. With the attractive templates, a library of objects, and wizard-like Opal Assistants, even beginning developers can create problem-free user interfaces that are both effective and visually appealing.

When you design Opal applications, you will begin at the user interface and work inward, using existing applications and information sources to build custom applications that meet current and evolving needs. Smart tools (for powerful code-free interface design) and Opal’s Data Mapper (to automate data integration) let you modernize existing applications quickly, easily, and affordably.

The Opal applications you create will simplify user access to enterprise-class applications. Opal improves user satisfaction and productivity while lowering training and support costs.

Opal Integrator runs under Windows 95 or Windows NT and provides intuitive and graphical methods for creating new applications. The applications you create can:

- **Integrate external data sources**, such as databases and mainframe applications.
- **Incorporate multimedia**, such as videos, animations, and sounds.
- **Provide intuitive interfaces** that require little or no training for users.

1-4 Opal Getting Started
About Opal

Opal Features

Opal Integrator

A brief introduction to some of the features of Opal integrator follows. For a more complete introduction to Opal Integrator and details about these and other important features, see the online User Guide.

Code-Free interface creation
Opal matches a point-and-click interface with drag-and-drop efficiency to make creating modern user interfaces simple and fast.

Opal Assistants
Opal Assistants are wizard-like utilities that guide you through creating and configuring Opal applications.

Opal Data Mapper
The Data Mapper automates data integration, making it easier to integrate data sources with an application. Simply drag data formats from the Data Mapper to the application, then use the Opal Assistants to ensure the data is incorporated into the application appropriately.

SmartPipes
SmartPipes move information between screen objects and data sources without programming.

Opal Objects
Opal includes a rich set of smart, configurable interface objects—edits, buttons, radio buttons, list and combo boxes, scrolls bars, tables, folders, and menus. Objects are fully customizable, so you can assign properties and behaviors that control their appearance, the data they store, and how they perform under specific conditions.

Scripting
You can use OpalScript, VBScript, or JScript to write simple expressions for property values and action parameters. Opal also contains a number of specialized VBScript functions that facilitate complex scripting tasks.
About Opal

Interactive Multimedia

You can add sound, video, animation, and images to your Opal applications to make them more attractive and easier to use. You can also add multimedia content to some Opal objects to increase their functionality.

Cross-Platform Support of Information Sources

Opal supports a variety of information sources, including 3270-, 5250-, VT220-, and HP700-compatible hosts, and ODBC-compliant relational databases.
Opal Server

Opal Server allows clients to access commonly used ODBC drivers from a shared location. The network administrator can set them up once, in one place, instead of installing them separately on individual computers. Opal Server also provides a central point for all communication with information sources. Specific Opal Server features include:

- Telnet and ODBC connectivity to data sources, eliminating the need for client-side emulators and ODBC drivers
- Web browser access to host and database information over intranets, extranets, or the Internet
- TCP/IP socket connections between Opal client and server, providing high performance and security
- Native services implementation, avoiding CGI bottlenecks and security issues
- Support for 40-bit DES encryption of files transferred between Opal Server and clients
Opal Player

After you finish developing and testing the application in Opal Integrator, you can distribute it to users to run on Opal Player. Opal Player can operate in three environments: as a stand-alone Opal Player for Windows 95/98 and Microsoft Windows NT 4.0, as a browser plug-in for Microsoft Internet Explorer, and as a browser plug-in for Netscape Navigator. In all cases, Opal Player works either through Opal Server, or with the local Opal TE connection or ODBC driver. Specific Opal Player features include:

- Web support via Microsoft Internet Explorer or Netscape Navigator/Communicator
- A desktop client for non-Web applications
- Desktop or server-based connectivity (thin or thick client)
- Intelligent caching for reduced network traffic
Conventions

All Opal documentation observes the following conventions to simplify locating and identifying information:

**Code Examples**

OpalScript, VBScript, or JScript syntax is represented in bold monospaced type. Arguments are represented in italics. For example:

```make
MAKEINT(string)
```

**Cross References**

The following conventions are used to refer to information elsewhere in the documentation:

- The book or online resource name in italics:
  
  See *Opal Online Help* for more information.

- The chapter name in double quotes:
  
  See Chapter 2, “Before You Install” for more information.

- Section name as it appears in the book:
  
  See Getting Help in this chapter.

**Important or New Terms**

Terms that are new or important to your understanding of Opal are represented in italic type. These terms are included in the glossary at the end of this book and in the glossary in the online *Opal User Guide*.

**Key Names and Key Combinations**

The names of keys, such as ENTER, CTRL, and DEL, appear in small capital letters.

When two key names are joined by a plus sign (+), for example, CTRL+S, you should hold down the first key in the sequence while you press the second key to complete the command. Release the second key first.

**Menu, Command, Option, and Field Names**

Menu, command, option, and field names are represented in bold type, as they appear on the screen. For example, the **Object** menu, or the **Configuration filename** field.
## Conventions

<table>
<thead>
<tr>
<th>Object Names</th>
<th>Opal object names are represented with initial capitalization. For example, the Button object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties, Messages, and Actions</td>
<td>In general, properties, messages, and actions and the categories that contain them are represented in bold type. For example, the Latching property in the Button property category. When a property, message, or action is discussed in relation to its category, periods separate the object name on the left from the category name in the middle and the property, message, or action name on the right. For example, Button.Button.Latching. In Opal, messages and actions are combined to form behaviors. Behaviors are also referred to as message/action pairs. Message/action pairs are represented in bold type. A slash separates the message name on the left from the action name on the right. For example, Button Activated/Launch Program. The object for which a behavior is defined may precede the message/action pair, followed by a period. For example, Button.Button Activated/Launch Program.</td>
</tr>
<tr>
<td>Window and Dialog Box Names</td>
<td>Window and dialog box names are represented with initial capitalization. For example, the Properties window.</td>
</tr>
<tr>
<td>Other Conventions</td>
<td>Throughout the Opal documentation, the term user describes the person who will eventually use the Opal applications you are developing. When we refer to you, we mean the developer of the Opal application or the administrator of Opal Server or Opal Player. Throughout this book, the term Opal is used to mean Opal Integrator. Opal Player and Opal Server are always discussed specifically.</td>
</tr>
</tbody>
</table>
Getting Help

Opal Integrator includes extensive online help. You can open the main help window to look for information, or you can display context-sensitive help for a specific menu command, property, message, or action. Opal also includes an online User Guide, which offers more detailed information than the online help.

The Main Help Window

Use the following procedure to gain access to online help in the main help window:

1. From the Help menu, select Contents or Search. An Opal Integrator Help window opens.
   - To browse the contents of the help, click the Contents tab.
   - To search the help index for specific subjects, click the Index tab and type the subject for which to search in the field provided.
   - To search for specific text in the help, click the Find tab and type the text for which to search in the field provided.

   **Note:** The first time you select the Find tab, the Find Setup Wizard appears. If you accept the default setting, help topic titles and graphic captions are indexed for the Find tab. If you select another setting, more of the text in the help topics is indexed, but performance is reduced. Click Next to register your selection, then click Finish to index the help topics according to your selection. The Find tab appears.

2. Close the Help window.
Getting Help

Context-Sensitive Help

Use the following procedure to gain access to context-sensitive online help:

1. Select the menu command, field, property, message, or action for which you want information.

2. Press F1, or right-click the selected item and select What’s This? from the shortcut menu. The help for the selected item appears.

Opal dialogs with a icon allow you to select this icon and then click the control for which you want help. To bring up the general dialog help, click the icon and then click the dialog title bar.

Note: If you select a category from the Properties window, then press F1, the Opal Integrator Help Index appears with the selected topic highlighted.

Online User Guide

The online Opal Integrator User Guide contains detailed information on how to use Opal to perform specific tasks, and includes reference information on all Opal features and functions. For information on how to gain access to the Opal Integrator User Guide and the online version of the Opal Getting Started, see Using Online Documentation in Chapter 3, “Opal Integrator.”

Note: Online documentation requires Adobe Acrobat Reader 3.01 or later. If Reader 3.01 or later is not installed on your computer, you must install it. Reader 3.01 is included on the Opal CD-ROM in \Acrobat\ar32e301.exe (this is a self-extracting installation routine). You can download the latest release of Adobe Acrobat Reader at no charge from http://www.adobe.com/
Internet Resources

Opal is supported through the Computer Associates Total Client Care (CA-TCC) system. CA-TCC provides real-time World Wide Web access to Computer Associates’ comprehensive support system, where you can:

- Open new support issues.
- Communicate with Computer Associates Technical Support about existing issues.
- Download Product Temporary Fixes (PTFs).
- Download Product Information Bulletins (PIBs).
- Search for problems, information, and solutions.

First time CA-TCC users must register using the online registration form at [http://webtrack.cai.com](http://webtrack.cai.com)

For more information on CA-TCC, or to submit questions to CA-TCC, refer to [http://www.cai.com/catotalclientcare.htm](http://www.cai.com/catotalclientcare.htm)

For a full description of Computer Associate’s support options, refer to [http://www.cai.com](http://www.cai.com)

Information about Opal is available at [http://www.cai.com/products/opal.htm](http://www.cai.com/products/opal.htm)

Contacting Technical Support

Before you contact technical support consult the “Troubleshooting” chapter in the Opal Integrator User Guide or in the Opal online help system. “Troubleshooting” will help you either independently resolve an issue or collect information that will help technical support resolve it.

Opal support contact information is available at [http://www.cai.com/products/opal.htm](http://www.cai.com/products/opal.htm)
Getting Help
Chapter 2

Before You Install

This chapter discusses information you will find helpful as you install, configure, and begin to use Opal.

Opal Integrator

Opal runs under Windows 95 or Windows NT 4.0, and requires a Pentium processor with 64MB RAM and 200MB of disk space.

If you will integrate database information into your Opal applications, you must know where the database is located and how you will connect to it. Opal can access databases through a local ODBC-compatible driver, or over a TCP/IP connection to Opal Server and a database on a remote computer. During application design, a local ODBC-compatible connection may yield better performance. Consult your computer support staff for connection information.

If you will integrate host information into your Opal applications, you need the following hardware and software:

- **Physical link to host computer**—If you can connect to host applications from your computer, this link already exists. If you need help setting up this link, contact your computer support staff.

- **Communications layer**—If you have access to a host application from your computer, a communications layer already exists. You can use a third-party WinHLLAPI communications layer or native Opal TE.
Opal Server

Opal Server runs under Windows NT Server 3.51 or 4.0, and requires at least a Pentium 200 with 128 MB of RAM. The server files require slightly more than 2 MB of disk space to install, and a running server requires at least 10 MB of disk space for log files, ODBC buffers, and other temporary files. TCP/IP must be installed on the server computer before you install Opal Server. You must have administrative rights on the server computer in order to perform the installation.

If you are using Opal Server to connect to host applications, the host computer and host TCP/IP port must be accessible from the Opal Server computer.

Opal Player

Opal Player can operate in three environments: as a stand-alone Opal Player for Windows 95/98 and Microsoft Windows NT 4.0, as a browser plug-in for Microsoft Internet Explorer, and as a browser plug-in for Netscape Navigator. The CD for Opal Player contains all the components needed to set up Opal Player for all three environments. The Opal Player executable, OPALPLAY.EXE, plays published Opal applications. The Dynamic Link Library NPOPAL32.DLL, provides the browser Internet plug-in support.

The Opal Player must be installed on each client computer. The Opal Player desktop client requires 16MB RAM; Opal Player browser clients require 32MB RAM. Opal applications may be installed on the client computer, a network drive, or on an HTML server for Web-style deployment. A large number of users, even users on different Windows platforms, can run the same Opal application simultaneously.
Installation Methods

You can use either of two methods to install Opal Integrator, Server, and Player:

- **Unattended installation**, in which you preconfigure the installation program and run it with a single command. The Opal component is installed without prompts or user interaction, as defined in the configuration file.

- **Attended installation**, in which you run the installation program and specify the installation path and program folder in response to prompts.

If you will install Opal Integrator, Server, or Player on a single computer, you will probably use an attended installation.

However, if you will install the Opal component on multiple computers, you can save a significant amount of time with an unattended installation. For example, to install Opal Integrator on each of 20 computers using an attended install would take 40 minutes or more. With unattended installation, you could type a single command on each computer and walk away—your preconfigured installation would do all the work, with no further intervention necessary. In fact, if your computers are on a LAN, you could write a simple script to install your Opal components on all 20 of the machines with a single command.


**Note:** You or your technical support staff can adapt the unattended installation procedures for various situations. For example, you could write a script that installs Opal Integrator to a number of computers on a LAN from the server, or you could create a self-extracting executable file that automatically performs an unattended install when a user runs it.
Opal Project Folders and Files

By default, Opal Integrator stores all Opal projects under the Projects folder. The Projects folder is located two levels below the Opal Integrator installation folder; for example, Opal\Program\Projects.

When you start a new project, Opal automatically creates a new folder for it. Opal uses the project name and location you specify (for example, Project1) to name the new folder. Opal also creates a project file (for example, Project1.PRO). The project file contains the Opal project settings. The following illustration shows a typical Opal Integrator folder structure:

As the illustration shows, Opal creates several folders within each Projects folder. They are:

- **content**—Some Opal objects require content created outside Opal, such as sounds, images, videos, and animation. Opal stores such content in files in the content folder. The content folder also contains .HSD (host session data) files created with the Data Mapper.

- **deleted**—Opal places pages you remove from a project in the deleted folder.
Opal Project Folders and Files

- **pages**—The *pages* folder contains .SDT (Opal page files in development) and .GOT files (Opal content files in development). Opal automatically creates a GLOBAL.SDT and a GLOBAL.GOT file in the *pages* folder for each project. Opal uses these files to store global objects and information used in the project.

- **preview**—You can isolate testing from development by using the *preview* folder as a temporary location for a published Opal project. Typically, you can deploy a project to the *preview* folder (as the target directory) and continue fine-tuning and republishing the project.

- **publish**—The *publish* folder contains the run-time files created when you publish the Opal project:
  - .PRP files, which are published Opal project files. These files are used to launch Opal applications.
  - .SDO files, which are published Opal page files.
  - .GO3 files, which are published Opal content files.
  - .HTM files, which are published Opal web pages. These files are used to launch Opal Internet applications.

Opal Player uses the files in the *publish* directory to run Opal applications.
Opal Objects

**Opal Objects**

*Objects* are the basic components of an Opal application. They can:

- Display information
- Accept user input
- Contain other objects
- Perform actions on other objects
- Integrate data from legacy applications or databases
- Display images, or play animations, videos, or sounds

*Properties* and *behaviors* define Opal objects. Properties specify the appearance, position, and other values associated with the object. Each behavior specifies an *action* the object performs when it receives a *message* from the user, another object, or Opal. Behaviors are also known as *message/action pairs*.

You can create all of the Opal objects manually, but Opal Assistants and the Data Mapper help create and configure many of them.

**Naming Objects**

Observe the following conventions when naming Opal objects:

- Object names can contain any alphabetic or numeric character.
- Object names cannot contain spaces. Opal removes spaces at the beginning and end of the name, and replaces spaces within the name with underscores (_).
- Object names cannot contain the following characters: ! $ % ^ & * ( ) [ ] \ | ; ': " < > , . ? / - # ~ @
- If you will reference a *compound* object’s name in VBScript, do not begin the name with a number. Doing so causes the VBScript error, “*Invalid number in line x.*”
Before You Integrate Host Applications

Before you integrate a host application into Opal, you should be familiar with the host application, how it is currently being used, and what features you will integrate. You will need to know the IP address or IP name of the host; the connection port number; the terminal type (if you are connecting to a 3270 host), and the language used in the host application. You should also be familiar with how Opal records and identifies host application screens and data elements.

Before You Integrate Databases

Before you integrate a database into an Opal application, you should have a basic understanding of relational databases, including such terms as field, row, column, record, table, query, and result set. An understanding of SQL is helpful but not necessary, unless you use advanced functions in Opal.

Before you integrate a specific database, you should be familiar with the database schema so you can identify the correct elements of the database. You should also understand how the database is currently being used.
Installing Opal Integrator

As explained in Installation Methods in Chapter 2, “Before You Install,” you can install Opal Integrator interactively or in the background.

Whether you choose attended or unattended installation, Opal Integrator is installed using the directory structure shown below. In this case, the installer specified C:\Opal as the destination directory, either at the prompt or in the silent.cfg file.

- The Assist folder contains a subfolder for each Opal Assistant.
- The Themes folder contains a subfolder for each theme shipped with Opal.
- The Tutorial folder contains files and data for the Opal Tutorial, discussed later in this book.
Installing Opal Integrator

Opal Installation Tips

Before beginning the installation of Opal Integrator or any Opal component product, make sure all application programs are closed. If you encounter problems in the Opal setup routine, try the following:

- Reboot the computer with a minimal configuration, then begin the Opal setup routine again. Your systems administrator can advise you on how to boot up minimally for your version of Windows. For example, you may need to bypass your startup files, boot up in Safe Mode, disable network support, and so on. Once Opal is installed successfully, you can boot your computer in the normal manner.

- If you are still encountering Opal setup errors or if setup hangs, call Opal technical support for assistance.
Attended Installation

If you have a previous version of Opal Integrator installed on your computer, it is recommended that you uninstall it first, before installing the new release. Uninstalling removes old program files and deletes outdated Registry entries. To uninstall a previous release of Opal Integrator, use the Add/Remove Programs icon located in the Windows Control Panel. Uninstalling will not delete the Opal Integrator folder or the contents of the Projects folder.

In most cases, Opal Integrator installation will start automatically after you insert the Opal Integrator CD into the drive. To start installation manually from Windows Explorer, double-click setup.exe located on the Opal Integrator CD. During an attended installation, the Setup Wizard automatically collects information about your computer’s configuration. Most of the installation is automated. However, the Setup Wizard will prompt you to accept or specify customized installation values. The following table lists the information for which you will be prompted.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Directory</td>
<td>The drive and path on which you will install the software. You can accept the default or browse and specify an alternate.</td>
</tr>
<tr>
<td>Program Folder</td>
<td>The folder into which Setup will add program icons. You can accept the default, select an existing folder from the list, or specify a new folder.</td>
</tr>
</tbody>
</table>
Unattended Installation

Use the following procedure to configure and run an unattended Opal Integrator installation. You can install Opal Integrator on computers running Windows 95 or Windows NT 4.0.

1. Copy the intgrtr folder and its contents from the Opal Integrator CD-ROM to your hard drive.

2. Use a text editor such as Notepad to open silent.cfg from the intgrtr folder on your hard drive. The contents of silent.cfg are as follows:

   [TargetMachine]
   InstallToPath=D:\Opal Integrator Silent
   ProgramFolder=Opal Integrator Silent

3. Change the InstallToPath and ProgramFolder values as necessary.
   - InstallToPath is the full path to which you want Opal Integrator installed.
   - ProgramFolder is the Start Menu Programs folder in which the Opal Integrator executables are placed.

   For example, to install Opal Integrator in the path C:\Opal and use the program folder Opal 2.12, you would change the parameters as follows:

   [TargetMachine]
   InstallToPath=C:\Opal
   ProgramFolder=Opal 2.12

4. Save your changes.

5. From the Windows Run dialog box, specify drive:\intgrtr\setup.exe -s, then click OK to begin the unattended installation of Opal Integrator. For example, if you copied the intgrtr folder to your C: drive, you would type the following in the Run dialog box:

   C:\intgrtr\setup.exe -s

   The installation program installs Opal Integrator as a background process, without user intervention.
Using Online Documentation

The Opal program CD-ROM contains this document and the Opal User Guide in online format. They are located in the \DOC directory, in PDF format. For your convenience, the Adobe Acrobat Reader is included in the \Acrobat directory. To install Acrobat Reader, run AR32E301.EXE, which is a self-extracting installation program. These files are located in the \DOC directory on the Opal CD-ROM:

- Opal Getting Started.PDF  Opal Getting Started
- Opal Integrator User Guide.PDF  Opal Integrator User Guide

After you install Adobe Acrobat Reader, you can read an online book by double-clicking its file name. You can read the online books from the Opal CD-ROM, or you can copy the .PDF files to your computer and read them from there.
Setting Your Opal Preferences

You can set preferences to customize aspects of how Opal responds during application design. The preferences you set only affect how you interact with Opal Integrator. These preferences do not directly affect the application you are developing. Use the following procedure to set Opal preferences:

1. From the **Tools** menu, select **Preferences**. The Preferences dialog box appears.

2. Click the **Workspace**, **Grid**, **Objects**, **Host Mapper**, or **Assistants** tab, then set the appropriate options. For information about specific preferences, refer to the Online Help or the online Opal User Guide.

3. To save the preferences and close the Preferences dialog box, click **OK**.
Chapter 4

Opal Server

This chapter describes Opal Server installation and configuration. Opal Server is distributed separately from the Opal Integrator product, and the information is included here for reference only.

Opal Server Overview

Opal Server provides a central point for all communication with information sources. To facilitate this, Opal Server stores all ODBC drivers and other connectivity software in a shared location. The network administrator can set them up once, in one place, instead of installing them separately on individual computers.

Connections to and from Opal Server are accomplished via sockets. The socket client (Opal Integrator, Player, or a browser plug-in) connects to Opal Server using the server’s IP address and port address. Opal Server’s default port address is 6725 (which spells Opal on your telephone keypad), but you can set up Opal Server to communicate at any port, based on the your network and internal security requirements. If you are using Opal Server for a web application, you may set it up at port 80, which is the port the World Wide Web uses. This way, anyone who can browse the web can connect to your Opal Server. When the client finishes using server resources, it ends the session.

Important! If you use port 80 for Opal, make sure to run your Web server at a different IP address so it does not conflict with Opal Server.
Because socket sessions only run on networks that support the TCP/IP transport protocol, Opal Server will not run on networks that do not support TCP/IP.

The same constraints that apply to clients connecting to Opal Server apply to Opal Server when it connects to hosts and database servers. If an Opal Server inside a firewall is to communicate with clients outside the firewall, the firewall must pass TCP/IP packets sent to the Opal Server’s IP address and port address.

Host sessions rely on Opal TE and the telnet protocol, usually available at port 23, to connect to hosts. As long as these hosts can be reached from Opal Server at the telnet port, host connections can be made.

Similarly, an Opal Server must be able to establish an ODBC connection to a database. Different databases may use different communications protocols. Often, this means that the system administrator must ensure that no firewall or proxy server lies between the Opal Server and the database.

**Note:** Opal Server can receive data from two separate network cards in the same computer; this allows computers on different subnets, which cannot communicate directly, to communicate through the Opal Server.
Installing Opal Server

As explained in Installation Methods in Chapter 2, “Before You Install,” you can install Opal Server interactively or in the background.

Attended Installation

During installation, the Setup Wizard automatically collects information about your computer’s configuration. Most of the installation is automated. However, the Setup Wizard will prompt you to accept or specify customized installation values. The following table lists the information for which you will be prompted.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Directory</td>
<td>The drive and path on which you will install the software. You can accept the default (C:\OpalSrv) or browse and specify an alternate.</td>
</tr>
<tr>
<td>Port Address</td>
<td>The port address at which Opal Server will “listen.” The default is 6725.</td>
</tr>
</tbody>
</table>
Installing Opal Server

Unattended Installation

Use the following procedure to configure and run an unattended Opal Server installation. You can install Opal Server on computers running Windows NT Server 3.51 or 4.0.

1. Copy the ServerCD folder and its contents from the Opal Server CD-ROM to your hard drive.

2. Use a text editor such as Notepad to open silent.cfg from the ServerCD folder on your hard drive. The contents of silent.cfg are as follows:

   [TargetMachine]
   InstallToPath=C:\OPALSRV
   Port=6725
   PlayerSecurityMode=0
   IntegratorSecurityMode=0
   PlayerSecurityList=
   IntegratorSecurityList=
   ArchiveFrequency=24
   ConnectionNumber=100
   Drivers=SockConn
   EventLogLevel=3
  LogFileLevel=3
   CryptoProviders=None DES

3. Change the silent.cfg parameter values as necessary.
   - **InstallToPath** is the full path to which you want Opal Server installed.
   - **Port** is the port at which Opal Server listens (that is, Opal Server’s port address).
   - **PlayerSecurityMode** is 1 if the **PlayerSecurityList** specifies the only Player IP addresses allowed access to Opal Server, or 0 if all Player IP addresses are allowed access to Opal Server.
   - **IntegratorSecurityMode** is 1 if the **IntegratorSecurityList** specifies the only Integrator IP addresses allowed access to Opal Server, or 0 if all Integrator IP addresses are allowed access to Opal Server.
Installing Opal Server

- **PlayerSecurityList** lists the IP addresses of Opal Player clients to allow access to Opal Server, separated with spaces.
- **IntegratorSecurityList** lists the IP addresses of Opal Integrator clients to allow access to Opal Server, separated with spaces.
- **ArchiveFrequency** is the number of hours before the log file is archived.
- **ConnectionNumber** is the maximum number of simultaneous connections to allow to the server.
- **Drivers** is a parameter reserved for Opal internal use. Do not change the Drivers setting.
- **EventLogLevel** is the logging level for the Windows NT Event Log. The following settings are available for this parameter. Since this is a Windows NT system log, we recommend that you check with your NT system administrator before setting this parameter.
  - 0 – disables Windows NT event logging for the Opal Server application
  - 1 or 2 – logs nothing
  - 3 – logs errors (default)
  - 4 – logs connects and disconnects
  - 5 – logs opens and closes
  - 6, 7, 8, or 9 – logs everything.
- **LogFileLevel** is the logging level for the Opal Server log file `servman.log`, which is located in the Opal Server installation directory.
  - 1 or 2 – logs nothing
  - 3 – logs errors (default)
  - 4 – logs connects and disconnects
  - 5 – logs opens and closes
  - 6, 7, 8, or 9 – logs everything.
Uninstalling Opal Server

- CryptoProviders is DES if you want only DES-encrypted clients to communicate with Opal Server; None if you want only unencrypted clients to communicate with Opal Server; or None DES if you want to select encryption or no encryption for each connection.

4. Save your changes.

5. From the Windows Run dialog box, specify drive:\ServerCD\setup.exe -s, then click OK to begin the unattended installation of Opal Server. For example, if you copied the ServerCD folder to your C: drive, you would type the following in the Run dialog box:

    C:\ServerCD\setup.exe -s

The installation program installs Opal Server as a background process without user intervention.

Uninstalling Opal Server

Use this procedure to uninstall Opal Server manually. If you are installing an Opal Server upgrade, the upgrade setup routine will automatically uninstall the older Opal Server version for you.

On your Windows NT server, select the Add/Remove Programs option in the Windows NT Control Panel. Select Opal Server Manager, click the Add/Remove button, then click OK. This will stop and remove the Opal Server Manager service. This will not delete the Opal Server file folder.
This chapter describes Opal Player installation and configuration. Opal Player and the Opal Player browser plug-ins are included on the Opal Player CD-ROM.

**Opal Player Overview**

Opal Player can operate in three environments: as a stand-alone Opal Player for Windows 95/98 and Microsoft Windows NT 4.0, as a browser plug-in for Microsoft Internet Explorer, and as a browser plug-in for Netscape Navigator. The CD for Opal Player contains all the components needed to set up Opal Player for all three environments. The Opal Player executable, OPALPLAY.EXE, plays published Opal applications. The Dynamic Link Library NPOPAL32.DLL, provides the browser Internet plug-in support.

The stand-alone Opal Player can run under Microsoft Windows 95, Windows 98, and Microsoft Windows NT 4.0. The Opal Player plug-in for Microsoft Internet Explorer runs under Internet Explorer version 3.02 or later. The Opal Player plug-in for Netscape Navigator runs under Netscape Navigator (Communicator) version 3.03 or later.

*You must install Opal Player on each client machine.* Opal applications may reside on a client machine, a network drive, or on an HTML server for Web-style deployment. A large number of users, even users on different Windows platforms, can run the same Opal application simultaneously.
The following illustration shows the deployment of Opal Player in a network.

When Opal Player is installed, it sets up the environment needed to run your published Opal applications. The Opal Player installation program detects the release of Microsoft Windows running on the target machine, and extracts the proper components for Opal Player. The Opal Player installation program also allows you to select the Opal Player browser plug-ins that you wish to install. The Opal Player browser plug-ins let you run Opal applications on the Internet. The Opal Player installation program sets up a Windows association between the .PRP file type and the Opal Player executable. This association allows you to easily create shortcuts for your Opal applications. The .PRP file is the published Opal project file of an Opal application.
An .HTM file is an Internet Document file (HTML). When you publish an Opal application, an .HTM file is automatically created in the project’s publish folder. This .HTM file contains a published Opal web page suitable for Internet or intranet deployment. There is only one .HTM file for each published Opal application, projectname.HTM. If an Internet browser is installed on your computer, you can easily launch the Opal .HTM file from the browser or from Windows Explorer, provided you have the proper Opal plug-in installed for either Netscape Navigator or Microsoft Internet Explorer.
Installing Opal Player

As explained in Installation Methods in Chapter 2, “Before You Install,” Opal Player comes in three versions, which you can install interactively or in the background.

Stand-Alone Player

Whether you choose attended or unattended installation, the Opal Player executable, OpalPlay.exe, is installed using the directory structure shown below. In this case, the installer accepted the default destination directory, C:\OpalPlay, either at the prompt or in the silent.cfg file.

Player for Microsoft Internet Explorer

Whether you choose attended or unattended installation, the Opal Player NPOPAL32.DLL component for Microsoft Internet Explorer is installed within Internet Explorer’s directory structure.

Player for Netscape Navigator

Whether you choose attended or unattended installation, the Opal Player NPOPAL32.DLL component for Netscape Navigator is installed within Internet Explorer’s directory structure.
Attended Installation

The Opal Player CD contains three Opal Player configurations:

- **Opal Player**—The stand-alone Opal Player for Windows 95, Windows 98, and Windows NT 4.0. It runs as an independent program.
- **Opal Player for Internet Explorer**—The Microsoft browser plug-in. It operates with Microsoft Internet Explorer version 3.02 or higher.
- **Opal Player for Netscape Navigator**—The Netscape browser plug-in. It operates with Netscape Navigator (Communicator) version 3.03 or higher.

If you have a previous version of Opal Player installed, we recommend that you uninstall it first, before installing the new version. Uninstalling removes old program files and deletes outdated Registry entries. To uninstall a previous release of Opal Player, use Add/Remove Programs on the Windows Control Panel. Uninstalling does not delete the Opal Player folder.

In most cases, Opal Player installation starts automatically after you insert the Opal Player CD into the drive. To start installation manually from Windows Explorer, double-click setup.exe on the Opal Player CD.

During installation, the setup program prompts you with the following dialog:

If you select Yes, Opal Player setup will search for installed Internet browsers on your system, and allow you to select the Opal Player browser plug-ins to install. The Opal Player browser plug-ins let you run Opal applications on the Internet.
Installing Opal Player

During installation, the Setup Wizard automatically collects information about your computer’s configuration. Most of the installation is automated. However, the Setup Wizard will prompt you to accept or specify customized installation values. The following table lists the information for which you will be prompted.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Directory</td>
<td>The drive and path on which to install the software. You can accept the</td>
</tr>
<tr>
<td></td>
<td>default or browse and specify an alternate.</td>
</tr>
<tr>
<td>Program Folder</td>
<td>The folder into which Setup will add program icons. You can accept the</td>
</tr>
<tr>
<td></td>
<td>default, select an existing folder from the list, or specify a new folder.</td>
</tr>
<tr>
<td>Add Internet Plug-in(s)</td>
<td>Setup asks whether you want to install Opal Player Internet plug-ins. The Opal</td>
</tr>
<tr>
<td></td>
<td>Player Internet plug-ins allows you to run Opal applications on the Internet.</td>
</tr>
</tbody>
</table>

5-6 Opal Getting Started
Web Server Configuration

If Opal applications are deployed on a Web Server, the following settings must be made on the Web Server machine, if applicable.

- For Web Servers using the Microsoft Internet Information Server (IIS) services, you will need to utilize the provided Registration Entries file, **IIS_Mime.reg**, which is located on the Opal Player CD. Double-clicking this file adds the following two string value entries to the Windows Registry:

  ![Registry Entry](registry.png)

  Windows will inform you when the values are successfully added to the Registry. These entries enable the Web Server to stream .PRP and .SDO file types to the Opal Player client. You must reboot the computer for the configuration to take effect.

- For Web Servers using any **mime.types** configuration files, you will need to add the following line to each file:

  ![Mime Types](mime.png)
Unattended Installation

Opal supports an unattended installation feature that allows a system administrator to install Opal Integrator, Opal Server, and Opal Player without user intervention.

You can set the options to run an unattended installation of Opal Player in `silent.cfg`, which is a configurable file located in the Opal Player installation directory. The Opal Player `silent.cfg` contains the following settings:

```
[TargetMachine]
  InstallToPath=C:\Player Silent
  ProgramFolder=Player Silent
```

To make a change to a default setting, modify the setting and save the file.

After you set up the `silent.cfg` file, use this procedure to run an unattended installation for Opal Player:

1. From the Start menu, select Run. The Run dialog box appears.

2. In the Open field, type “E:\setup” -s, where E: is the path to the installation directory (CD-ROM drive or network directory), then click OK. The Opal Player unattended installation will run.

**Note:** Opal Player for Netscape Navigator and Opal Player for Microsoft Internet Explorer do not have customizable `silent.cfg` files. The Opal Player plug-in components are automatically created in subfolders of Internet Explorer and Netscape Navigator.
Opal Player Proxy Server Setup

Opal Player can be configured to connect to a Web Server through a Proxy server, overriding any browser Connection Proxy server options in effect. A Proxy server sits between a client application and a real server. It intercepts requests to the real server to see if it can fill the requests itself. If not, it forwards the request to the real server. Proxy servers can improve performance and filter requests made by the client.

To configure Opal Player for a Proxy server:

1. Install Opal Player as outlined above. During installation, make sure you add Opal Player's Internet plug-in to your currently installed Internet browser(s).

2. Start Opal Player.

3. Select the Proxy button on the Opal Player – Open Page dialog.

The following Proxy Setup dialog will appear:

4. To enable the Proxy Settings, check the Override Internet Proxy Settings check box.
Opal Player Proxy Server Setup

5. Enter the web address of the Proxy server you are using. For example, www.myproxy.server.com/

6. Enter the Port number of the Proxy server you are using. For example, 80.

Note: If you are using Microsoft Internet Explorer as your browser, and the Override Internet Proxy Settings check box is not checked, Opal Player will use any IE Connection Proxy server options that are in effect.

If you are using Netscape Navigator as your browser, and you wish to connect Opal Player to a Proxy server, you must configure the Proxy Setup dialog as outlined in the above steps. Opal Player will not use any Netscape Navigator Connection Proxy server options.
Using Opal Player

After you install Opal Player, you can use it to run published Opal applications on the Internet or on your network.

Running Opal Internet Applications

If your Opal application is located on an Internet or intranet Web Server, you can easily launch it from your Web browser, provided you have installed the Opal Player Internet plug-in for your browser.

Open your Internet Web browser, and type or select the URL of the Opal application. Once loaded, the Opal application starts in its own window.
Using Opal Player

Running Opal Network Applications

If your Opal application is located on the network, the Opal Player can run the application in several ways, depending on your requirements.

- Click the Windows Start button and point to Programs. Select Opal Player from the Programs menu. Opal Player starts, then displays the Opal Player – Open Page dialog box. Browse to the appropriate directory, then select the .PRP file (published Opal project file) you want to open.

- You can also start Opal Player from Windows Explorer or the Start button Run menu by browsing to the OpalPlay.exe file and executing it. When started using this technique, Opal Player displays the same dialog box illustrated above.

- Because the Opal Player installation program sets up a Windows association between the .PRP file type and the Opal Player executable, you can easily create shortcuts to start Opal applications located on your network. From Windows Explorer, locate the published Opal project file (.PRP) for the application. Right-click the .PRP file, and select Create Shortcut from the shortcut menu. You can then move your shortcut icon anywhere you choose.
Opal Player can start Opal network applications directly from batch files, such as an AUTOEXEC.BAT, a DOS command line, or on the Windows Start button Run menu. Use the following command format:

```
<Opal Player executable path> <Opal PRP file path>
```

For example:

```
c:\Opal\program\OpalPlay.exe c:\Opal\program \projects\projectname\publish\projectname.prp
```
Active X and Internet Explorer 3.02

The Opal Player 2.12 plug-in for Internet Explorer no longer includes the Active X plug-in. Some earlier releases of Opal Integrator automatically included an object tag for Active X support in the published HTML Opal project code (.HTM file). Because of the presence of this object tag, Opal projects published with Opal Integrator releases prior to 2.12 may now fail to launch using the Opal Player 2.12 plug-in. This is only a problem when using Microsoft Internet Explorer 3.02.

If you experience a problem launching an Opal application from Internet Explorer 3.02:

- Republish the Opal project using Opal Integrator 2.12, so it is compatible with Opal Player plug-in 2.12.

or

- Remove the object tag code from the Opal project .HTM file. The .HTM file, for example, ProjectName.HTM, is usually located in the Opal project Publish folder. The object tag code appears **bold** in the following .HTM file:

```
<HTML>
<HEAD>
<TITLE>Opal - Db2-475</TITLE>
</HEAD>
<BODY bgcolor="000000">
<CENTER>
<object ID="Opal"
classid="clsid:BDA043F0-1A35-11D1-B424-006097583F67"
width="640" height="480">
<EMBED SRC="Page1.sdo" width=640, height=480>
<PARAM NAME="PageName" VALUE="Page1.sdo">
</object>
</CENTER>
</BODY>
</HTML>
```

**Note:** Contact Opal Technical Support if you need additional assistance in modifying older Opal projects to run using the Opal Player 2.12 plug-in with Internet Explorer 3.02.
In this chapter, we will walk you through the process we used to develop some exercises for an Opal training session. For more information on any of the concepts introduced in these tours, refer to your online Opal Integrator User Guide, or the Opal Integrator Online Help.

This chapter consists of three tours:

- **Tour 1: Behaviors and Properties**—We will talk about the basic building blocks of any Opal application—objects, properties, and behaviors—and show how you can put them together to build the foundation of your Opal application.

- **Tour 2: Database Integration**—We will show you how easy it can be to integrate information from an external database into your Opal application.

- **Tour 3: Host Integration**—We will integrate information from a host application into an Opal application. Along the way, you will see how Opal can make a dull, uninteresting host application into an exciting, dynamic Web page.

**Note:** Opal allows you to set preferences according to how you work. Therefore, some of the windows shown in this tour may differ from what you see when you run Opal. For example, you can disable many of the Opal Assistants if you prefer not to use them.
Tour 1: Behaviors and Properties

For this tour, we built a simple Opal application that uses Opal behaviors to hide and display an image object. We also changed the properties of objects, and moved and aligned objects on the Opal page. On the pages that follow, we will show you how we got the results below.
Creating a New Project

First, we used Opal Assistants to create a new project. Opal Assistants combine several steps from common tasks into a compact, self-contained procedure.

When we started Opal, the Welcome to Opal assistant appeared. To start our new project, we clicked Create New Project.

Opal displayed the New Project assistant, in which we named our project and specified where to store it. When we finished with this assistant, we clicked the next button.
Tour 1: Behaviors and Properties

The second page of the New Project assistant appeared. On this page, we specified a default theme for our project.

Opal includes a number of attractive themes, or you can create your own. Specifying a default theme ensures that the application has a common overall look and feel, regardless of the number of pages in it.

We selected the Caribbe theme, then clicked next.

The third page of the New Project assistant appeared, on which we would select a default page variation for our project.

A variation is a special page layout based on the project theme. These often have specific purposes in an application. For example, we might use the Log In variation for a page used to log on to a host system or network. Opal shows all of the page variations available for a theme across the bottom of the assistant. When you click on a variation, a larger picture of it appears in the center of the assistant.

We used the Blank variation. By default, Opal used the Blank variation for each new page we added to our project. After we selected the variation, we clicked next.
The next page of the New Project assistant appeared.

This is where you would specify the operating system on which your application will run, the size of the Opal screen to display, and the screen resolution your application will look best on. In most cases, you can just click the next button on this assistant and accept the default settings, as we did.

Another page of the New Project assistant appeared.

On this page, we specified a name for the first page of our application, then we clicked next.
The last page of the **New Project** assistant appeared, informing us that our application will be created when we click **finish**. This page also lists some menu commands we can use to add a new page, save our project, run our project, and change the project properties.

We clicked **finish** to create our project using the information collected by the **New Project** assistant.
When we clicked **finish**, the assistants closed, and the first page of our new application appeared in the Opal workspace. As you can see, the new project is somewhat different from the finished project shown at the beginning of this tour. In the next section, we will discuss how we deleted the extra buttons, changed the heading of the page, and changed the button labels.
**Tour 1: Behaviors and Properties**

**Setting Up the Page**

Page variations are used for a variety of purposes, so you will probably need to customize the page as Opal creates it so it works for your application. For example, the page we created for this tour had more buttons on it than we needed, so we deleted some.

We could have deleted the buttons by selecting each one and pressing the DELETE key on our keyboard. To save time, we used the mouse to drag a rectangle around three of the buttons on the page. When we released the mouse button, a blue dashed line surrounded the objects, indicating they were selected. When we pressed DELETE, Opal removed all three selected buttons from the page.
We also wanted to change the word **Title** at the top of our page to something more interesting. In Opal, the word **Title** is a Static Text object.

Every Opal object has a set of characteristics called **properties**. Related properties are grouped into **categories**. An easy way to change a property value is to right-click the object so that a shortcut menu appears, then select **Properties** from the menu. This displays the Properties window, which lists all of the property categories for the selected object.

To change the word **Title** to **Behaviors and Properties**, we changed the value of the **Static Text** category **Text** property of the Static Text object.

To change the font, size, and style of the Static Text object, we clicked the column to the right of the **Static Text** category **Font** property to display the Font dialog box, then changed the values to **Arial 28 Bold**.
Tour 1: Behaviors and Properties

We were finished with the Static Text object, but we needed to change the button labels. We wanted one of the button labels to say Hide, and the other to say Show.

We clicked one of the Button objects at the bottom of our page to select it. When we did that, the list in the Properties window changed to show the properties for a Button object.

To change the word Button to Hide, we changed the value of the Button objects Label category Text property.

We repeated this action for the other button, changing the Text property value to Show.

We were finished working with properties for the moment, so we clicked to close the Properties window.
At this stage, our project began to look more like our target. In the next section, we will discuss how we added the image to the page and positioned the objects on the page.
Adding an Image and Arranging Objects

Next, we used the Opal Content Browser to add an image to our page. The Content Browser manages content for Animation, Image, Sound, and Video objects in Opal. You can also add content to other object types from the Content Browser. For example, you can add an image to a Button object. Any time you add an object that requires content to your Opal page, the Content Browser appears.

We wanted to add an Image object, so from the Object menu we pointed to New, then selected Image.

Image objects require content, so when we selected Image, the Opal Content Browser appeared. As shown below, the Content Browser contains two panes. The left pane, called the Content Locations pane, displays content that has already been added to the page (Current Content), and locations you can browse to find available content (My Computer). The right pane, called the Selected Content View pane, displays a preview of the selected content and information about the properties of that content.
Our project had no existing content, so we browsed to find our image. The Content Browser simplifies the task of browsing for content by listing only files that are valid as content for the object added or for the currently selected object. Because we added an Image object, only files in one of the valid image formats (.BMP, .GIF, .JPG, or .TIF) were listed in the Content Locations pane. We selected a bitmap image called Fish9.bmp.

We clicked OK, and the New Object dialog box appeared so we could name the Image object. Opal supplies a default name based on the name of the content file. We clicked OK to accept the default name, Fish9.
Opal centered the image object on the page. Our page was looking pretty good, but we needed to arrange the objects a little better. We decided to leave the Image object where it was, and move the Hide and Show buttons.

First, we selected the two Button objects and dragged them so they were positioned roughly where we wanted them to end up.

Then, just to make sure the buttons were correctly aligned, we selected them both, and from the Object menu, we pointed to Align, then selected Bottom. This aligned the two buttons along the bottom edge of the lowest button.
Now our project looked finished, but nothing happened when we clicked the Hide and Show buttons. In the next section, we will show you how we added behaviors to the Button objects and the Image object, and how we tested our project to make sure it worked when we were finished.
Tour 1: Behaviors and Properties

Adding Behaviors to Objects

In Opal, a behavior consists of actions an object performs in response to messages it receives. Some objects have built-in behaviors. For example, a Video object has the built-in ability to play a video clip. Even a built-in behavior, however, must be triggered by a behavior you specify. The behaviors you define (and those that Opal automatically defines for you) are also called message/action pairs.

In our project, we wanted to click the Hide button to make the Image object disappear and we wanted to click the Show button to make the Image object reappear.

An easy way to add behaviors is to right-click the object so that a shortcut menu appears, then select Behavior from the menu. This displays the Behavior window, which has three columns: the Message column lists messages that the object sends; the Action column lists actions the object performs in response to messages; and the Note column contains notes you add for your project.

We clicked the button to close the Note column, because we did not want to use it.

To select a message, you click an empty row in the Message column to display the button. When you click the button, a list of available message categories appears. To expand a category (thus displaying the messages in the category), click the button. To select a message, click it. You would use the same procedure to select an action, except you would select from the Action column.
We wanted the Hide button to act as soon as we clicked it, so we selected the **Button Activated** message from the **Button Messages** category.

When we click the Hide button, we want the Image object to disappear. However, the action we pair with the **Button Activated** message must be something that the Hide button can do.

The Hide button cannot make the Image object disappear, but it can send a message to the Image object. Therefore, we selected the **Send Message** action from the **Send Message Actions** category.

We also had to specify to which object the Hide button would send the message, and the message to send.
To do this, we set parameters of the **Send Message** action. We selected the name of our Image object (**Fish9**) from the **Object** parameter drop-down list, and the name of the message to send from the Message parameter drop-down list. We wanted the Hide button to request the Image object to disappear, so we set the Message parameter to **Hide Request**.

Some messages are typically paired with specific actions. When this is the case, Opal automatically completes the message/action pair for the target object. When we clicked the Go to Target Object button on the Behavior window, Opal displayed the behaviors that it automatically set for the Image object.

Because we defined that the Hide button would send a **Hide Request** message to the Image object, Opal automatically specified a **Hide Request** message for the Image object, and paired it with its most typical action—**Hide Object**.
We were finished setting up the Hide button, but we still had the Show button to define. We selected the Show button on our Opal page, and Opal displayed the empty Behavior window for the object.

Like the Hide button, we wanted the Show button to act as soon as we clicked it, so we selected the **Button Activated** message from the **Button Messages** category. Again, we selected the **Send Message** action from the **Send Message Actions** category because the Show button can only send a message to another object.

We set the **Send Message** action **Object** parameter to **Fish9**, and we set the **Message** parameter to **Show Request**.

Once again, when we clicked the Go to Target Object button, Opal displayed the behaviors that it automatically set for the Image object.

Because we defined that the Show button would send a **Show Request** message to the Image object, Opal automatically specified a **Show Request** message and paired it with its most typical action—**Show Object**.
Our project was finished and ready to be tested.

We clicked \( \times \) to close the Behavior window, then we clicked the Run button \( \Rightarrow \) on the Opal Toolbar. We clicked Yes when Opal prompted us to save our work.

Opal then displayed our finished project in Run mode. When we clicked the Hide button, the Image object disappeared. When we clicked the Show button, the Image object reappeared.

When we finished testing our project, we clicked the Stop button \( \square \) on the Opal Toolbar to return to Edit mode.

This concludes the first Opal Integrator tour. By now, you should be familiar with the some of the basic concepts used in developing an Opal Application.
Tour 2: Database Integration

For this tour, we used the Opal Data Mapper to build an application that integrates an ODBC-compliant database. The application looks up information and adds, deletes, and modifies records.
Tour 2: Database Integration

Creating a New Project

As we did in the first tour, we used Opal Assistants to create a new project, which we named Database Manipulation. This time, we selected the Kiwi theme and the Browser variation. When we clicked finish, the assistants closed and the first page of our new application appeared in the Opal workspace. Again, the new project differed slightly from our goal, shown above.

We selected the Button objects we had no use for and deleted them. We also changed the value of the Static Text category Text property for the page header Static Text object from Title to Database Manipulation.

In the next section, we will discuss how we integrated records from an internal database with our Opal project.
**Adding Database Fields**

Opal uses a tool called the Data Mapper to integrate information from external sources with Opal applications. You can use the Data Mapper to integrate data from ODBC-compliant databases or from host terminal applications. In this tour, we will concentrate on integrating database information.

From the **Tools** menu, we selected **Data Mapper**. The **Data Mapper Welcome** assistant appeared, and we selected **Database Session**.

When we clicked **finish** on the **Data Mapper Welcome** assistant, the **Data Session Properties** dialog box appeared. We clicked the **Database** tab, and selected the name of the database we want to integrate from the **Data source** drop-down list.
Tour 2: Database Integration

When we clicked **OK** on the Data Session Properties dialog box, the Data Mapper appeared. Opal connected to the database we specified, and listed the data table (**Items**) in the database in the left pane of the Data Mapper. When we clicked the table name, the list expanded to show the columns of information in the table (**ItemNumber**, **ImageNumber**, and so on).

To enable Opal to integrate data from the database, we must create a connection or **pipe** between our application and the database. To do this, we dragged the table name (**Items**) from the left pane of the Data Mapper to the right pane. When our mouse pointer was over the right pane, it changed to indicate that we were creating the data pipe.
When we dropped the data table on the right pane of the Data Mapper, the pipe between the database and our Opal application was created, and data fields in the Items table appeared in the right pane. Opal automatically named the database pipe we created Items_Pipe, as shown in the Pipe drop-down list.
Tour 2: Database Integration

We wanted the database information sorted in ascending order by item number on our Opal page.

To accomplish this, we selected the Data Mapper Sort tab and clicked the Sort column to the left of Items.ItemNumber (Opal's name for the item number column in the database) until the word ascending appeared.

Next, we wanted to make sure that the item number and category information was listed before the other database field on our Opal page.

To do this, we returned to the Columns tab, clicked the Items.Category field, and dragged it so it was just below the Items.ItemNumber field.
Now we could place all of this database information on our Opal page (this is called *mapping*).

First, we selected all of the database fields in the right pane of the Data Mapper by right-clicking a field name in the right pane. A shortcut menu appeared, from which we selected *Select All*.

Then, we dragged the selected fields off of the Data Mapper, and dropped them on our Opal page. As our drag cursor neared the edge of the Data Mapper, the Data Mapper hid itself so we could see the Opal page.
Tour 2: Database Integration

When we dropped the database fields on the Opal page, the Data Mapper Drag-and-Drop assistant appeared. We selected to drop the database fields as Edit Fields, because we wanted to be able to add, delete, and update records in the database.
When we clicked **finish** from the **Data Mapper Drag-and-Drop** assistant, our Opal page appeared with the newly dropped data fields already formatted and labeled. We selected all of the data fields and moved them to the center of the page. Then, we clicked the button to test our project. As shown below, the fields were populated with information from the database, indicating that we had successfully mapped data to our Opal page.

![Database Manipulation](image)

In the next section of this tour, we will show you how we added controls to our Opal page to go to the next or previous database records, and how we added a Combo Box object to filter the data.
Adding Controls and Filtering Data

Mapping an entire database of information to an Opal project is not very effective if you can only display one record of data. So, we added behaviors to the right and left arrow buttons at the bottom of our Opal page that would query the database and display the next or previous record.

First, we added a **Button Activated/Go to Next Record** message/action pair to the right arrow Button object on our page. The **Go to Next Record** action is in the **Data Pipe Actions** category on the **Action** list.

We specified the name of the pipe (**Items_Pipe**) through which the Next button would access the database in the **Go to Next Record** action **Data pipe** parameter.

Then, we added a **Button Activated/Go to Previous Record** message/action pair to the left arrow Button object on our page. The **Go to Previous Record** action is in the **Data Pipe Actions** category.

We specified the name of the pipe (**Items_Pipe**) through which the Previous button would access the database in the **Go to Previous Record** action **Data pipe** parameter.
Next, we added a Combo Box object to the page that would allow us to search for database entries by category.

We returned to the Data Mapper and expanded the Items table in the left pane, then dragged only the Category field to the right pane. Opal named this new database pipe Items_Pipe2.

Because we did not want duplicate product category entries to appear in the Combo Box object on our Opal page, we cleared the Allow duplicates check box on the Data Mapper Filter tab.

After we cleared the Allow duplicates check box, we switched back to the Columns tab and dragged the Category field to our Opal page. Again, the Data Mapper hid itself so we could see the page.
When we dropped the Category field on our Opal page, the Data Mapper Drag-and-Drop assistant appeared. Because we were dropping a single field, the assistant displayed a wider variety of choices. We selected to drop the Category field as a Combo Box object because we want our users to select the product category from a drop-down list.

After we clicked finish from the Data Mapper Drag-and-Drop assistant, we dragged the Combo Box object on our Opal page to position it just above the Labeled Edit objects we created earlier.
To link the database information that populates the Labeled Edit objects to the **Category** information in the Combo Box object, we needed to set up a **filter**. A filter screens the data sent through the database pipe so that only the appropriate information is displayed on the Opal page.

To create our filter, we returned to the Data Mapper and selected **Items_Pipe** from the **Pipe** drop-down list so the database information we mapped to the Labeled Edit objects on our Opal page appeared in the right pane.

We then selected the **Filter** tab, and in the second row of the **Branch/Column** column, we selected **Category** from the drop-down list of fields.

On the same row in the **Operators** column, we selected `=` from the drop-down list.
Tour 2: Database Integration

In the same row of the **Globals/Properties...** column, we selected **Object Properties** from the drop-down list. When we did so, the **Object Properties** category expanded, and we selected **Items_Category_Combo_Box**. This category expanded, and we selected **Text**. This category also expanded, and we selected **Text** again. Thus, we had defined a filter to tell Opal to select records for display on the page whose product category exactly matched the value the user selected from the Combo Box object.

The final step in filtering data was to assign behaviors to the Combo Box object so it would properly interact with the database and our Opal page.

We closed the Data Mapper. When our Opal page appeared, we right-clicked the Combo Box object and selected **Behavior** from the shortcut menu to display the Behaviors window.

Opal automatically added a number of behaviors to the Combo Box object when we dropped the **Category** field to the page. We needed to add one more behavior so the object would interact properly with the database.
Tour 2: Database Integration

We added the Drop Down Up message from the Combo Box Messages category to the first empty row in the Message column.

Then we added the Get Data from Source action from the Data Pipe Actions category to the same row in the Action column.

As a result of this message/action pair, when a user makes a selection from the Combo Box object at run time, Opal will query the database and update the information on the Opal page.

Our final task in integrating a database with our Opal application was to add the ability to insert, delete, and update records in the database. In the next section of this tour, we will describe how we added properties and behaviors to Button objects to accomplish this task.
Inserting, Deleting, and Updating Database Records

As you have probably realized, adding a few behaviors to a Button object can make it very useful—and very powerful—in an Opal project. For our Database Manipulation project, we wanted our users to be able to update the database with new information, or to delete invalid information from the database.

To do this, we added behaviors to the three buttons on the left side of the Opal page.

First, we changed the **Label** category **Text** property values for the three buttons to match their functionality—we defined an Add button, a Delete button, and an Update button.

Then we added behaviors to each of the buttons.

For the Add button, we paired the **Button Messages** category **Button Activated** message with the **Data Pipe Actions** category **Add Record** action, and set the **Data pipe** parameter to **Items_Pipe**. Then we paired a second **Button Activated** message with the **Data Pipe Actions** category **Get Data from Source** action, also setting its **Data pipe** parameter to **Items_Pipe**.

For the Delete button, we paired the **Button Messages** category **Button Activated** message with the **Data Pipe Actions** category **Delete Record** action, and set the **Data pipe** parameter to **Items_Pipe**. Then we paired a second **Button Activated** message with the **Data Pipe Actions** category **Get Data from Source** action, also setting its **Data pipe** parameter to **Items_Pipe**.

For the Update button, we paired the **Button Messages** category **Button Activated** message with the **Data Pipe Actions** category **Send Data to Source** action and set the **Data pipe** parameter to **Items_Pipe**.

That completed our Database Manipulation project. When we saved the project and ran it, we were able to display items from the database, select different categories of items to display, and even add, delete, and update information in the database.
Tour 3: Host Integration

For the tour, we used the Data Mapper to build an Opal application that integrates information from a 3270 host application. The four-page application navigates between terminal screens in the host session and displays information from the host on the Opal pages.

In our application, we created a **Menu** page from which users can access information on three other pages. Forward and Back buttons on the **Orders** page allow users to scroll through pages of data integrated from the host. The **Year at a Glance** page presents tabular data integrated from a table in the host application. It, too, has buttons with behaviors that allow scrolling. The **Financial** page summarizes sales figures for the last quarter and the last year. All three pages include an Exit button with which the user can return to the **Menu** page.
Creating a New Project and Adding Additional Pages

As in the previous tours, we used the New Project assistant to create a new project, which we named Basic Host Navigation. For this project we selected the Monochrome theme and the Blank variation. We named our first page Menu. When we clicked finish, the assistant closed and the first page of our new application appeared in the Opal workspace.

We selected all of the buttons on our new Menu page and deleted them. We also changed the value of the Static Text category Text property for the page header Static Text object from Title to MENU.

Then, we created the three pages that will integrate host information.
To create a new page, we selected **New** from the Opal Integrator **Page** menu. The **New Page** assistant appeared, and we specified **Orders** as the name of the new page.

When we clicked **finish** to close the **New Page** assistant, the new page appeared in the Opal workspace.

As we did with the **Menu** page, we changed the **Static Text** category **Text** property value of the Static Text object from **Title** to **Orders**. We repeated this process twice more to create a **Financial page** and a **Year at a Glance** page.
Tour 3: Host Integration

Creating an Opal Host Session

Our pages were ready for data, so we initiated a connection (using WinHLLAPI emulation) between Opal and the host computer, and we navigated through the host screens from which we wanted to integrate data, recording each screen and action as we progressed.

First, we selected Data Mapper from the Tools menu. The Data Mapper Welcome assistant appeared, and we selected Host Session.
When we clicked `finish` on the Data Mapper Welcome assistant, the Data Session Properties dialog box appeared.

On the **Session** tab, we verified that the **Global object** check box was selected, because we want this host session to be available from every page in our project.

![Data Session Properties dialog box](image)

We clicked the **Connections** tab and selected **Local Using WinHLLAPI** from the **Connection Type** drop-down list.
Tour 3: Host Integration

Then we clicked the **Host** tab and set the WinHLLAPI **Short name** field to **N**, which matched the WinHLLAPI short name already set up in our terminal emulator program.

This completed our host session setup, so we clicked **OK**. The Data Mapper displayed a blank terminal emulator screen.
Our next task was to navigate and record the host terminal screens.

From the Data Mapper, we clicked the red Record button. The Data Mapper connected us to the host session and displayed the New Screen dialog box. The first screen of our host application appeared behind the New Screen dialog box. In the **Name** field, we specified the name of the first screen, **Banner**. Because the **Banner** screen was unknown to the Data Mapper, the **Record as path to new screen** option was automatically selected. Then, we clicked **OK**.
Tour 3: Host Integration

When we clicked OK, the New Screen dialog box closed, leaving the Banner screen displayed in the Data Mapper. The Data ID Tags automatically defined by the Data Mapper for the Banner screen were highlighted in green. The Data Mapper uses Data ID Tags to identify individual host screens.
We pressed ENTER to go to the next host screen. The New Screen dialog box appeared, and behind it we could see a new host screen. Again, the Data Mapper automatically selected **Record as path to new screen**. We named this host screen **Login**, then clicked **OK** to record the screen.

When we clicked **OK**, the New Screen dialog box closed, leaving the **Login** screen displayed in the Data Mapper.
We pressed ENTER to go to the next host screen. The New Screen dialog box appeared, and behind it we could see a new host screen. Again, the Data Mapper automatically selected **Record as path to new screen**. We named this host screen **Menu**, then clicked **OK** to record the screen.

When we clicked **OK**, the New Screen dialog box closed, leaving the **Menu** screen displayed in the Data Mapper.
The cursor was in the Customers field on the Menu screen. However, we wanted to record the Orders screen so we could integrate its data into our project, so we pressed the TAB key to move the cursor to the Orders field, then pressed ENTER.

Again, the New Screen dialog box appeared. The Data Mapper automatically selected Record as path to new screen. We named this host screen Orders, then clicked OK to record the screen.
Tour 3: Host Integration

When we clicked OK, the New Screen dialog box closed, leaving the Orders screen displayed in the Data Mapper.

![Data Mapper - Global Host Session](image)

In our Opal application, the Orders page will include three buttons—a Forward button to go to the next record, a Back button to return to the previous record, and an Exit button to return to the Menu page. PF keys (at the bottom of the Orders screen) perform those same actions for the host application. As we recorded host screens, the Data Mapper also recorded how we navigated from one screen to the next. We also had to record navigation within a screen.
To record the first of these actions, we pressed F8. The New Screen dialog box appeared, but this time the Data Mapper selected Define as data action.

We clicked the Action tab, selected Next Record from the Action drop-down list, then clicked OK to record the data action. This recorded the action of pressing F8 as a Go to Next Record action in Opal.
Tour 3: Host Integration

When we clicked OK, the New Screen dialog box closed, leaving the Orders screen displayed in the Data Mapper.

To record the next button action, we pressed F7. The New Screen dialog box appeared, with Define as data action selected. We clicked the Action tab, selected Previous Record from the Action drop-down list, then clicked OK to record the data action. This recorded the action of pressing F7 as a Go to Previous Record action in Opal.

When we clicked OK, the New Screen dialog box closed, leaving the Orders screen displayed in the Data Mapper.

To record the final button action, we pressed F3. The New Screen dialog box appeared. Because F3 has the host function of exiting the current screen (the Orders screen) and returning to the previous screen (the Menu screen), the Data Mapper automatically selected Record as path to previously encountered screen and set the screen name to Menu on the New Screen dialog box.

When we clicked OK, the New Screen dialog box closed, leaving the Menu screen displayed in the Data Mapper. We repeated the process we used to record the Orders screen and its AID keys to record the Last Quarter Sales, Last Year Sales, and Year at a Glance screens.

For the Last Quarter Sales and Last Year Sales screens, we recorded only the F3 action to exit to the previous screen.

For the Year at a Glance screen, which displays tabular data, we recorded the F8 key with a Table Page Down action, the F7 key with a Table Page Up action, and the F3 key with an exit to previous screen action.
Tour 3: Host Integration

When we clicked OK to close the New Screen dialog box, recording the F3 action for the **Year at a Glance** screen, the **Menu** screen appeared in the Data Mapper. We pressed F3 once more, and clicked OK when the New Screen dialog box appeared to record the exit to previous screen action for the **Banner** screen. When the New Screen dialog box closed, the **Banner** screen appeared in the Data Mapper.

So far, we had created four screens for our Opal application and had recorded their host counterparts. In the next section, we will describe how we added controls to the Opal pages and integrated the host data with them.
Tour 3: Host Integration

Adding Data and Controls to Pages

Next, we integrated host data with our Opal pages and added controls with which we could navigate the application and the data.

First, we integrated data from the host Orders screen with our Opal Orders page.

From the Data Mapper Tools menu, we selected Field to put the Data Mapper in the proper mode to map data fields.

The drop-down list at the top of the Data Mapper lists all of the host screens we recorded. We selected Orders from the drop-down list, and the Orders terminal screen appeared.
The first field we wanted to map from the **Orders** screen was the **CUSTOMER ID** field. We double-clicked the area to the right of the field label, and a Field dialog box appeared.

We named the field **Customer ID**, then clicked **OK**. The Field dialog box closed, and the Data Mapper displayed the newly mapped field with a purple outline.

We repeated the process to map the **Part Number**, **Part Description**, and **Order Quantity** fields on the **Orders** screen, naming them accordingly.

We selected all four fields by dragging a rectangle around them with the mouse. Then, we right-clicked one of the selected fields, displaying a shortcut menu. We selected **Integrate**.
Tour 3: Host Integration

The Data Mapper Drag-and-Drop assistant appeared, and we selected Edit Fields, then clicked finish.

Note: The right-click method of integrating data is an alternative to dragging the selected fields to the Opal page.

We then minimized the Data Mapper and centered the four new Labeled Edit objects on our Opal page.
Of the five buttons across the bottom of the Orders page, we deleted the second and fourth. We changed the **Label** category **Text** property values for each of the remaining buttons as follows:

- We changed the leftmost button to **Back**.
- We changed the center button to **Forward**.
- We changed the rightmost button to **Exit**.

Next, we programmed buttons on our Opal page to interact with the host data. We right-clicked the Back button and selected **Behavior** from the shortcut menu. The Behavior window appeared.
So the Back button would display the previous record of Orders data when we clicked it, we selected the **Button Activated** message from the **Button Messages** category and the **Go to Previous Record** action from the **Data Pipe Actions** category. Opal automatically completed the **Data pipe** field.

We wanted the Forward button to display the next record of Orders data when we clicked it. We selected the Forward button and specified the **Button Activated** message from the **Button Messages** category and the **Go to Next Record** action from the **Data Pipe Actions** category. Again, Opal automatically completed the **Data pipe** field.
Finally, we wanted the Exit button to switch from the Orders page to the Menu page. We selected the Exit button and specified the Button Activated message from the Button Messages category and the Change Page action from the Page Actions category. We selected Menu from the Page name drop-down list.

Next, we integrated data from the host Last Quarter Sales and Last Year Sales screens with our Opal Financial page.

From the Opal View menu, we selected Pages. The Pages window appeared, from which we selected Financial to switch to the Financial page.

Because we would be combining the data from two host pages in the Financial page, we added two Static Text objects as section labels for the page.
Tour 3: Host Integration

For each object we added, we pointed to New from the Object menu and selected Static Text. The New Object – Static Text assistant appeared, on which we named the object and specified its label text.

We also added a horizontal Separator object to the page. We pointed to New from the Object menu and selected Separator. We centered the Separator object on the Opal page, and set its Size category Width property to 640 to make it the width of the page.

We deleted all but the rightmost button from the Financial page, then set the button’s Label category Text property to Exit. We specified a Button Messages.Button Activated/Page Actions.Page Change behavior for the button, selecting Menu from the Page name drop-down list.
We arranged the Static Text objects and the Separator object so our **Financial** page resembled the following:
Tour 3: Host Integration

To map the data fields from the Last Quarter Sales and Last Year Sales host screens to the Financial page, we repeated the procedure by which we mapped fields from the Orders screen. We mapped the Sales Units, Sales Dollars, and Net Profit fields from both sales screens and arranged them on the Financial page as shown below:

Next, we integrated data from the host Year at a Glance screen with our Opal Year at a Glance page.

From the Pages window, we selected Year at a Glance to switch to the Year at a Glance Opal page. We deleted all of the buttons from the bottom of the page, then switched to the Financial page. On the Financial page, we selected the Exit button, then selected Copy from the Edit menu. We then switched back to the Year at a Glance page and selected Paste from the Edit menu. The Exit button appeared in the bottom right corner of the Year at a Glance page, with the same behaviors we had defined for it on the Financial page.
Then, from the Data Mapper Tools menu, we selected Table to put the Data Mapper in the proper mode to map tabular data.

We selected Year at a Glance from the pages drop-down list, and the Year at a Glance terminal screen appeared, displaying 10 Table Record Value fields.

We selected all 10 fields, and the Column dialog box appeared.

In the Name field we typed Table Data, then we clicked OK.
Tour 3: Host Integration

We dragged the selected table data to the Opal page and the Data Mapper Drag-and-Drop assistant appeared, from which we selected Table, then clicked finish.

The Table Opal created on the page was sized to show one page of table data, and had a scroll bar with which users could scroll up and down the data. We wanted users to be able to instantly display the next or previous page of table information, so we added “predefined” buttons to the page by dragging Table Action identifiers from the Data Mapper.
We clicked the Data Mapper Action tab. The Action tab displays recorded data actions, such as the F8 (Table Page Down) and F7 (Table Page Up) actions we recorded earlier.

With Year at a Glance selected in the pages drop-down list, we selected Table from the drop-down list just below it. The left pane of the Actions tab then displayed the name of the table we dragged to the Opal page (Table_Data), and the right pane displayed the data actions we defined for it.

From the right pane of the Action tab, we dragged Table Page Down to the Opal page, creating a Table Page Down button. We did the same with Table Page Up, creating a Table Page Up button.
Tour 3: Host Integration

On the Opal page, we arranged the Table object and the two buttons we just created, so our **Year at a Glance** page ended up looking like this:

![Year at a Glance page screenshot](image)

Our final task, before testing and publishing the project, was to integrate the host **Menu** screen with our Opal **Menu** page, enabling our users to access the **Orders**, **Financial**, and **Year at a Glance** pages from it.
We switched to the Opal **Menu** page, then selected **Menu** from the page drop-down list. The **Menu** terminal screen appeared.

From the **Tools** menu, we selected **Menu** to put the Data Mapper in the proper mode to map menu items.

We double-clicked the **Orders** menu item, and the Menu Item dialog box appeared.

In the **Label** field, we typed **Orders** as the menu item label, then clicked the **Path** tab.

On the **Path** tab, we verified that this menu item was set to navigate to the Opal **Orders** page.

We clicked **OK**, and the Menu Item dialog box closed.
Tour 3: Host Integration

On the host Menu screen, we double-clicked the Year at a Glance menu item. When the Menu Item dialog box appeared, we typed Year at a Glance in the Label field, then clicked the Path tab.

From the Path drop-down list on the Path tab, we selected Year at a Glance as the Opal page to which this menu item would navigate, then clicked OK.

Back on the host Menu screen, we double-clicked the Last Quarter Sales menu item. When the Menu Item dialog box appeared, we typed Financial in the Label field, then clicked the Path tab.

From the Path drop-down list on the Path tab, we selected Last Quarter Sales as the Opal page to which this menu item would navigate, then clicked OK.

On the host Menu screen, we selected the three menu items we just defined, right-clicked one of them, and selected Integrate from the shortcut menu. The Data Mapper Menu Drag-and-Drop assistant appeared, from which we selected Button Set as the display method for the menu items and clicked finish.
We then closed the Data Mapper and positioned the menu buttons in the center of the Opal **Menu** page, as shown below.

![Menu page screenshot](image)

Opal automatically set behaviors for the menu buttons when we dragged them from the Data Mapper, but we needed to check the parameter settings.
We right-clicked the Orders button and selected Behavior from the shortcut menu. The Behavior window appeared.

We selected the Get page data check box so that each time a user clicked the Orders button, the Orders page would appear with the current data. We also verified that the other parameters were set appropriately.

On the Opal page, we clicked the Year at a Glance button, and its behavior appeared in the Behavior window.

Again, we selected the Get page data check box and confirmed the other parameter settings.

On the Opal page, we clicked the Financial button, and its behavior appeared in the Behavior window.
In this instance, we needed to modify the default behaviors. Because we originally mapped this page from the Last Quarter Sales host screen, the Terminal screen parameter was set to Last Quarter Sales. Because our Financial page was to display data from two different host screens, we needed to instruct Opal to get all the data from both screens when it navigated.

We wanted the Financial button to display the Financial page and to direct the host to automatically navigate to all the necessary terminal screens.

To do this, we changed the Opal page parameter value to Financial and the Terminal screen parameter value to <Auto>.

We then closed the Behavior window.

Our final task was to specify that the user would see the Menu page first when accessing the application.

From the View menu, we selected Pages, then we clicked to the left of Menu on the page list. A running man icon appeared, indicating that the Menu page would always appear first in the application.

We then saved, tested, and ultimately published our Host Integration application. When we ran the application, we were able to access the Orders, Year at a Glance, and Financial pages from the Menu page and return to the Menu page from them. On the Orders page, we could scroll forward or back through the host data records using our Forward and Back buttons. On the Year at a Glance page, we could page up and down through the tabular data using the “predefined” buttons we dropped from the Data Mapper. The Financial page drew its data from two separate host screens, Last Quarter Sales and Last Year Sales.
Tour 3: Host Integration
action
A behavior performed by an Opal object at run time in response to a message. See also message and object.

AID key
Attention Identifier Key. In 3270 and 5250 host environments, an AID key sends a signal to the host when it is pressed. In these environments, users can enter data, such as a user ID and password, which is not sent to the host until the AID key (in this case, the ENTER key) is pressed. The Data Mapper Path pane shows the keystrokes that have been recorded as navigational paths to each screen.

application
A complete, self-contained program that performs a specific function directly for the user. This is in contrast to system software such as the operating system kernel, server processes, and libraries that support application programs. Editors, spreadsheets, and text formatters are common examples of applications. Opal Integrator lets developers create applications incorporating external databases and host terminal applications to replace outmoded terminal- or character-based applications.

assistant
An automated process that combines two or more tasks to create an Opal object based on your input.

behavior
An object’s ability to interact with other objects. In Opal, objects perform actions in response to specific messages. Each message/action pair is referred to collectively as a behavior, as are all the message/action pairs assigned to any object, or all the message/action pairs contained in a page or project. Note that, in a different sense, Opal objects encapsulate characteristic methods of behavior that are brought into play by assigning message/action pairs. See also action, message, and object.

bitmap
An image file format used by Opal Integrator. Bitmap files have a .BMP extension. See also content and Content Browser.
**browser**
1) A program, such as Microsoft Internet Explorer, that allows users to read files from the Internet or intranets. 2) A program, such as Windows Explorer, that allows users to navigate a network or file structure by clicking or selecting items from a directory tree or icons that represent items on a directory tree. See also Content Browser.

**caption menu**
A pop-up menu available from the Opal Integrator Properties, Behavior, and Objects/Pages windows, and as an alternative to the standard menu bar. To view the caption menu, click the button in the upper right corner of the Properties, Behavior, or Objects/Pages window or the Opal Integrator workspace.

**character**
Any symbol that can be stored and processed by a computer.

**child screen**
In the Data Mapper, a host terminal screen that can be reached from the current screen. Any host screen can be a child of the current screen if you record the navigational keystrokes needed to reach it.

**client**
A computer or other object that relies on another. In Opal, a client computer relies on a server for storage of enterprise class applications. Also in Opal, an object in an Opal application that relies on a Data Pipe object for data from an external data source. See also client/server and server.

**client/server**
A common form of distributed system in which tasks are split between a server and a network of clients connected to it. A client sends requests to a server, according to some protocol, asking for information or action, and the server responds. There may be one centralized server or several distributed ones. This model allows independent placement of clients and servers on nodes in a network, possibly on different hardware and operating systems appropriate to their function; for example, fast server/cheap client. See also, client and server.

**CN3**
32-bit extension used in content files. Includes .WAV, .AVI, .BMP, and .CUR files.

**communications layer**
The emulator software used to connect a personal computer to an external information source. Third-party communications layers provide a standard interface that bridges the language gap between applications running on host computers and software running on personal computers. In Opal, the communications layer is controlled through session objects on the Opal page. See also HLLAPI, WinHLLAPI.

**Container objects**
Container objects organize sub-objects within some Compound objects—including group frames, folders, labeled edits, and tables. Containers divide the Compound object into groups of sub-objects that can be manipulated as a whole, while maintaining separation from other groups. At the same time, containers may themselves reside in larger containers. Group Frame objects, Folder objects, and Table objects are Container objects.
**content**
An image, sound, video, animation or cursor associated with an Opal object. Content can be used to create a custom appearance for a standard Opal object, or to associate custom sounds or cursors with objects. See also Content Browser and Opal Integrator.

**Content Browser**
A window in Opal Integrator for locating multimedia files used as content for Opal objects. See also browser, content, and Opal Integrator.

**context dragging**
See shortcut dragging.

**data**
Information in a form that can be used by a computer.

**data client**
See Display object.

**Display object**
Any Opal object that can receive data from a Data Pipe object and displays it for the user. Display objects may also collect user input. Edit, Labeled Edit, Table, and Static Text objects typically serve as Display objects.

**Data Mapper**
A tool for integrating external information sources into Opal applications. The Data Mapper records general data about an external database or host terminal application, provides a visual context for selecting and manipulating essential elements, and generates mapped Opal objects for passing information between the external data source and the user.

**Data Pipe object**
The central figure in integrating information sources, Data Pipe objects pass information from external sources and to Opal Display objects. They also return user input from the Display object to the external source. The Data Pipe object transforms the data it receives from the external source into text properties. It uses these text properties to update any Display objects whose Text property is linked to it. This allows uniform treatment of differing information sources and provides a way for non-related information sources to communicate with each other. See also information source.

**data source**
See information source.

**database**
A system for organizing, storing, retrieving, and manipulating data. Relational databases organize data from a variety of sources into a set of tables, define the relationships between tables, and provide mechanisms for manipulating and analyzing the data.

**database management**
The task of storing data in a database and retrieving information from that data.

**database session**
A single connection to a database established through a communications layer. In Opal Integrator the database session is managed through a Database Session object. You can establish multiple database sessions in an application. These may be connected to the same or to different databases. See also database and session.

**DLL**
A file containing program procedures needed at run time.
**expression**
In programming, an expression is any legal combination of symbols that represents a value. In Opal, simple expressions can be inserted as property values and as parameters for some object actions. Expressions in Opal are written in OpalScript.

**field**
In a database, a field is the location within a record where a certain type of data is stored, such as last name, first name, or salary. In relational databases, fields are called columns.
In host terminal applications, a field is a portion of the terminal screen reserved for a particular use. Fields can be identified by the row coordinate of the left edge, and the height and width in character coordinates. See also protected field and unprotected field.

**frame**
A single image in a series of images attached to an Animation object.

**global**
In Opal, an object or variable that is available throughout an entire project as opposed to a single page. A global object appears on every page of an Opal application and can be used anywhere in the application. Contrast with local.

**GO3**
An Opal extension added to the graphical object file that contains all the image and animation content for the project. Content stored in the .GO3 file has been compiled and compressed into a single, internally priority-sorted file.

**Graphical User Interface (GUI)**
A user interface that allows users to manipulate buttons, scrollbars, and other images, instead of working in text-based screens.

**HLLAPI**
A component of third-party communications layers that bridges the language gap between applications running on your personal computer and applications running on host computers.

**host**
A host is the main computer in a system of computers and terminals connected by communications links. Host computers are usually mainframe or minicomputers. Large databases and applications are run on the host computer, and users interact with these databases and applications through terminals.

**host applications**
A character-level application stored on a host computer and made available to terminals in other locations. Used interchangeably with host terminal application.

**host field**
A data field on the host terminal screen.

**host session**
A single connection to a host application established through the communications layer. In Opal, multiple host sessions can be set up and connected to the same or to different host computers. In Opal Integrator, a Host Session object is generated for each host session. The Host Session object represents the host session on the Opal page. See also host and session.
**host table**
Data that is displayed in tabular form on the host terminal screen.

**host terminal application**
See host applications.

**index**
1) A number denoting an item’s position on a menu or list. For example, the index of the first item in a menu is 1. Also, a number denoting a character’s position within a text string. 2) A number denoting a string’s position within a text string. The index of a string corresponds to the index of the first character. For example, in the word Hello, the index of H is 0, the index of e is 1, and so on.

**information source**
A place where data is stored. You can add or retrieve data from an information source. Data Pipe objects serve as intermediate information sources between host applications or databases and Display objects on the Opal page. Interchangeable with data source.

**integrate**
To embed links to external databases or host applications in an Opal application.

**JScript**
Microsoft’s version of JavaScript, which is built into Microsoft Internet Explorer. Netscape’s JavaScript and Microsoft’s JScript are not entirely compatible, so Web pages containing JavaScript/JScript code may run differently in Netscape Navigator and Microsoft Internet Explorer.

**linking**
Making a property connection between two Opal objects. Setting an object’s Data Source property to point to a Data Pipe object creates a link. See also data source and Data Pipe object.

**local**
Pertaining to a particular page; for example, a local object or a local variable. Contrast with global.

**macro**
An instruction that invokes a sequence of actions.

**mapping**
Linking an object in an external database or on a host terminal screen to an Opal object so data can be transferred between them.

**master detail**
1) The technique of establishing a relationship between two separate data sources, especially when data drawn from one data source is used to drive the other. Master detail relationships provide a simple method for allowing two disparate data sources to communicate. 2) The relationship between two blocks of information defined on the primary key of the master block and the corresponding foreign key of the detail block. See also database, ODBC, and SQL.

**member**
Opal objects belonging to a Group Frame object are said to be members of the group frame.

**menu item**
The options listed on a one-level Menu object. On a two-level Menu object, the options listed on a secondary menu are called menu items. See also menu title.
**menu title**
The options listed on the primary menu of a two-level Menu object. Each menu title opens a secondary menu. See also menu item.

**message**
A communication that triggers an object to perform an action. Messages can come from other objects or from the system. Messages must be paired with actions to trigger a response. See also action, behavior, and object.

**mnemonic accelerator key**
In Windows applications, mnemonic accelerator keys provide keyboard access to menus and other command objects. For example, users can open the File menu by clicking ALT+F; the underscore beneath the letter F indicates that it is a mnemonic key. You can create mnemonic accelerator keys following this standard when developing Opal applications.

**network**
A series of computers linked by communications hardware and software.

**notification message**
Messages sent automatically by Opal objects or by the operating system when particular events occur. Notification messages may be sent to a single object or to multiple objects. You can use them in message/action pairs to make an object perform actions in response to notification messages it receives. You can also send notification messages to simulate the occurrence of an event. Typical notification messages include all mouse, keyboard, and Windows messages, Page Change Notification messages, Gain Focus or Lose Focus messages, Animation Start, and Animation End messages.

**object**
Anything added to an Opal page. Objects are defined by their behaviors and properties. They can receive messages and perform actions. See also property and behavior.

**ODBC**
A standard software component for accessing database systems. See also SQL.

**Opal Integrator**
The authoring environment used to create Opal applications.

**Opal Player**
A separate program distributed with Opal Integrator. Opal Player is used to run published Opal applications created in Opal Integrator. Opal Player can run on a local computer, through a network using Opal Server, or through the Internet or an intranet using a browser.

**OpalScript**
The native scripting language used in Opal Integrator to incorporate simple expressions into property values and action parameters.

**Opal Server**
Stores all ODBC drivers and other connectivity software in a shared location. This allows the network administrator to set them up one time, in one place, rather than installing them separately on individual computers. Also provides a means of centralized maintenance and upgrades to Opal applications and Opal Player.
page
You add objects to Opal pages to create an Opal application. After publication, each page functions as a single application screen. During development, Opal page files have the extension .SDT. After publication, Opal pages bear the extension .SDO.

parameter
Action parameters are values inserted by the developer to supply specific information necessary to perform an action. For example, to perform a Send Message action, the sending object needs two parameters: the name of the target object and the name of the message to send.

parent screen
In the Data Mapper, a terminal screen with recorded navigational paths leading from it to one or more child screens.

path
In host terminal applications, the sequence of keystrokes required to reach a given terminal screen. The Data Mapper records navigational path data along with screen layout information for each terminal screen in preparation for generating mapped Display objects.

project
A single Opal application during development. Each project consists of one or more pages. Project files have the extension .PRO. See also page and publish.

property
Editable values that control an object’s appearance, size, position, and other characteristics. Object properties can be specified by the theme of the project or page, or by the application designer.

property linking
A technique in which the property of one object is determined dynamically at run time by the property of another. Used as the basis for data transfer. See also property.

protected field
A field on a host terminal screen that does not accept user input. Protected fields typically display labels, headings, and other screen elements that remain constant when screen data is refreshed. See also field and unprotected field.

publish
To compile Opal pages to create a compressed and portable application that can be run from Opal Player.

query
A request for manipulation and retrieval of data from a database. The output of a query is a result set. See also database and SQL.

record
1) A collection of fields with a unique name and specific type in a database. 2) In host terminal integration, the Data Mapper stores or records essential data about an application. The Data Mapper records screen layout and path data about the terminal screens in an application, in preparation for generating mapped Opal objects. See also survey.

relational database
See database.

role
The assigned function of an object within an Opal application. Specifying an object’s role ensures that the object will be successfully transposed if you change the project or page theme. See also object and theme.
run time
The stage at which an application is in use, as opposed to application design, during which the application is still under development. At run time, an Opal application runs in Opal Player or in Run mode in Opal Integrator.

SDO
An Opal file extension denoting a published page that can be distributed and run on Opal Player. See also SDT and publish.

SDT
An Opal file extension denoting a page that has not yet been published. See also SDO and publish.

server
A shared computer that provides storage and other services for other computers connected to it via a network. See also client and client/server.

session
In Opal, a single connection between a personal computer and an external database or host terminal application. See also database session and host session.

shortcut dragging
The technique of using the right mouse button to drag an object. This invokes a menu of options that are specific to the object being dragged. Interchangeable with context dragging.

shortcut menu
A context-sensitive pop-up menu that appears when you right-click a selection, a toolbar, or a taskbar button, and when you right-drag an object on the Properties window, Behavior window, or Data Mapper while they are positioned over the Opal page. The contents of the menu depend on the context.

splash screen
In interactive software, a first screen containing a logo, author credits, and a copyright notice. Splash screens often make use of animation, sounds, and eye-catching graphics.

SQL
A unified programming language for defining, querying, modifying, and controlling the data in a relational database. SQL is often embedded in other programming languages. See also ODBC.

style
A two-dimensional or three-dimensional rendering option for text and borders, such as Plain, Raised, Recessed, or Drop Shadow.

survey
In Opal host application integration, selecting specific elements in the screen layout and path data recorded by the Data Mapper, in preparation for mapping them to objects on the Opal page. See also record.

template
In Opal, an object that is duplicated by the application designer to create additional objects with similar properties and behaviors. Also, an Opal page that is duplicated by the application designer to create more pages with similar objects. See also theme.
**terminal emulator**  
A program that allows a personal computer to connect to a host computer and access host terminal application screens. Interchangeable with communications layer.

**Terminal tab**  
In the Data Mapper, the Terminal tab displays the host application as it would appear from a terminal.

**theme**  
A template for quickly creating Opal applications with a default layout and style.

**trace**  
In Opal, a tool window that logs all events within an Opal application as they occur at run time.

**unprotected field**  
A field in a host-terminal application that accepts user input. See also field and protected field.

**user interface (UI)**  
The aspects of a computer program with which users interact to operate the computer. See also Graphical User Interface.

**VBScript**  
A scripting language developed by Microsoft and supported by Microsoft’s Internet Explorer Web browser. VBScript is based on Visual Basic, but is much simpler. In many ways, it is similar to JavaScript. It enables Web authors to include interactive controls, such as buttons and scrollbars, on their Web pages.

**Web**  
See WORLD WIDE WEB.

**WinHLLAPI**  
An implementation of HLLAPI that is optimized for Windows environments. See also HLLAPI.

**World Wide Web**  
A graphics-intensive environment running on the Internet, through which graphically-enhanced hypertext documents can be accessed through browsers.

**WWW**  
See World Wide Web.
Index

A
about Opal, 1-4
Active X and Internet Explorer 3.02, 5-14
attended installation
  Opal Integrator, 3-3
  Opal Player, 5-5
  Opal Server, 4-3
audience, 1-2

B
behaviors and properties tour, 6-2

C
configuring the Web server configuration for
  Opal Player, 5-7
content summary, 1-3
context-sensitive help, 1-12

D
database applications, 2-7
database integration tour, 6-21

F
features
  Opal Integrator, 1-5
  Opal Player, 1-8
  Opal Server, 1-7
files, Opal Integrator, 2-4
folders, Opal Integrator, 2-4

H
help
  context-sensitive, 1-12
  Internet resources, 1-13
  online, 1-11
  online User Guide, 1-12
  technical support, 1-13

Index-1
host applications, 2-7
host integration tour, 6-37

I

installation methods, 2-3
installation tips, 3-2
installing
  Opal Integrator, 3-1
  Opal Player for Microsoft Internet
  Explorer, 5-4
  Opal Player for Netscape Navigator, 5-4
  Opal Server, 4-3
  stand-alone Opal Player, 5-4
Integrator
  attended installation, 3-3
  database applications, 2-7
  features, 1-5
  host applications, 2-7
  installing, 3-1
  naming objects, 2-6
  objects, 2-6
  project folders and files, 2-4
  requirements, 2-1
  setting preferences, 3-6
  tours, 6-1
  unattended installation, 3-4
Internet applications, running, 5-11
Internet resources, 1-13

N

Netscape Navigator, installing Opal Player,
5-4
network applications, running, 5-12

O

objects
  naming, 2-6
  Opal Integrator, 2-6
online documentation, 3-5
online help, 1-11
online User Guide, 1-12
Opal overview, 1-4
Opal Integrator
  attended installation, 3-3
  database applications, 2-7
  features, 1-5
  host applications, 2-7
  installing, 3-1
  naming objects, 2-6
  objects, 2-6
  project folders and files, 2-4
  requirements, 2-1
  setting preferences, 3-6
  tours, 6-1
  unattended installation, 3-4
Opal Integrator tours, 6-1
  behaviors and properties, 6-2
  database integration, 6-21
  host integration, 6-37
Opal Player
  attended installation, 5-5
  features, 1-8
  installing for Microsoft Internet
  Explorer, 5-4
  installing for Netscape Navigator, 5-4
  installing stand-alone, 5-4

Microsoft Internet Explorer, installing Opal
Player, 5-4
overview, 5-1
proxy server setup, 5-9
requirements, 2-2
running Internet applications, 5-11
running network applications, 5-12
unattended installation, 5-8
Web server configuration, 5-7

Opal Server
attended installation, 4-3
features, 1-7
installing, 4-3
overview, 4-1
requirements, 2-2
unattended installation, 4-4
uninstalling, 4-6

overview
Opal Player, 5-1
Opal Server, 4-1

Player
attended installation, 5-5
features, 1-8
IE 3.02, 5-14
installing for Microsoft Internet Explorer, 5-4
installing for Netscape Navigator, 5-4
installing stand-alone, 5-4
overview, 5-1
proxy server setup, 5-9
requirements, 2-2
running Internet applications, 5-11
running network applications, 5-12
unattended installation, 5-8
Web server configuration, 5-7
preferences, setting for Opal Integrator, 3-6
prerequisite knowledge, 1-2
project folders and files, Opal Integrator, 2-4
proxy server setup, Opal Player, 5-9

Republish projects for IE 3.02, 5-14
requirements
Opal Integrator, 2-1
Opal Player, 2-2
Opal Server, 2-2

Server
attended installation, 4-3
features, 1-7
installing, 4-3
overview, 4-1
requirements, 2-2
unattended installation, 4-4
uninstalling, 4-6
stand-alone Opal Player, installing, 5-4
style conventions, 1-9

technical support, 1-13
tours, Opal Integrator, 6-1

unattended installation
Opal Integrator, 3-4
Opal Player, 5-8
Opal Server, 4-4
uninstalling Opal Server, 4-6
W

Web server configuration, Opal Player, 5-7
welcome to Opal, 1-1

what this guide provides, 1-3
what you should know, 1-2
who should read this guide, 1-2