

OMNIPILOT®
SOFTWARE



Dreamweaver MX Edition

 Lasso 8

Development Studio

Trademarks

Lasso, Lasso Professional, Lasso Studio, Lasso Dynamic Markup Language, Lasso, Lasso Service, Lasso Connector, Lasso Web Data Engine, and OmniPilot are trademarks of OmniPilot Software, Inc. All other products mentioned may be trademarks of their respective holders.

Third Party Links

This guide may contain links to third-party Web sites that are not under the control of OmniPilot. OmniPilot is not responsible for the content of any linked site. If you access a third-party Web site mentioned in this guide, then you do so at your own risk. OmniPilot provides these links only as a convenience, and the inclusion of the links does not imply that OmniPilot endorses or accepts any responsibility for the content of those third-party sites.

Copyright

Copyright © 2005 OmniPilot Software, Inc. This manual may not be copied, photocopied, reproduced, translated or converted to any electronic or machine-readable form in whole or in part without prior written approval of OmniPilot Software, Inc.

Third Edition: September 19, 2005

Version: 8.1

OmniPilot Software, Inc.
1815 Griffin Road
Dania Beach, Florida 33004
U.S.A.

Telephone: (954) 874-3100
Email: info@omnipilot.com
Web Site: <http://www.omnipilot.com>

Contents

Chapter 1

Introduction	9
Lasso Studio 8	9
<i>Figure 1: Lasso Studio Components</i>	10
What's New in 8.1	12
The Lasso 8 Product Line	13
Documentation	15
Customer Support	18
Usage Rights.	18

Chapter 2

Installing Lasso Studio	21
System Requirements	21
Installation Instructions	23
<i>Figure 1: Single-User Installation</i>	23
Getting Started	31
Installation Contents	32
Extended Configuration	34
Uninstalling Lasso Studio.	36

Chapter 3

Configuring Lasso Professional 8	39
Using Lasso Professional 8	39
<i>Figure 1: Lasso Site Administration</i>	41
<i>Figure 2: Lasso Database Browser</i>	43

Data Source Configuration	44
<i>Figure 3: Host Listing</i>	45
<i>Figure 4: Database Listing</i>	46
<i>Figure 5: Host Listing</i>	47
<i>Figure 6: Database Listing</i>	48
User Configuration	50
<i>Figure 7: Group Listing</i>	51
<i>Figure 8: Database Permissions</i>	52
<i>Figure 9: Table Permissions</i>	53
<i>Figure 10: User Listing</i>	54

Chapter 4

Using Lasso Studio	55
Overview	55
<i>Figure 1: Authoring Environment</i>	56
Lasso Studio Menu	57
<i>Figure 2: Lasso Studio Menu</i>	57
Panels	58
<i>Figure 3: Lasso Studio Panel</i>	59
<i>Figure 4: Lasso Form Builder Panel</i>	60
<i>Figure 5: Lasso Inspector Panel</i>	61
<i>Figure 6: Lasso Objects</i>	61
<i>Figure 7: Lasso Updater</i>	62
<i>Figure 8: FileMaker CDML Converter</i>	62
<i>Figure 9: Lasso Shell</i>	63
Screens and Editors	63
<i>Figure 10: Snapshot Editor Screen</i>	64
<i>Figure 11: Site Builder Screen</i>	65
<i>Figure 12: Tag Editor Screen</i>	66
Testing Solutions	66
<i>Figure 13: Groups Page</i>	67
Learning Resources	70

Chapter 5

Using Database Snapshots	73
Overview	73
Creating Snapshots	75
<i>Figure 1: Snapshot Editor</i>	76
Using Snapshot Files	77
<i>Figure 2: Database Selector</i>	78
Editing Snapshots	82
<i>Figure 3: Edit Connection Screen</i>	83

Figure 4: Schema Editor	85
Data Preview	87
Figure 5: Before Data Preview.	88
Figure 6: After Data Preview.	89
Figure 7: Data Preview Settings.	90
Figure 8: Edit Record Set	92

Chapter 6

Using Form Builder 95

Introduction.	95
Using the Form Builder	97
Figure 1: Database Selector.	97
Figure 2: Lasso Form Builder.	98
Searching Records	100
Figure 2: Lasso Form Builder Search Records	102
Figure 3: Lasso Form Builder List Records	104
Figure 4: Lasso Form Builder Display Records.	106
Adding Records	107
Figure 5: Lasso Form Builder Add Records	108
Figure 6: Lasso Form Builder Add Response	110
Updating and Deleting Records.	111
Figure 7: Lasso Form Builder Update Records.	112
Figure 8: Lasso Form Builder Update Response.	114

Chapter 7

Using Site Builder. 115

Overview	116
Figure 1: Lasso Site Builder.	116
Figure 2: Site Types	117
Search Site Creation	118
Figure 3: Search Site	118
Figure 1: Lasso Site Builder Database Selector	120
Figure 5: Lasso Site Builder Create Search Page	121
Figure 6: Lasso Site Builder Create Listing Page	122
Figure 7: Lasso Site Builder Create Display Page	123
Figure 8: Lasso Site Builder Create Update Page.	124
Add Site Creation.	125
Figure 9: Add Site	125
Figure 9: Lasso Site Builder Database Selector	127
Figure 11: Lasso Site Builder Create Add Page	128
Figure 12: Lasso Site Builder Create Display Page	129
Figure 13: Lasso Site Builder Create Update Page.	131

Site Builder Pages	132
<i>Figure 14: Search.lasso</i>	133
<i>Figure 15: Listing.lasso</i>	134
<i>Figure 16: Display.lasso</i>	136
<i>Figure 17: Add.lasso</i>	137
<i>Figure 18: Update.lasso</i>	138
<i>Figure 19: AddResponse.lasso</i>	139
<i>Figure 20: UpdateResponse.lasso</i>	140
Testing Your Site	141
Site Builder Templates.	142
<i>Figure 21: Site Builder Template</i>	143

Chapter 8

Using Objects and Tags145

Overview	145
<i>Figure 1: Lasso Objects</i>	146
<i>Figure 2: Records Container Tag Objects</i>	149
Lasso Data Access Objects	152
<i>Figure 3: Lasso Data Access Objects</i>	152
Lasso Programming Objects.	156
<i>Figure 4: Lasso Programming Objects</i>	156
Lasso Form Objects	162
<i>Figure 5: Lasso Form Objects</i>	162
Using Lasso Objects	166
<i>Figure 6: Lasso Data Access Objects</i>	167
<i>Figure 7: Inline Example</i>	167
<i>Figure 8: Inline Inspector</i>	168
<i>Figure 9: Lasso Tag Editor</i>	169
<i>Figure 10: Lasso Parameter Editor</i>	170
<i>Figure 11: Inline Example</i>	171
<i>Figure 12: Inline Example</i>	171
Lasso Display Preferences	172
<i>Figure 13: Lasso Display Preferences</i>	172

Chapter 9

Using Lasso Inspectors175

Overview	175
Using the Lasso Inspector.	176
<i>Figure 1: Lasso Tag Inspector</i>	177
<i>Figure 2: Lasso Form Inspector</i>	179
<i>Figure 3: LassoScript/Expression Inspector</i>	180
Substitution and Process Tags	180

<i>Figure 4: Lasso Inspector</i>	181
<i>Figure 5: Lasso Inspector</i>	183
Container Tags	183
<i>Figure 6: Lasso Inspector</i>	185
<i>Figure 7: Code Preview</i>	185
<i>Figure 8: Code Preview</i>	186
Member Tags	186
<i>Figure 9: Lasso Inspector</i>	187
Form Elements	188
<i>Figure 10: Lasso Inspector</i>	188
LassoScript and Expressions	189
<i>Figure 11: Lasso Inspector</i>	190
<i>Figure 12: Lasso Inspector</i>	191

Chapter 10

Using Lasso Tag Editors	193
Tag Editor	193
<i>Figure 1: Tag Editor</i>	194
Parameter Editor	195
<i>Figure 2: Lasso Parameter Editor</i>	196
Tag Selector	199
<i>Figure 3: Lasso Tag Selector</i>	199

Chapter 11

Using Lasso Updater and CDML Converter	201
Using Lasso Updater	201
<i>Figure 1: Lasso Updater</i>	202
Using FileMaker CDML Converter	203
<i>Figure 2: FileMaker CDML Converter</i>	203

Chapter 12

Using Lasso Shell	205
Lasso Shell	205

Appendix A	
License Agreement	209
Appendix B	
Glossary.....	213
Appendix C	
Index	223

1

Chapter 1

Introduction

This chapter provides an overview of Lasso Studio and the Lasso 8 product line, including new features, documentation, and other useful information. It is recommended that all users read this chapter.

- *Lasso Studio 8* introduces the features of Lasso Studio, and describes the available editions.
- *What's New* describes new features introduced in Lasso Studio.
- *The Lasso 8 Product Line* introduces the full range of Lasso 8 products and how they relate to Lasso Studio.
- *Documentation* describes the documentation included with Lasso Studio.
- *Customer Support* introduces the resources available to help you set up and use Lasso Studio.
- *Usage Rights* includes important information about usage rights for Lasso Studio.

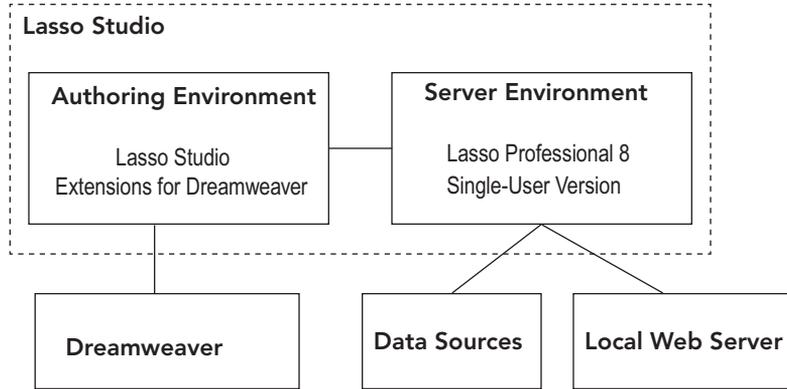
Lasso Studio 8

Welcome to Lasso Studio, the professional-grade authoring tool for Lasso Professional. Lasso Studio gives you the ability to create powerful data-driven Web sites using the visual tools of Macromedia Dreamweaver, and the ability to test them with an included single-user version of Lasso Professional Developer.

Components

Lasso Studio consists of two main components, which are the authoring environment and server environment.

Figure 1: Lasso Studio Components



Authoring Environment

The authoring environment is the Lasso Studio component that works within Dreamweaver and constitutes the Lasso Studio product. The features of the Lasso Studio authoring environment are described in the *Features* section below.

Server Environment

An included single-user license of Lasso Professional Developer allows you to test your Lasso solutions on your local computer before uploading them to a production server, and also provides the Lasso Studio authoring environment with information about eligible databases. This feature interfaces with the built-in Apache Web server or WebSTAR V on Mac OS X or the built-in IIS Web server on Windows, and operates separately from Dreamweaver. A connection to a Lasso Professional server environment is required for the data access features within the Lasso Studio authoring environment to function.

Lasso Professional Developer included with Lasso Studio is identical to Lasso Professional Server, but goes into single-user mode when initialized with a Lasso Studio serial number. Single-user mode imposes the following restrictions over a full multi-user license of Lasso Professional Server:

- Lasso Service will only serve to the first two client IP addresses which connect via a Web browser.

- Only 500 connections (or Lasso Service actions) are allowed per minute.
- The information bar in Lasso Site Administration says Single-User instead of Multi-User.

Features

This section describes the features of the authoring environment provided in Lasso Studio. For a description of features in Lasso Professional 8, see the *Lasso Professional 8* chapter in the Lasso Professional 8 Setup Guide.

Lasso Studio provides the following features:

- **Server Connections and Snapshots** – Instantly connect Dreamweaver to a Lasso Professional server to retrieve database schema information (i.e. database, table, and field names) for use in Lasso pages, or Web pages. This information can be saved in Lasso Studio as snapshot files, which can be swapped or edited for different projects and shared with other Lasso Studio users.
- **Lasso Shell** – Execute or check the syntax of Lasso code within a Dreamweaver panel.
- **Lasso 8 Syntax** – Support for Lasso 8 syntax including both colon and parentheses syntax, inline operators, LassoScript, and new Lasso 8 tags.
- **Syntax Coloring and Code Completion** – Lasso code is shown with syntax coloring within source views. Code completion makes it easy to select Lasso tags by typing just the first few characters.
- **Data Preview** – Instantly view and edit the visually-displayed properties of stored data in a database in the Design view of Dreamweaver. This feature allows you to develop and edit dynamic data content while seeing exactly what the end-user will see.
- **Form Builder** – The Lasso Studio Form Builder allows you to rapidly develop dynamic Web sites by creating forms based on your current database configuration for the most common Lasso actions.
- **Site Builder** – The Lasso Studio Site Builder allows you to create an entire form-based Web site based on your current database configuration using a guided, dialog-driven approach.
- **Drag-and-Drop Lasso Objects** – Drag and drop Lasso Objects representing Lasso tags and form elements for programming Lasso operations within a page, or edit Lasso code in the Code view of Dreamweaver. Drag-and-drop Lasso Objects include data access tags, conditional logic, variables, math operations, string operations, list manipulations, file input/output, and more.
- **Inspectors** – Lasso Inspectors allow you to see and modify the attributes of any Lasso tag or Lasso-enabled form element. The Lasso Inspectors

provide access to the most common attributes of tags and allow you to launch the Lasso Tag Editors for more complete control over complex tags.

- **Tag Editor** – The Lasso Tag Editor gives you more control over powerful tags such as the Inline tag and allows you to embed sub-tags so you can build complex expressions.
- **Lasso Updater** – The Lasso Updater allows instant conversion from previous versions of Lasso to Lasso 8 code. The Lasso Updater can perform a batch conversion on an entire folder of Lasso pages or on a single Lasso page.
- **FileMaker CDML Converter** – The FileMaker CDML Converter allows instant conversion from CDML code to Lasso 8 code. The CDML Converter can perform a batch conversion on an entire folder of CDML pages or on a single CDML page.

What's New in 8.1

This section outlines what has changed in Lasso Studio 8 since Lasso Studio 7.

Installation

Lasso Studio is now installed using the Macromedia Extension Manager. The extension manager should launch automatically at the end of the Lasso Studio 8 installer. Simply accept the license agreement to install Lasso Studio into Dreamweaver 8 or Dreamweaver MX 2004.

If the extension manager does not launch automatically, locate the Lasso Studio 8.1 for Dreamweaver.mxp file in the Lasso Studio 8 application folder and open it with the Macromedia Extension Manager application.

If necessary, Lasso Studio can be temporarily disabled or permanently uninstalled using the extension manager.

New Features

The following features are new to Lasso Studio 8, and not available in previous versions of Lasso Studio.

- **Lasso Professional Developer 8** – A single-user testing version of Lasso Professional Developer is included with all editions of Lasso Studio. This can be used to set up a full-featured testing environment for testing Lasso solutions built using Lasso Studio, and provides Lasso Studio with database schema information.

- **Lasso Shell** – Execute or check the syntax of Lasso code within a Dreamweaver panel.
- **Lasso 8 Syntax** – Support for Lasso 8 syntax including both colon and parentheses syntax, inline operators, LassoScript, and new Lasso 8 tags.
- **FileMaker CDML Converter** – The FileMaker CDML Converter is included as a built-in component of Lasso Studio.

Updated Features

- **Lasso Updater** – The Lasso Updater can now perform a batch conversion on an entire folder of Lasso pages or on a single Lasso page.
- **Lasso Parser** – The tag parser within Lasso Studio has been rewritten to recognize parentheses syntax and to better preserve white space which is contained within Lasso tags.
- **Parentheses Syntax** – The Lasso Studio Palette now includes a preference for whether colon or parentheses syntax should be used throughout Lasso Studio. Most components of Lasso Studio will default to this preference when generating code including object dragged from the object palette, new tags edited in the inspector, and code created by the Site Builder and Form Builder.
- **Snapshots** – The caching of values lists in snapshots is now optional. This will reduce the size of snapshot files and increase their performance. A new control in the Lasso Studio Palette and Snapshot Editor makes it easy to fetch values lists when necessary.

The Lasso 8 Product Line

This section provides an overview of the Lasso product line, and shows how Lasso Studio fits in. The Lasso product line consists of a set of applications and connectors that enable you to build data-driven Web sites and serve them via a Web server.

Lasso Professional Server 8

Lasso Professional Server 8 is the product for serving Lasso-based data-driven Web sites. It consists of a core Lasso Service application and several Lasso Connectors which establish links to Web servers and data sources. Lasso Professional 8 is available for Mac OS X and Windows. Lasso Professional 8 has the following components.

- **Lasso Service** – The core application or Web Data Engine. Includes a built-in SQLite data source.

- **Lasso Connector for Apache** – Allows Lasso solutions to run via the built-in Apache Web server in Mac OS X.
- **Lasso Connector for IIS** – Allows Lasso solutions to run via the built-in IIS Web server in Windows.
- **Lasso Connector for MySQL** – Allows Lasso to access MySQL data sources.
- **Lasso Connector for FileMaker Pro** – Allows Lasso to access FileMaker Pro 4.x, 5.x, and 6.x data sources.
- **Lasso Connector for JDBC** – Allows Lasso to access JDBC-compliant data sources, including Microsoft SQL Server, PostgreSQL, Frontbase, Oracle, and others.

For complete documentation of the Lasso Professional 8 architecture and features, see the *Lasso Professional 8* chapter in the Lasso Professional 8 Setup Guide.

Lasso Professional Developer 8

Lasso Professional Developer is a single-user edition of Lasso Professional Server. It has identical features except for two limitations. Only two IP addresses can view Web sites served by Lasso Professional Developer and only 500 transactions will be served per minute.

Lasso Studio 8

Lasso Studio is an optional development tool for visually building data-driven Web sites within either Adobe GoLive or Macromedia Dreamweaver. Lasso Studio consists of a single-user version of Lasso Professional Developer, plus a set of extensions for either Macromedia Dreamweaver or Adobe GoLive CS. Solutions built using Lasso Studio are deployed using the full, multi-user version of Lasso Professional Server 8.

Lasso Studio for Eclipse

Lasso Studio for Eclipse is a development tool which installs into the Eclipse integrated development environment and provides advanced tools for creating and debugging Lasso pages. Lasso Studio for Eclipse includes an integrated debugger, advanced editing and outlining tools, and more. Solutions built using Lasso Studio are deployed using the full, multi-user version of Lasso Professional Server 8.

Documentation

This chapter contains information about the documentation included with Lasso Studio, and describes recommended usage. Comments, suggestions, or corrections to the documentation are appreciated and may be sent to the following email address.

lassodocumentation@omnipilot.com

Lasso Studio Documentation

The documentation for Lasso Studio is divided into several different references. All documentation, with the exception of the Lasso Reference, is included in PDF format in the Documentation folder inside the Lasso Studio 8 application folder installed to your hard drive. The following manuals and resources are included.

- **Lasso Studio 8 User Guide** – This book includes all documentation for the Lasso Studio authoring environment, and it is recommended that this guide be read first. All documentation that follows is specific to the included single-user version of Lasso Professional.
- **Lasso Professional 8 Setup Guide** – This book includes documentation of the architecture and features of Lasso Professional, the administration interface, and Lasso Security topics.
- **Lasso 8 Language Guide** – The documentation of Lasso (Lasso Dynamic Markup Language), the language used to access data sources, specify programming logic, and much more.
- **Lasso Reference** – Provides detailed documentation of each tag in Lasso 7. This is the definitive reference to the language of Lasso 8. This reference is provided as a LassoApp installed with Lasso Professional, and is also available as an online resource from the OmniPilot Web site.

<http://ldml.omnipilot.com/>

Lasso Studio 8 User Guide

This is the book you are reading. This book contains the following chapters which detail how to install and use Lasso Studio.

- *Chapter 1: Introduction* introduces Lasso Studio and the Lasso 8 product line.
- *Chapter 2: Installing Lasso Studio* provides step-by-step instructions for installing Lasso Studio for Dreamweaver.

- *Chapter 3: Configuring Lasso Professional 8* provides step-by-step instructions for installing and configuring the single-user testing version of Lasso Professional 8 included with Lasso Studio.
- *Chapter 4: Using Lasso Studio* provides an introduction to using the authoring environment of Lasso Studio.
- *Chapter 5: Using Database Snapshots* describes creating server connections between Lasso Studio and Lasso Professional 8, and describes creating database snapshot files.
- *Chapter 6: Using Form Builder* describes using the Lasso Studio Form Builder for creating simple, form-based data-driven solutions.
- *Chapter 7: Using Site Builder* describes using the Lasso Studio Site Builder for creating more complex data-driven solutions.
- *Chapter 8: Using Objects and Tags* describes using Lasso Objects and writing Lasso code in Lasso Studio.
- *Chapter 9: Using Lasso Inspectors* describes the function and usage of Lasso Inspectors in Lasso Studio.
- *Chapter 10: Using Lasso Tag Editors* describes using the Lasso Tag Editor in Lasso Studio.
- *Chapter 11: Using Lasso Updater and FileMaker CDML Converter* describes using the Lasso Updater and FileMaker CDML Converter in Lasso Studio.
- *Appendix A: License Agreement* presents the license agreement for Lasso Studio. The Lasso Studio license agreement must be read and agreed to before installing and using Lasso Studio.
- *Appendix B: Glossary* provides a glossary of key terms used within the Lasso Studio 8 User Guide. If you do not understand the meaning of a word when using this guide, please refer to the glossary.

Documentation Conventions

This documentation uses several conventions in order to make finding information easier.

Definitions are indicated using a bold, sans-serif type face for the defined word. This makes it easy to find defined terms within a page. Terms are defined the first time they are used.

Cross References are indicated by an italicized, sans-serif typeface. For instance, the next section in this chapter is *Customer Support*.

Code is formatted in a narrow, sans-serif font. Code includes HTML tags, Lasso tags, and any text which should be typed into a Lasso page. Code is

represented within the body text (e.g., [Field] or <body>), or is specified in its own section of text as follows:

```
[Field: 'Company']
```

Code Results represent the results after code is processed. They are indicated by a black arrow, and will usually be the value that is sent to the client's Web browser. The following text could be the result of the code example above.

```
→ OmniPilot
```

Lasso tags always appear in code format exactly as they appear except when referring to a group of related tags. For example, -Encoding... refers to all of the encoding keywords (-EncodeNone, -EncodeHTML, etc.). Also, member tags are always referred to beginning with their data type followed by the member tag symbol. For example, the member tag for specifying a character in a text string is referred to as [String->Get].

File Paths can be local or fully-qualified file paths on either Windows or Mac OS X, and are formatted using the same font as code. File paths are represented within the body text (e.g., Lasso Professional 8/Admin/Setup or C:\inetpub\wwwroot), or are specified in their own section of text as follows:

```
C:\Program Files\OmniPilot Software\Lasso Professional 8
```

File paths in Mac OS X contain forward slashes (/), while file paths for Windows contain backward slashes (\). If a file path is identical on both Mac OS X and Windows, then the file path will be shown once with forward slashes (/). Users of the command line utility in Windows will need to enter backward slashes instead of forward slashes for these examples.

Note: Notes are included to call attention to items that are of particular importance or to include comments that may be of interest to select readers. Notes may begin with **Warning**, **Tip**, **FileMaker Pro Tip**, **IIS Tip**, etc. to specify the importance and audience of the note.

To perform a specific task:

The documentation assumes a task-based approach. The contents following a task heading will provide step-by-step instructions for the specific task.

Customer Support

There are many resources available to help you when installing and using Lasso Studio. It is recommended that the following resources be used in the order in which they are presented below.

To find answers to questions about Lasso Studio 8:

- 1 Documentation** – The documentation should be your first resource. Check to see if the Lasso Studio 8 User Guide, Lasso Professional 8 Setup Guide, Lasso 8 Language Guide, or Lasso Reference have the answer to your questions.
- 2 OmniPilot Support Central** – A Web resource that allows you to search for updates to the documentation, software updates, and tips about how to best utilize Lasso products.
<http://support.omnipilot.com>
- 3 Lasso Studio Talk Email Discussion List** – A broad community of Lasso Studio users who help to answer each other's questions about using Lasso Studio products. If you can't find an answer to your question, then posting to the list will often result in an answer in a short period of time. Information about subscribing to the list and searchable list archives can be found at the following address.
<http://listsearch.omnipilot.com/lassostudiotalk.lasso>
- 4 Email and Phone Support** – Provided by OmniPilot to qualified customers during normal business hours. Visit the following URL for more information.
<http://support.omnipilot.com/>

Usage Rights

These are the usage rights for Lasso Studio. Please consult the Lasso Professional 8 Setup Guide for important information about the usage rights and license agreements specific to those products.

New Purchase

Your license permits a single copy of Lasso Studio to be installed and used on a single computer. While certain components of Lasso Studio (i.e. the single-user testing version of Lasso Professional Developer) may be installed on a separate computer from the authoring environment of Lasso Studio, only a single instance of each component is permitted. The

license does not permit development or deployment using non-purchased versions or evaluation versions.

Upgrade Purchase

It is standard industry practice and understood that by upgrading one's software one no longer uses the old version and the license to use and transfer said license ceases.

If you have upgraded to Lasso Studio 8 from any previous version of Lasso, you must no longer use that version. Please see the termination provisions in the accompanying license agreement for further details. Alternatively, you could purchase a new Lasso product license and not be bound by such upgrade restrictions.

Evaluation Versions

Evaluation versions are provided for one-time 30-day evaluation and initial product testing use. Evaluation versions are not licensed for use for extended development. The documentation provided with evaluation versions is to be used strictly within the evaluation time period.

2

Chapter 2

Installing Lasso Studio

This chapter provides a step-by-step guide to installing and initializing the Lasso Studio Dreamweaver extensions (i.e. authoring environment) and single-user version of Lasso Professional Developer (i.e. server environment) included with Lasso Studio.

- *System Requirements* lists the minimum system requirements for Lasso Studio for Dreamweaver.
- *Installation Instructions* includes step-by-step instructions for installing a default installation of Lasso Studio, which consists of installing the Lasso Studio authoring environment and the included single-user version of Lasso Professional Developer on the same machine.
- *Getting Started* provides instructions for what to do after Lasso Studio is first installed and initialized.
- *Installation Contents* lists the files installed with the authoring and server environments in Lasso Studio.
- *Extended Configuration* describes installing and connecting the Lasso Studio authoring environment to machines running the full multi-user version of Lasso Professional Server.
- *Uninstalling Lasso Studio* includes step-by-step instructions for removing Lasso Studio from your system.

System Requirements

Lasso Studio will run on systems which meet the minimum requirements listed below. These requirements are for both the Lasso Studio authoring environment (i.e. Dreamweaver extensions), and the included single-user

version of Lasso Professional Developer. Although Lasso Studio may run on machines which do not meet these requirements, these installations are not officially supported.

Deployment Note: To deploy Lasso solutions created using Lasso Studio to the Web, a Web server running a full multi-user license of Lasso Professional Server is required, and is purchased separately from Lasso Studio. If you are not hosting your own server, you may serve your solutions via a Lasso ISP. OmniPilot maintains a list of Lasso ISPs at <http://www.lassohost.com>.

Mac OS X

- G4, G3, or Power Macintosh (or compatible) computer.
- 128 MB of RAM. More recommended.
- Standard installation of Mac OS X (10.3 or higher) or Mac OS X Server (10.3 or higher). This should include an HFS+ formatted hard drive with BSD subsystem option. UFS systems are not supported for the server environment.
- Default Mac OS X installation of Apache Web Server (included with Mac OS X). Required for the server environment.
- Standard installation of Macromedia Dreamweaver 8 or Dreamweaver MX 2004 for Mac OS X. Required for the authoring environment.
- X11 for Mac OS X 10.3 must be installed in order for the [Image] tags in Lasso Professional 8 to function. For information on obtaining and installing X11, see <http://www.apple.com/macosx/x11/>.

Note: The [Image] tags are not supported on Mac OS X 10.2 as X11 is not officially provided or supported by Apple for Mac OS X 10.2 as of the release of this software.

- Adobe Acrobat 5.0 or higher to view the electronic documentation.
- Monitor capable of 800 x 600 resolution.

Windows 2000/2003/XP

- 300 MHz or higher Pentium-compatible CPU.
- 128 MB of RAM. More recommended.
- Microsoft Windows 2000, Windows XP Professional, or Windows 2003 Server.
- Microsoft Internet Information Services (IIS) 5 or higher. IIS 5 is included with Windows 2000 and Windows XP Professional, and can

be installed from the Windows installation CD. Required for the server environment.

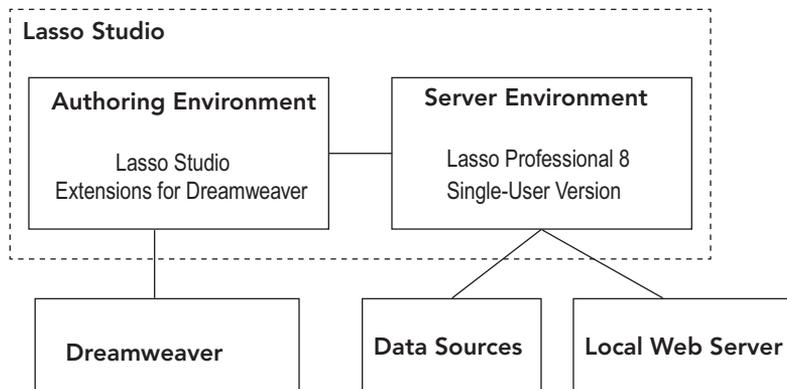
- Java Virtual Machine compatible with Sun’s JRE 1.4 (Java 2). Required for Java-based components in the server environment. See the *Configuring on Windows* chapter in the Lasso Professional 8 Setup Guide for Java installation instructions.
- Standard installation of Macromedia Dreamweaver 8 or Dreamweaver MX 2004 for Windows. Required for the authoring environment.
- Standard installation of ImageMagick, which can be downloaded at: <http://lassodownload.blueworld.com/pub/Lasso8/>.
- Adobe Acrobat 5.0 or higher to view the electronic documentation.
- Monitor capable of 800 x 600 resolution.

Installation Instructions

This section describes installing a default installation of Lasso Studio. A **Default Installation** is where both the Lasso Studio extensions for Dreamweaver (i.e. authoring environment) and included single-user version of Lasso Professional Developer (i.e. server environment) are installed on the same machine. This is referred to as a local single-user testing configuration, and is recommended for most users.

For best results, please perform the procedures in the following sections in the order they are presented.

Figure 1: Single-User Installation



Before installing a default installation of Lasso Studio, verify the following.

- The machine you are installing to meets all system requirements outlined in the *System Requirements* section of this chapter.
- No existing version of Lasso Studio is installed on the same machine. For information on upgrading a previous version of Lasso Studio, see the *Upgrading* section of this chapter.
- No existing version of Lasso Professional is installed on the same machine. For instructions on installing only the Lasso Studio authoring environment to a machine with an existing installation of Lasso Professional, see the *Extended Configuration* section of this chapter.

Upgrading

Lasso Studio 8 can read snapshot files and preferences created with previous versions of Lasso Studio.

However, the included Lasso Professional Developer requires using the included Lasso Settings Migrator for upgrades. Please read the instructions for the Lasso Settings Migrator included in the Lasso Studio download package before you uninstall the prior version of Lasso Studio.

To upgrade to Lasso Studio 8 on Mac OS X:

Consult the documentation of the Lasso Studio Migrator for details about upgrading the included Lasso Professional Developer installation.

First, uninstall the existing version of Lasso Studio. Do this by using the Uninstall Lasso Studio application which is included with the Lasso Studio installer.

Install Lasso Studio 8 using the Lasso Studio installer application. Any existing snapshot files will remain on the system and can be used with Lasso Studio.

To upgrade to Lasso Studio 8 on Windows:

Consult the documentation of the Lasso Studio Migrator for details about upgrading the included Lasso Professional Developer installation.

First, uninstall the existing version of Lasso Studio. Do this by using the Uninstall option in the Lasso Studio installer application, or by using Add/Remove Programs in the Windows Control Panel. After uninstallation, leave the remaining files on the system.

Then, install and initialize Lasso Studio as described in the following *Installing Lasso Studio* and *Initialization* sections.

Installing Lasso Studio

Follow this procedure to install both the Lasso Studio Dreamweaver extensions (i.e. authoring environment) and included single-user version of Lasso Professional Developer (i.e. server environment) on your Mac OS X or Windows machine.

To install Lasso Studio 8 on Mac OS X:

- 1 Before using the installer, make sure that Dreamweaver is not running.
- 2 Double-click the Lasso Studio 8 for Dreamweaver Installer application. Mac OS X will prompt for an administrator name and password to perform the installation.
- 3 Enter the username and password of a Mac OS X user on the machine who has administrative rights. This launches the Lasso Studio 8 installation window.
- 4 Select Continue. This will display the license agreement.
- 5 After reading and agreeing to the terms, select the Accept button. This will display the Lasso Studio Release Notes. The release notes contain important late-breaking information that might not be covered in the documentation.
- 6 After reading the release notes which describe what is included with Lasso Studio, select Continue.
- 7 From the pull-down menu, select Easy Install. This will install the Lasso Studio extensions of Dreamweaver (i.e. authoring environment) and included single-user version of Lasso Professional Developer (i.e. server environment) on the machine you are using.
- 8 Select Install to install Lasso Studio.
- 9 Select Quit when the installer has completed. Upon quit the installer automatically restarts the Apache Web server and starts the Lasso Professional server environment.
- 10 The Macromedia Extension Manager will be launched to install the Lasso Studio 8.1 for Dreamweaver.mxp file from the Lasso Studio 8 application folder. Simply agree to the license agreement to complete the installation.

If Dreamweaver 8 is found then Lasso Studio will be installed into it automatically. The extension manager can be used to also or alternately install Lasso Studio into Dreamweaver MX 2004.

Important: If the Macromedia Extension Manager is not automatically launched then you must manually open the `Lasso Studio 8.1 for Dreamweaver.mxp` file in order to complete the installation of Lasso Studio.

After the installer quits, launch your Dreamweaver application to initialize Lasso Studio as described in the *Initialization* section.

To install Lasso Studio 8 on Windows:

- 1 Before using the installer, make sure that Dreamweaver is not running.
- 2 Double-click the Lasso Studio 8 for Dreamweaver Installer.msi application. This launches the Lasso Studio 8 for Dreamweaver Setup window.
- 3 Select Next >. This will display the license agreement.
- 4 After reading and agreeing to the terms, select the I accept the license agreement radio button and select Next >. This will display the Lasso Studio Release Notes. The release notes contain important late-breaking information that might not be covered in the documentation.
- 5 After reading the release notes which describe what is included with Lasso Studio, select Next >.
- 6 Select the drive and destination folder to which Lasso Studio will be installed and select Next >. The drive and destination folder is C:\Program Files\OmniPilot Software\ by default. To ensure all features in Lasso Studio to work properly, it is strongly recommended that the \Program Files\OmniPilot Software\ destination folders not be changed.
- 7 Select the Typical radio button and select Next >. This will install the Lasso Studio extensions for Dreamweaver (i.e. authoring environment) and included single-user version of Lasso Professional Developer (i.e. server environment) on the machine you are using.
- 8 Select Next > to install Lasso Studio.
- 9 Select Finish when the installer has completed. Upon quit, the installer automatically restarts IIS and starts the Lasso Professional server environment.
- 10 The Macromedia Extension Manager will be launched to install the Lasso Studio 8.1 for Dreamweaver.mxp file from the Lasso Studio 8 application folder. Simply agree to the license agreement to complete the installation.

If Dreamweaver 8 is found then Lasso Studio will be installed into it automatically. The extension manager can be used to also or alternately install Lasso Studio into Dreamweaver MX 2004.

Important: If the Macromedia Extension Manager is not automatically launched then you must manually open the Lasso Studio 8.1 for Dreamweaver.mxp file in order to complete the installation of Lasso Studio.

After the installer quits, launch your Dreamweaver application to initialize Lasso Studio as described in the *Initialization* section.

Initialization

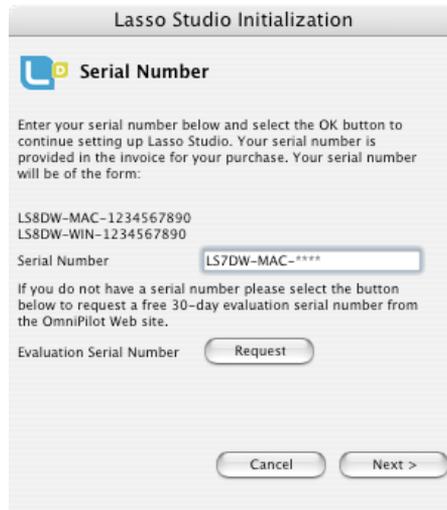
This section describes initializing both the authoring environment and server environment of Lasso Studio, which is done via a single interface within Dreamweaver. The Initialize Lasso Studio... command in the Lasso Studio menu is where Lasso Studio initialization is performed. This involves the following:

- Entering the Lasso Studio serial number, which is used by both the authoring environment and single-user version of Lasso Professional Developer (i.e. server environment).
- Initializing Lasso Professional Developer by specifying a server administrator username and password, email preferences, and a site administrator username and password.
- Entering the site administrator username and password into Lasso Studio.
- Creating the initial connection between the authoring environment and server environment, and creating a snapshot file for the first time.

To initialize Lasso Studio 8:

- 1 Launch the Dreamweaver application in which the Lasso Studio authoring environment is installed.
- 2 From the Lasso Studio menu, select Initialize Lasso Studio.... This will show the Lasso Studio Initialization screen.

Important: If you don't see a Lasso Studio menu within Dreamweaver then you must manually open the Lasso Studio 8.1 for Dreamweaver.mxp file in order to complete the installation of Lasso Studio. If you have both Dreamweaver 8 and Dreamweaver MX 2004 installed make sure that Lasso Studio was Installed into both using the Macromedia Extension Manager.

Figure 2: Initialization Serial Number

- 3 Enter the Lasso Studio serial number in the Serial Number field. This must be entered exactly as it appears in your OmniPilot invoice using one of the following formats:

LS8DW-MAC-123456789
 LS8DW-WIN-123456789

Serial numbers are case-sensitive. Make sure that all capitalization is correct and that all letters and numbers are entered, including the LS6GL prefix. A single 30-day evaluation serial number may be requested by selecting the Request button, which loads the evaluation request page on the OmniPilot Web site in a Web browser.

Note: A default Web browser must be configured in Dreamweaver in order for the Request button to work. Refer to the Dreamweaver documentation for instructions on how to configure a default Web browser.

- 4 Select the Next > button. This shows the Host Address screen.

Figure 3: Initialization Host Address

- 5 In the Host Address field, enter the IP address or host name of the machine to which the Lasso Professional server environment is installed. Use 127.0.0.1 if the server environment is on the machine you are currently using (default installation).
- 6 Select the Lasso Site Administration button to visit the Server Administration initialization screen. Enter the Lasso Studio serial number to initialize Lasso Professional Developer. Select a server administrator username and password and a site administrator username and password. Note the site administrator username and password to enter into the next screen of Lasso Studio's initialization process.
- 7 Select the Next > button. This shows the Username and Password screen.

Figure 4: Initialization Username and Password

Lasso Studio Initialization

Username and Password

Enter the username and password to connect to Lasso Professional.

For a default installation, enter the site administrator username and password which were used to initialize the default site of Lasso Professional.

Username:

Password:

- 8 Enter the site administrator username and password. These allow Lasso Studio to connect to Lasso Professional Developer and retrieve information about the databases that are currently available.
- 9 Select the Next > button. This causes Lasso Studio to connect to the Lasso Professional server environment for the first time. Lasso Studio stores the connection information and initial database schema information retrieved from the server environment in a snapshot file, which is described in the next step.

Warning: The initial connection to the Lasso Professional server environment can take between several seconds and several minutes depending on how many data sources have been configured in Lasso Professional 8. Do not cancel the connection process prematurely if it takes more than a minute.

- 10 After Lasso Studio is able to successfully connect to a Lasso Professional server environment, the Name Snapshot File screen is shown, which asks you to enter a name for the snapshot file that was just created. Enter a name for your snapshot in the Snapshot field.

Figure 5: Initialization Name Snapshot File

- 10 Select the Next > button. After the snapshot is saved, a Lasso Studio Initialization Complete screen is shown.

In the event of an error during initialization, make sure that the Lasso Professional server environment has started properly as described in the *Configuring Lasso Professional 8* chapter *Using Lasso Professional 8* section. Otherwise, make sure that the installer was able to successfully configure the server environment for Apache on Mac OS X or IIS on Windows, as described in the installation chapters of the Lasso Professional 8 Setup Guide.

Lasso Studio has now been installed and initialized. See the following *Getting Started* section for instructions on what to do next.

Getting Started

This section provides an overview of getting started with Lasso Studio, including data source configuration and building an example solution.

To get started with Lasso Studio:

- 1 For setting up a database for use with Lasso Studio, refer to the *Configuring Lasso Professional 8 v Data Source Configuration* section. All data sources must be configured in your server environment before

Lasso Studio can use them. This chapter also provides general instructions for starting and stopping the server environment.

- 2 For an overview of the Lasso Studio authoring environment interface, refer to the *Using Lasso Studio* chapter. This chapter also provides information on how to use the rest of this guide.

Installation Contents

This section contains a list of all files and folders that are installed during Lasso Studio installation. This includes both the Lasso Studio authoring environment and single-user version of Lasso Professional 8.

Lasso Studio 8 Files

The following lists all files installed with the authoring environment of Lasso Studio. This includes the Lasso Studio 8 folder, and the core files installed within Dreamweaver.

Lasso Studio 8 Folder

Lasso Studio installs the following files and folders in the /Applications/Lasso Studio 8 folder in Mac OS X, or in the Program Files\OmniPilot Software\Lasso Studio 8 folder in Windows.

- **Dreamweaver alias** – An alias is created for the Dreamweaver application file. The Lasso Studio authoring environment is automatically launched whenever Dreamweaver is started.
- **Documentation folder** – Contains three sub-folder which contain the documentation for Lasso Studio.
 - 1 - Read Me First – Contains License Agreement.txt and Release Notes.txt files, which are also displayed within the Lasso Studio installer as part of the installation process.
 - 2 - Authoring Environment – Contains the Lasso Studio 8 User Guide - Dreamweaver Edition.pdf file, which is the manual you are reading now.
 - 3 - Server Environment – This folder is a dynamic link to the local Lasso Professional 8/Documentation folder, which contains all documentation for the Lasso Professional server environment in PDF format.
- **Extensions folder** – Contains the Lasso Studio Connector folder, which contains the LassoStudio.LassoApp file used for connecting Lasso Studio to external versions of Lasso Professional as described in the *Extended Configuration* section of this chapter. The source code for the LassoStudio.LassoApp file is also provided in the included Source Code folder.

- **Lasso Site Administration.htm** – When loaded in a Web browser, launches Lasso Site Administration for the local installation of Lasso Professional.
- **Lasso Studio 8.1 for Dreamweaver.mxp** – This Macromedia Extension Manager file is used to perform the installation of the Lasso Studio extensions for Dreamweaver. This file is normally installed automatically at the end of the Lasso Studio installer, but can also be installed manually.
- **Lasso Professional 8 folder** – Provides a shortcut to the local Lasso Professional 8 folder if a Lasso Professional server environment is installed on the same machine..
- **Snapshots folder** – This folder is for placing snapshot files created in Lasso Studio. An example snapshot file named Example Snapshot.xml is the only file installed in this folder prior to initialization.

Macromedia Dreamweaver Folder

Lasso Studio installs the following files and folders into the user specific Macromedia Dreamweaver configuration folder. This is the `/Users/<user>/Library/Application Support/Macromedia/Dreamweaver (8 or MX 2004)/Configuration/Shared/LassoStudio` folder on Mac OS X, and the hidden `\Documents and Settings\<user>\Application Data\Macromedia\Dreamweaver (8 or MX 2004)\Configuration\Shared\LassoStudio` folder on Windows.

Warning: The Lasso Studio files installed here should not be removed or modified, or else Lasso Studio will not function properly. If necessary, use the Macromedia Extension Manager to temporarily disable or permanently uninstall Lasso Studio.

- **CodeColoring** – One file is installed in the `Configuration/CodeColoring` folder.
- **CodeHints** – One files is installed in the `Configuration/CodeHints` folder.
- **Commands** – Seven `LassoStudio_DLG` files are installed in the `Configuration/Floaters` folder.
- **Floaters** – Six `LassoStudio_PLT` files are installed in the `Configuration/Floaters` folder.
- **JSExtensions** – A LassoStudio library file is installed in the `Configuration/JSExtensions` folder.
- **Menus folder** – A `LassoStudio_Menu.htm` file is installed in the `Configuration/Menus` folder, and is required for Lasso Studio to operate.
- **Objects folder** – Lasso Data Access, Lasso Programming, and Lasso Form folders are installed in the `Configuration/Objects` folder, and is required for Lasso Studio to operate.

- **Shared folder** – A Lasso Studio folder is installed in the Configuration/Shared folder, and contains the core Lasso Studio interface extension files for Dreamweaver.
- **Startup folder** – A LassoStudio_Startup.htm file is installed in the Configuration/Startup folder.
- **Translators folder** – Two LassoStudio_Translator files are installed in the Configuration/Translators folder.

Lasso Professional Files

For a list of all files installed with the included single-user version of Lasso Professional, see the *Installation Contents* sections for Mac OS X and Windows in the installation chapters of the Lasso Professional 8 Setup Guide.

The included single-user installation of Lasso Professional Developer is identical to the full version except the core LassoService file accepts Lasso Studio 8 serial numbers to go into single-user mode. The single-user version may be converted to a full multi-user version simply by entering a purchased Lasso Professional Server serial number in Lasso Site Administration.

Extended Configuration

This section describes how to use Lasso Studio with an installation of Lasso Professional Server. The following scenarios are supported installations of Lasso Studio.

- **Default Installation** – The default installation of Lasso Studio includes the authoring environment installed into Dreamweaver on the same machine as the included copy of Lasso Professional Developer.

The instructions in this section are not necessary for this installation since it is fully documented earlier in this chapter.

Note: It is also possible to use Lasso Studio with a full version of Lasso Professional Server. This configuration can be accomplished simply by entering a full server serial number into the local copy of Lasso Professional.

- **Remote Installation** – Lasso can connect to a remote installation of Lasso Professional Server or Lasso Professional Developer. Two configuration steps must be performed for this installation: the new version of LassoStudio.LassoApp must be installed on the remote machine and the LassoStudio group must be set up on the remote server (for Lasso Shell support).

Licensing Note: Per the Lasso Studio 8 license agreement, only one copy of the Lasso Studio authoring environment and included single-user version of Lasso Professional Developer may be installed per serial number.

LassoStudio.LassoApp

In order for Lasso Studio to connect to a remote installation of Lasso Professional the LassoStudio.LassoApp must be installed on the remote machine. For best results the newest version of LassoStudio.LassoApp (included with Lasso Studio) should be installed. This will ensure that all new features work as expected.

To install the LassoStudio.LassoApp file:

The LassoStudio.LassoApp file is located in the Lasso Studio 8 application folder inside the Extensions/Lasso Studio Connector subfolder. A build of the LassoApp for Lasso 7 and Lasso 8 are included here. Copy the appropriate LassoApp to the target machine.

- **Lasso 8** – The LassoApp should be placed in the LassoApps folder within the Lasso Professional 8 application folder. The site should be restarted after this file is installed.

The following locations should be checked and any LassoStudio.LassoApp files in them should be removed: the LassoStartup folder in the Lasso Professional 8 application folder and the Lasso folder in the Web root.

- **Lasso 7** – The LassoApp should be placed in the LassoStartup folder within the Lasso Professional 7 application folder and also within the Lasso folder in the Web root. Lasso Service should be restarted after this file is installed.

LassoStudio Group

In order for the Lasso Shell to be able to execute and check Lasso syntax on a given Lasso Professional server it is necessary for either the site administrator username and password to be provided or for a username and password for a user within the LassoStudio group to be provided.

Follow the instructions in the *User Configuration > Creating a LassoStudio group* section in the following chapter for full details.

Note: A Lasso Studio authoring environment may communicate with as many remote multi-user serving configurations as needed, and may be done so concurrently with any local server environment configuration.

Uninstalling Lasso Studio

This section describes uninstalling both the Lasso Studio extensions for Dreamweaver (i.e. authoring environment) and single-user version of Lasso Professional Developer (i.e. server environment) on Mac OS X and Windows.

Mac OS X Uninstallation

On Mac OS X, Lasso Studio can be uninstalled using the Macromedia Extension Manager to uninstall the extensions for Dreamweaver and the provided uninstaller application to remove Lasso Professional Developer.

To uninstall Lasso Studio 8 on Mac OS X:

- Launch the Macromedia Extension Manager and uninstall the Lasso Studio 8 extensions. You will need to restart the Dreamweaver application in order to see the uninstall take effect. If you have both Dreamweaver 8 and Dreamweaver MX 2004 installed then you will need to uninstall Lasso Studio from both.
- Run the uninstaller application provided in the Lasso Studio download package. Mac OS X will prompt for an administrator name and password to remove the installation. Follow the onscreen instructions to uninstall Lasso Professional Developer and the remaining Lasso Studio files.

The uninstaller program will not remove any non-default files created after the time of install. See the *Files and Folder Not Removed* section for more information.

Note: Some of the text within the uninstaller reads as if software is being installed. This is a quirk of the Apple Installer and can be ignored.

Windows Uninstallation

On Windows, Lasso Studio can be uninstalled using the Macromedia Extension Manager to uninstall the extensions for Dreamweaver and the original installer or the Add or Remove Programs control panel, to uninstall Lasso Professional Developer.

To uninstall Lasso Studio 8 on Windows:

- Launch the Macromedia Extension Manager and uninstall the Lasso Studio 8 extensions. You will need to restart the Dreamweaver application in order to see the uninstall take effect. If you have both Dreamweaver 8 and Dreamweaver MX 2004 installed then you will need to uninstall Lasso Studio from both.
- Launch the installer for Lasso Studio. Use the Remove option to uninstall the Lasso Professional Developer and the remaining Lasso Studio files.
- Alternately, open the Add or Remove Programs control panel from the Start Menu. Scroll down to the Lasso Studio for Dreamweaver item. Select Remove to uninstall Lasso Professional Developer and the remaining Lasso Studio files.

The installer program will not remove any non-default files created after the time of install. See the *Files and Folder Not Removed* section for more information.

Files and Folders Not Removed

The Lasso Studio installer program will not remove any non-default files created after the time of install, or any default Lasso files modified since the time of install. These files include the SQLite databases, Lasso log files, Lasso setups, and custom Lasso pages and LassoApps. These files remain in the following locations:

- **Dreamweaver Preference folder** – A LassoStudio Dreamweaver preference folder remains on the system that stores user preferences and data for the Lasso Studio authoring environment. This is the `/Users/<user>/Library/Application Support/Macromedia/Dreamweaver/Configuration/Shared/LassoStudio` folder on Mac OS X, and the hidden `\Documents and Settings\<user>\Application Data\Macromedia\Dreamweaver\Configuration\Shared\LassoStudio` folder on Windows.
- **Lasso Studio 8 folder** – The `/Applications/Lasso Studio 8` or `Program Files\OmniPilot Software\Lasso Studio 8` folder is left on the hard drive, which contains any snapshot files created during use in the contained Snapshots folder.
- **Lasso Professional 8 folder** – The `/Applications/Lasso Professional 8` or `Program Files\OmniPilot Software\Lasso Professional 8` folder is left on the hard drive, which contains folders with any custom modules, JDBC drivers, LassoApps, Lasso libraries, sites, and SQLite databases that have been created.:
- **Web Serving folder** – The Web serving folder (e.g. `/Library/WebServer/Documents` or `C:\inetpub\wwwroot`) retains all Lasso pages created or modi-

fied since the time of installation. Customized Lasso pages, Web pages, and scripts will not be deleted during uninstallation.

These files and folders may be backed up for preservation, or may be deleted to completely remove Lasso Professional and all settings from the system.

Reinstallation Note: Any files remaining on the system from a previous installation of Lasso Studio will not be overwritten by installing a newer version. Therefore, all previous settings will be retained if a newer version of Lasso Studio is installed over the files not removed during the uninstallation of the previous version.

Running Lasso Professional 8

Lasso Professional consists of one main service known as Lasso Service and one or more child processes. Lasso Service is the core executable of Lasso Professional which communicates with a Web server. The sections below describe starting Lasso Service on Mac OS X and Windows.

Mac OS X

Lasso Service is started and stopped using the terminal or scripts located in the `/Applications/Lasso Professional 8/Tools` folder.

- Type the following command into the Mac OS X Terminal application. You will be prompted for your password. This command must be accessed from an administrator account.

```
sudo lasso8ctl start
```

- Alternately, open the Tools folder in the Lasso Professional Server application folder and double click on the `startLassoService.command` file. This will launch a terminal window that prompts you for your password and starts Lasso Service.

For complete instructions on how to start and stop Lasso Service on Mac OS X using these scripts, see the *Configuring on Mac OS X* chapter *Running Lasso Professional 8* section in the Lasso Professional 8 Setup Guide. This section also provides instructions for how to check if Lasso Service is running and how to run Lasso in console mode.

Windows

Lasso Service is started and stopped using the Services control panel.

- From the Windows Task Bar, select *Start > Settings > Control Panel > Administrative Tools > Services*. Right click on Lasso Professional Server 8, then select Start to start Lasso Service, or Stop to stop Lasso Service.

For complete instructions on how to start and stop Lasso Service on Windows using the services menu, see the *Configuring on Windows* chapter *Running Lasso Professional 8* section in the Lasso Professional 8 Setup Guide. This section also provides instructions for how to check if Lasso Service is running and how to run Lasso in console mode.

Lasso Site Administration

Lasso Site Administration is a convenient, Web-based interface for configuring Lasso Professional global settings, configuring data source connections, configuring Lasso Security, monitoring events, and much more. It

is within this interface that all Lasso settings and databases are set up and configured.

Figure 1: Lasso Site Administration



Lasso Site Administration consists of the SiteAdmin.LassoApp file served directly from the LassoApps folder where Lasso Professional 8 is installed. For more information on LassoApps, see the *LassoApps* chapter in the Lasso 8 Language Guide.

Accessing Lasso Site Administration

In a Web browser, visit <http://www.example.com/SiteAdmin.LassoApp>. Replace www.example.com with your domain name or IP address of the Web server running Lasso Professional 8. Use 127.0.0.1 as the IP address if on the same machine. If an error is displayed, make sure Lasso Service is running on the Web server as described previously.

Note: Lasso Site Administration may also be accessed on the same machine by selecting Lasso Site Administration... from the Lasso Studio menu in Dreamweaver.

Using Lasso Site Administration

An overview of the sections and features of Lasso Site Administration is in the *Using Lasso Site Administration* chapter of the Lasso Professional 8 Setup Guide. The points below describe using Lasso Site Administration as it relates to Lasso Studio.

- For instructions on how to set up data sources in Lasso Site Administration for use with Lasso Studio, see the *Data Source Configuration* section of this chapter.
- For instructions on how to set up users other than the Lasso site administrator for use with Lasso Studio, see the *User Configuration* section of this chapter. This is recommended for ISPs and advanced administrators.

- For discussion of setting up Lasso Site Administration for testing and serving custom solutions built using Lasso Studio, see the *Using Lasso Studio* chapter *Testing Solutions* section.

Setup Note: For easy debugging when testing custom solutions created using Lasso Studio, it is strongly recommended that the page error reporting setting be changed to Full. This is set in the *Setup > Global > Settings* section of Lasso Site Administration.

Lasso Database Browser

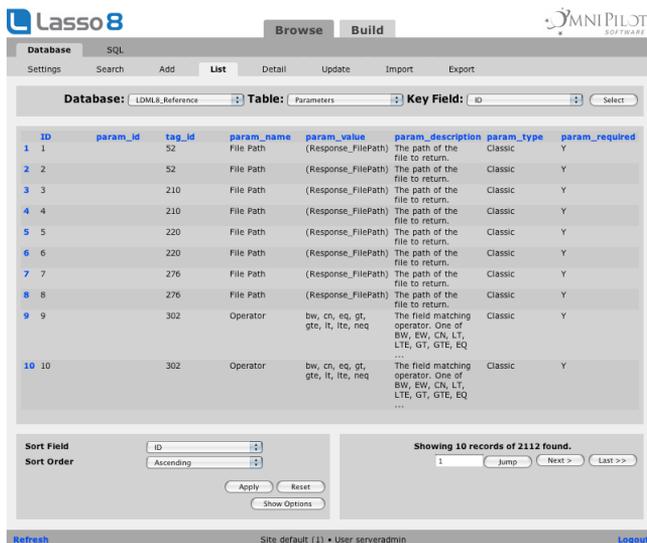
The Lasso Database Browser allows administrators to interact with the records of all databases enabled in Lasso Site Administration. This includes searching, viewing, adding, updating, and deleting records in any database accessible by Lasso. The Lasso Database Browser also allows the administrator to build custom MySQL databases using a Web-based interface for use in Lasso solutions.

The Lasso Database Browser consists of the DatabaseBrowser.LassoApp file served directly from the LassoApps folder where Lasso Professional 8 is installed.

Accessing the Lasso Database Browser:

In a Web browser, visit <http://www.example.com/DatabaseBrowser.LassoApp>. Replace `www.example.com` with your domain name, IP address, or `127.0.0.1` if on a local machine.

Figure 2: Lasso Database Browser



Security Note: The Lasso Database Browser will prompt for the Lasso site administrator username and password (as set up during initialization). Only the site administrator may use the Lasso Database Browser.

Build Section Features

The Build section allows the administrator to create MySQL databases, tables, fields, and indices, and edit the schema of MySQL databases via a Web-based interface. No knowledge of SQL is required to build MySQL databases using the Build section.

Browse Section Features

The Browse section is useful in that it allows the administrator to test databases once they have been enabled, as well as modify records in existing databases. It is always recommend that databases be tested in the Browse section of the Lasso Database Browser before attempting to interact with them via custom Lasso code. The Browse section also allows the administrator to import and export data.

Data Source Configuration

When a Lasso Professional server environment is first installed and initialized, the Lasso Studio authoring environment can only initially communicate with the default SQLite databases installed with Lasso Professional 8, as these are the only databases configured by default after a new installation. This section describes configuring additional data sources in Lasso Professional for use with Lasso Studio.

In order for Lasso products to dynamically communicate with databases, the databases need to be open and running in their respective data sources. A data source must be configured to communicate with a Lasso Professional server environment before it can be used within a Lasso Studio authoring environment. Follow the procedures below for setting up a FileMaker, MySQL, or JDBC-compatible database for use with Lasso Studio.

SQLite

SQLite is the internal data source in Lasso Professional 8, and all databases installed and running in SQLite are instantly available to the Lasso Studio authoring environment. No additional setup is required for Lasso Studio to communicate with SQLite databases in the initialized server environment.

However it is not generally recommended that SQLite be used for solution databases. Instead, a FileMaker, MySQL, or other data source should be configured.

FileMaker

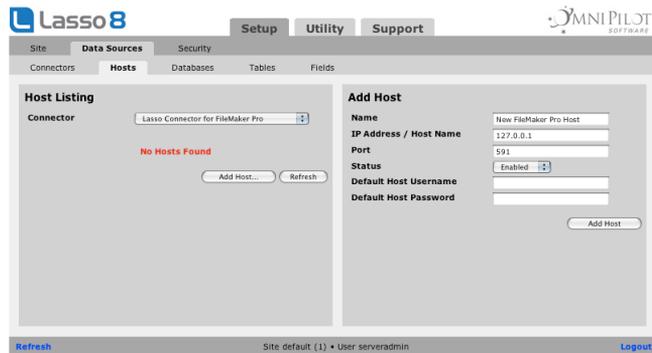
Follow the procedure below to set up a FileMaker database for use with Lasso Studio. This involves configuring the Filemaker application, setting up the FileMaker connection information in Lasso Site Administration, and enabling or disabling FileMaker databases for use with Lasso from within Lasso Site Administration.

To set up a Filemaker host for use with Lasso Studio:

- 1 First, follow the instructions in the *Setting Up Data Sources* chapter in the Lasso Professional 8 Setup Guide for preparing FileMaker to communicate with the Lasso Professional server environment. This involves configuring the Filemaker Pro Web Companion or FileMaker Server Advanced Web components.

- 2 Load Lasso Site Administration in a Web browser. For instructions on how to load Lasso Site Administration, see the installation chapters of this guide.
- 3 Go to the *Setup > Data Sources > Hosts* section in Lasso Site Administration.
- 4 In the Host Listing panel, select Lasso Connector for FileMaker Pro or Lasso Connector for FileMaker SA from the Connector pull-down menu.
- 5 In the Host Listing panel for FileMaker, select the Add Host button.

Figure 3: Host Listing



- 6 In the Add Host panel, enter a host name in the Name field. This is simply the name for the host that will be displayed in Lasso Site Administration.
- 7 Enter the IP address or domain name of the FileMaker host in the IP Address/Host Name field. This is the address of the machine running the FileMaker Pro application.
- 8 Enter the port of the FileMaker host in the Port field. The port number is the port that the FileMaker Pro Web Companion is configured for (defaults to 591) or the port that the FileMaker Server Advanced Web components are listening on (defaults to 80). For more information, see the *Setting Up Data Sources* chapter in the Lasso Professional 8 Setup Guide.
- 9 Select Enabled from the Status pull-down menu to enable the host.
- 10 Enter a default username for the host in the Default Host Username field. Lasso Professional 8 will connect to the data source and all databases therein using this username by default. If the host does not require a username, then leave this field blank.

Note: No username should be entered for a FileMaker Pro host if FileMaker Pro Access Privileges is used as the security method for the FileMaker Pro

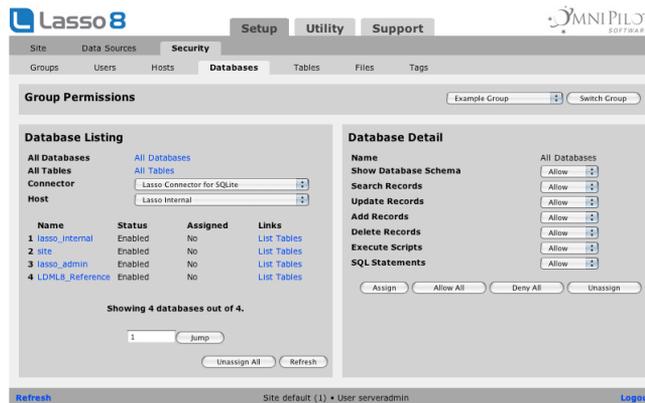
data source. FileMaker Server Advanced hosts require both a username and password.

- 11 Enter a default password for the host in the Default Host Password field. Lasso Professional 8 will connect to the data source and all databases therein using this password by default. If the host does not require a password, then leave this field blank.
- 12 Select Add Host. This will display the added FileMaker Pro host name in the Host Listing panel. The next step is to configure individual databases for use with Lasso.

To set up a Filemaker database for use with Lasso Studio:

- 1 Next to the FileMaker host name in the Host Listing panel, select List Databases. If databases do not appear in the subsequent Databases Listing panel immediately, then try selecting the Refresh button.

Figure 4: Database Listing



- 2 Select a database you want to enable in the Databases Listing panel. This will show the Database Detail panel to the right.
- 3 In the Database Detail panel, select Enabled from the Status pull-down menu to enable the database.
- 4 Select Yes from the Always Connect with Defaults pull-down menu to ensure that Lasso will connect to the database with the username and password specified in the Hosts page in Lasso Site Administration.
- 5 For FileMaker Pro hosts only, select No from the Do ISO/Mac Conversion pull-down menu unless Lasso Service and FileMaker Pro are on different operating systems.
- 6 Select Update. This enables the database and all layouts within for use with Lasso.

The selected database is now enabled for use with Lasso Professional 8 and Lasso Studio. Repeat the procedure above for all databases you intend to use with Lasso. The Enable All New button in the Database Listing panel may also be used to quickly enable all Filemaker databases in the specified host. For more information on using the *Setup > Data Sources* section of Lasso Site Administration.

MySQL

Follow the procedure below to set up an external MySQL database for use with Lasso Studio. This involves configuring the external version of MySQL to talk to Lasso Professional, setting up the MySQL connection information in Lasso Site Administration, and enabling or disabling MySQL databases for use with Lasso from within Lasso Site Administration.

To set up an external MySQL host for use with Lasso Studio:

- 1 First, follow the instructions in the *Setting Up Data Sources* chapter in the Lasso Professional 8 Setup Guide for preparing MySQL to communicate with the Lasso Professional server environment.
- 2 Load Lasso Site Administration in a Web browser. For instructions on how to load Lasso Site Administration, see the installation chapters of this guide.
- 3 Go to the *Setup > Data Sources > Hosts* section in Lasso Site Administration.
- 4 In the Host Listing panel, select Lasso Connector for MySQL from the Connector pull-down menu.
- 5 In the Host Listing panel for MySQL, select the Add Host button.

Figure 5: Host Listing

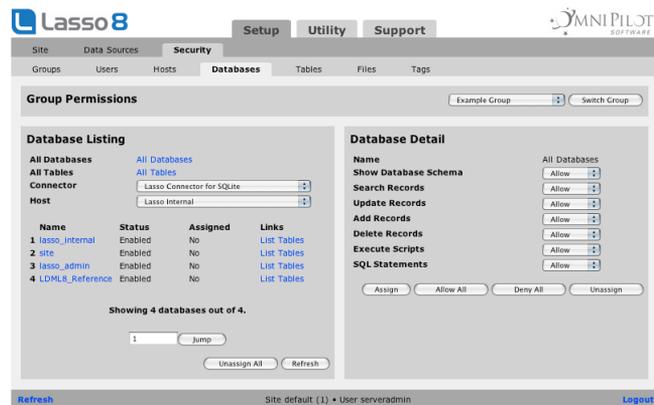


- 6 In the Add Host panel, enter a host name in the Name field. This is simply the name for the host that will be displayed in Lasso Site Administration.
- 7 Enter the IP address or domain name of the MySQL host in the IP Address/Host Name field. This is the address of the machine running the external version of MySQL you are connecting.
- 8 Enter the port of the MySQL host in the Port field (e.g. 3306). For more information, see the *Setting Up Data Sources* chapter in the Lasso Professional 8 Setup Guide.
- 9 Select Enabled from the Status pull-down menu to enable the host.
- 10 Enter a default username for the host in the Default Host Username field. Lasso Professional 8 will connect to the data source and all databases therein using this username by default. If the host does not require a username, then leave this field blank.
- 11 Enter a default password for the host in the Default Host Password field. Lasso Professional 8 will connect to the data source and all databases therein using this password by default. If the host does not require a password, then leave this field blank.
- 12 Select Add Host. This will display the added MySQL host name in the Host Listing panel. The next step is to configure individual databases for use with Lasso.

To set up an external MySQL database for use with Lasso Studio:

- 1 Next to the MySQL host name in the Host Listing panel, select List Databases. If databases do not appear in the subsequent Databases Listing panel immediately, then try selecting the Refresh button.

Figure 6: Database Listing



- 2 Select a database you want to enable in the Databases Listing panel. This will show the Database Detail panel to the right.
- 3 In the Database Detail panel, select Enabled from the Status pull-down menu to enable the database.
- 4 Select Yes from the Always Connect with Defaults pull-down menu to ensure that Lasso will connect to the database with the username and password specified in the Hosts page in Lasso Site Administration.
- 5 Leave the SQL Stop Words field blank unless you are familiar with and need SQL stop word operation. This is described in the *Setup* chapter of the Lasso Professional 8 Setup Guide.
- 6 Select Update. This enables the database and all tables within for use with Lasso.

The selected database is now enabled for use with Lasso Professional 8 and Lasso Studio. Repeat the procedure above for all databases you intend to use with Lasso. The Enable All New button in the Database Listing panel may also be used to quickly enable all new databases in the specified host. For more information on using the *Setup > Data Sources* section of Lasso Site Administration, see the *Setup* chapter in the Lasso Professional 8 Setup Guide.

JDBC Data Sources

Follow the procedure below to set up a JDBC data source for use with Lasso Studio. This involves configuring the JDBC data source and JDBC driver for use with Lasso Professional 8, setting up the JDBC data source connection information in Lasso Site Administration, and enabling or disabling databases for use with Lasso from within Lasso Site Administration. JDBC data sources include Microsoft SQL Server, Frontbase, Sybase, Oracle, PostgreSQL, and many more.

To set up a JDBC data source for use with Lasso Studio:

Due to the large variation in setup procedures for the various JDBC data sources, a single procedure cannot be listed here. See the OmniPilot Support Central Web site for instructions on setting up specific JDBC data sources.

<http://support.omnipilot.com>

User Configuration

In default installations of Lasso Studio, the Lasso site administrator password is used for the initial Lasso Studio authoring environment initialization. This makes it easy for the developer using Lasso Studio to instantly have access to all databases in the server environment, and is recommended for most users.

This section describes creating and using dedicated Lasso groups and users in the server environment for use with Lasso Studio as opposed to the Lasso site administrator account. Creating specific groups and users serves to provide Lasso Studio developers with dedicated access only to certain databases while restricting others. This is recommended for administrators and ISPs who do not wish to grant global control to individual Lasso Studio developers.

A LassoStudio group is required for any users other than the site administrator who will access the Lasso Shell palette. Only the site administrator or users who belong to a LassoStudio group will be able to check or execute code on a Lasso Professional machine through the Lasso Shell.

The procedures described in this section can be applicable to either the single-user version of Lasso Professional Developer included with Lasso Studio, or to full versions of Lasso Professional Server that are intended to communicate with a Lasso Studio authoring environment.

Important: It is not necessary to perform the procedures in this section if both the authoring environment and server environment in Lasso Studio are controlled exclusively by the same person, or developer restrictions are not desired or necessary.

Creating a LassoStudio group

In Lasso Site Administration, a group defines a set of database permissions for one or more users. For more information on groups, see the *Setting Up Security* chapter *Groups and Users* section in the Lasso Professional 8 Setup Guide.

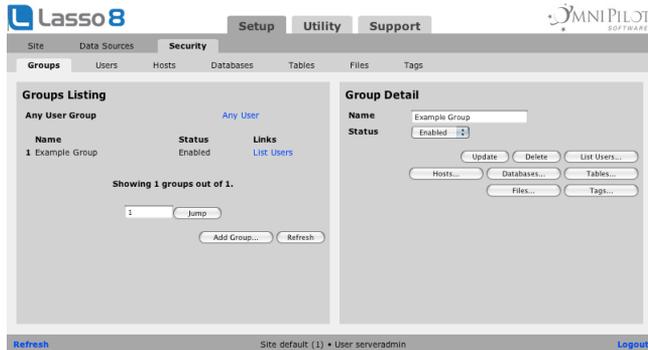
Follow the procedures below to create a dedicated Lasso group for Lasso Studio using the Add Group page in Lasso Site Administration. The Lasso group we create will have permissions to perform actions only on the databases to which permissions are explicitly allowed.

To create a LassoStudio group:

- 1 In Lasso Site Administration, go to *Setup > Security > Groups*.

- 2 In the Groups Listing panel, select the Add Group button. This shows the Add Group panel to the right.
- 3 Enter a name for the group in the Name field (e.g. LassoStudio).
- 4 Select Enabled from the Status pull-down menu.
- 5 Select Add Group.

Figure 7: Group Listing



The group should now be shown in the Groups Listing panel and in the Group Detail panel to the right. Now, we need to assign database and table permissions for the group. No permissions are allowed unless they are explicitly assigned.

To assign database permissions to the LassoStudio group:

- 1 In the Group Detail panel for the group, select the Databases... button.
- 2 In the Databases Listing panel, select a database to be enabled under the appropriate data source connector and host. This should show the Database Detail panel for the database to the right.
- 3 Select Allow to grant permissions for specific actions in the Show Database Schema, Search Records, Update Records, Add Records, Delete Records, Execute SQL and Execute Scripts pull-down menus. These permissions are individually described in the *Setting Up Security* chapter in the Lasso Professional 8 Setup Guide.

Note: Only Show Database Schema and Search Records permissions are required for operation of the data access features of the Lasso Studio authoring environment are required. Other permissions may be added if the LassoStudio group will be used for solution testing, which is described later.

- 4 Select Assign.

Figure 8: Database Permissions



Next, we need to assign appropriate permissions for the tables in the database. Lasso Professional 8 features table-level security, and access can be granted explicitly to individual tables in a database while restricting others.

All Tables Note: To instantly grant table-level access for all tables in all allowed databases, see the documentation for the All Tables setting in the *Setup* chapter of the Lasso Professional 8 Setup Guide. Otherwise, follow the procedure below for each table for which you want to allow access.

To assign table permissions to the LassoStudio group:

- 1 In the Database Listing panel, select List Tables next to the database that was configured previously.
- 2 In the Table Listing panel, select a table for which to set permissions. This will show the Table Detail panel for that table to the right.
- 3 Select Allow to grant permissions for specific actions in the Show Database Schema, Search Records, Update Records, Add Records, Delete Records, Execute SQL and Execute Scripts pull-down menus. These permissions are individually described in the *Setup* chapter in the Lasso Professional 8 Setup Guide.

Note: Only Show Database Schema and Search Records permissions are required for the data access features of the Lasso Studio authoring environment are required. Other permissions may be added if the LassoStudio group will be used for solution testing, which is described later.

- 4 Select Assign.

Figure 9: Table Permissions



Members of the group may now access the allowed databases and tables from within the Lasso Studio authoring environment. Next, we need to create users within the group.

Creating Lasso Studio Users

In Lasso Site Administration, one or more users may be added to a single pre-defined group. These users will have the database and table permissions of the group to which they are assigned. The Users section in Lasso Site Administration allows new users to be created.

Users which are added to the LassoStudio group will also have the ability to check and execute code through the Lasso Shell palette. Since these users can execute arbitrary code through Lasso Professional it is important to only provide usernames to trusted developers.

To add a user to the LassoStudio group:

- 1 In Lasso Site Administration, go to *Setup > Security > Users*.
- 2 In the User Listing panel, select the Add User button. This displays the Add User panel to the right.
- 3 Enter a username in the Username field. This will be used within the Lasso Studio authoring environment to connect to the server environment.
- 4 Enter a password in the Password field. This will be used within the Lasso Studio authoring environment to connect to the server environment.
- 5 Select the name of the group in the Groups field to which the user is to be assigned.
- 6 Select Add User.

Figure 10: User Listing



This will create the user specified in the selected group. The Lasso Studio developer to which this username is assigned will be required to know this username and password to gain access to the databases and tables allowed.

Once a group and user account is created with privileges for specific databases and tables, it may then be used by a Lasso Studio developer for connections to that server environment. See the *Using Database Snapshots* chapter for instructions on how to initiate a new connection.

4

Chapter 4

Using Lasso Studio

This chapter provides an overview of using the Lasso Studio authoring environment to build Lasso solutions using Dreamweaver.

- *Overview* provides an overview of this chapter and the authoring environment of Lasso Studio.
- *Lasso Studio Menu* describes the Lasso Studio menu in the authoring environment interface.
- *Panels* describes the various panels available in the Lasso Studio authoring environment interface.
- *Screens and Editors* describes the various screens and editors available in the Lasso Studio authoring environment interface.
- *Testing Solutions* describes testing and implementing solutions built using Lasso Studio in a Lasso Professional server environment.
- *Learning Resources* provides resources for learning Lasso.

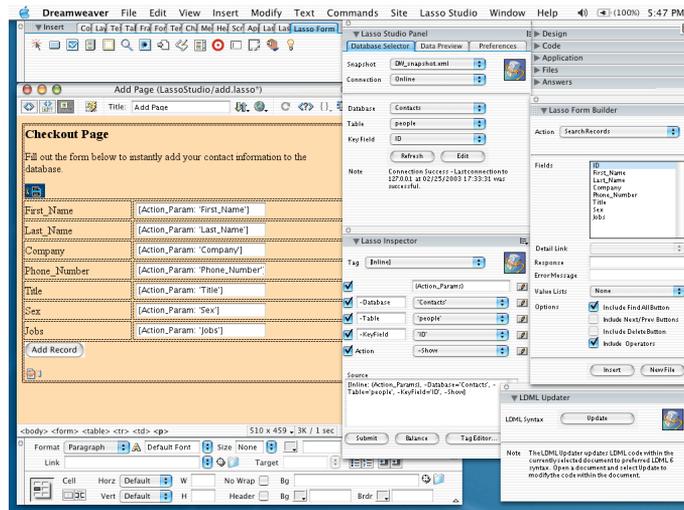
Overview

Lasso Studio is a powerful development tool for Lasso Professional 8, and contains many features that simplify the process of programming data-driven Web sites. The Lasso Studio authoring environment is integrated within the Dreamweaver interface, and is instantly launched whenever the Dreamweaver application is started.

This chapter provides an overview of all interface elements in the authoring environment, a tutorial for quickly building a data-driven Web site, and describes testing and troubleshooting Lasso solutions once they

are created. It also provides a list of resources for learning how to build Lasso solutions.

Figure 1: Authoring Environment



Interface Elements

The interface elements of the Lasso Studio authoring environment are described in the following *Lasso Studio Menu*, *Panels*, and *Screens and Editors* sections of this chapter. These sections provide an overview of all Dreamweaver interface elements that make up the Lasso Studio authoring environment, and provide references to the chapters in which the features are described.

Testing Your Work

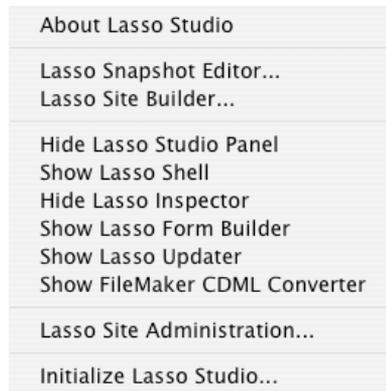
The *Testing Solutions* section provides guidelines for testing and implementing solutions built using Lasso Studio in a Lasso Professional server environment. This section describes the necessary steps involved, such as setting up user database permissions in Lasso Site Administration and uploading files. This section also offers troubleshooting tips.

Lasso Studio Menu

The Lasso Studio installer adds a Lasso Studio menu to the menu bar in Dreamweaver. All of the screens and panels in Lasso Studio can be accessed from this menu.

Important: If you don't see a Lasso Studio menu within Dreamweaver then you must manually open the *Lasso Studio 8.1 for Dreamweaver.mxp* file in order to complete the installation of Lasso Studio. If you have both Dreamweaver 8 and Dreamweaver MX 2004 installed make sure that Lasso Studio was installed into both using the Macromedia Extension Manager.

Figure 2: Lasso Studio Menu



Each command is discussed in detail below.

- **About Lasso Studio** – This command shows the About screen for Lasso Studio.
- **Lasso Snapshot Editor...** – Opens the Lasso Snapshot Editor screen, which allows you to create and load snapshot files of your database schema. The Snapshot Editor is described in the *Using Database Snapshots* chapter
- **Lasso Site Builder...** – Shows or hides the Lasso Site Builder screen, which allows you to create a set of Lasso-enabled Web pages that perform various database actions. The Site Builder is described in the *Using Site Builder* chapter.
- **Show/Hide Lasso Studio Panel** – Shows or hides the Lasso Studio panel, which allows you to use the Database Selector, set Data Preview settings, and set general Lasso Studio preferences. The Lasso Studio panel is described primarily in the *Using Database Snapshots* chapter.

- **Show/Hide Lasso Shell** – Shows or hides the Lasso Shell panel, which allows you to send commands to Lasso Service and see the results. The Lasso Shell panel is described primarily in the *Using Lasso Shell* chapter.
- **Show/Hide Lasso Inspector** – Shows or hides the Lasso Inspector panel, which allows you to inspect and update Lasso Objects and tags. The Lasso Inspector is described in the *Using Lasso Inspectors* chapter.
- **Show/Hide Lasso Form Builder** – Shows or hides the Lasso Form Builder panel, which allows you to insert a Lasso-enabled form into the current document. The Form Builder is described in the *Using Form Builder* chapter.
- **Show/Hide Lasso Updater** – Shows or hides the Lasso Updater panel, which allows you to update Lasso code from older versions of Lasso to Lasso 8 syntax. The Lasso Updater is described in the *Using Lasso Updater and FileMaker CDML Converter* chapter.
- **Show/Hide FileMaker CDML Converter** – Shows or hides the FileMaker CDML Converter panel, which allows you to update CDML code from FileMaker Pro custom Web solutions to Lasso 8 syntax. The CDML Converter is described in the *Using Lasso Updater and FileMaker CDML Converter* chapter.
- **Lasso Site Administration...** – Loads Lasso Site Administration in a Web browser for the local Lasso Professional server environment installed on the same machine. This requires that a default Web browser be configured in Dreamweaver. For more information on Lasso Site Administration, see the *Configuring Lasso Professional 8* chapter.
- **Initialize Lasso Studio...** – Shows the Lasso Studio initialization screen for product initialization, as described in the installation chapters of this guide.

Panels

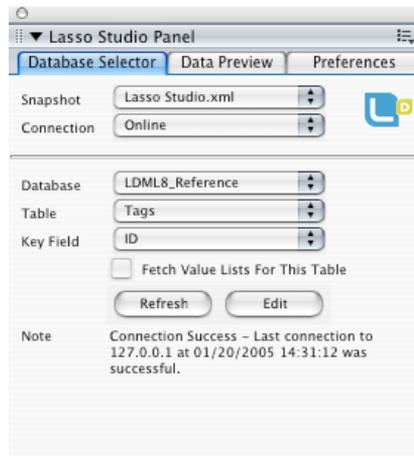
Lasso Studio adds several panels to Dreamweaver, which are described in this section. These panels can be docked with other floating panels in Dreamweaver, and each panel will remember its position between restarts of the authoring environment.

Lasso Studio Panel

The Lasso Studio panel is the central interface element for Lasso Studio, and is accessed by selecting Show Lasso Studio Panel from the Lasso Studio menu. The Lasso Studio panel provides three separate tabs, which are described below.

- **Database Selector** – Allows you to choose a database, table and key field from a snapshot file to work with. The current database and table selected in this interface is the working database and table in Lasso Studio for all data access features such as Form Builder or Site Builder. This interface also allows database snapshot files to be created, edited, and saved via the Snapshot Editor, which is described in the *Screens and Editors* section. The Database Selector is documented in the *Using Database Snapshots* chapter.
- **Data Preview** – Allows the Live Data Preview options to be set. This includes enabling and disabling Live Data Preview, and editing and navigating sample record sets. This interface also allows sample record data to be created and stored in snapshot files via the Snapshot Editor, which is described in the *Screens and Editors* section. The Data Preview feature is documented in the *Using Database Snapshots* chapter.
- **Preferences** – Allows general preferences for Lasso Studio such as Lasso display options to be set. This is described primarily in the *Using Objects and Tags* chapter.

Figure 3: Lasso Studio Panel

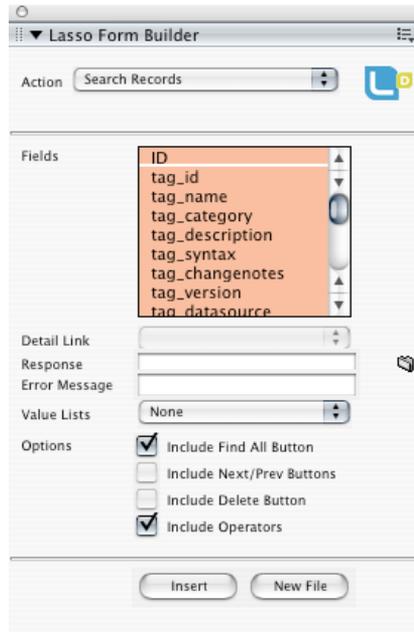


Snapshot and Connection pull-down menus are shown at the top of each section. The Snapshot pull-down menu allows the snapshot currently in use to be changed. The Connection pull-down menu allows the connection state of Lasso Studio to be specified. If the connection is Online, then the server will be checked for any database schema changes each time a database or table is visited.

Lasso Form Builder Panel

The Lasso Form Builder panel provides access to the Form Builder, which is used to insert a Lasso form in a single Web page. Forms that search records, add records, and more can be instantly created using the Form Builder. The Form Builder panel is accessed by selecting Show Form Builder from the Lasso Studio menu, and is described in the *Using Form Builder* chapter.

Figure 4: Lasso Form Builder Panel



Lasso Inspector Panel

The Lasso Inspector panel allows you to view and modify the common properties of any Lasso tag, hidden input, or form element. The Lasso Inspector panel is accessed by selecting Show Lasso Inspector from the Lasso Studio menu, and then selecting a Lasso Object that has been added to a Web page. This feature is described in the *Using Lasso Inspectors* chapter.

Figure 5: Lasso Inspector Panel



Lasso Objects

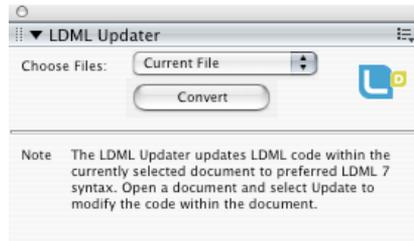
Lasso Studio adds three sections of objects to the Insert panel in Dreamweaver, which is accessed by selecting Insert from the Window menu in Dreamweaver. These objects allow you to drag-and-drop Lasso tags into your Web pages. The Lasso Object sections include Lasso Data Access, Lasso Programming, and Lasso Form, and are described in the *Using Objects and Tags* chapter.

Figure 6: Lasso Objects



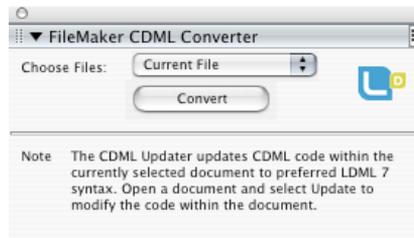
Lasso Updater

The Lasso Updater panel allows you to update Lasso code from older versions of Lasso to Lasso 8 syntax. The Lasso Updater is accessed by selecting Show Lasso Updater from the Lasso Studio menu, and is described in the *Using Lasso Updater and FileMaker CDML Converter* chapter.

Figure 7: Lasso Updater

FileMaker CDML Converter

The FileMaker CDML Converter panel allows you to update CDML code from FileMaker Pro custom Web publishing solutions to Lasso 8 syntax. The FileMaker CDML Converter is accessed by selecting Show FileMaker CDML Converter from the Lasso Studio menu, and is described in the *Using Lasso Updater and FileMaker CDML Converter* chapter.

Figure 8: FileMaker CDML Converter

Lasso Shell

The Lasso Shell panel allows you to send Lasso code to Lasso Service and see the results. This allows you to try out code snippets and see if there are any errors in them before you add them to your Web site. The Lasso Shell is accessed by selecting Show Lasso Shell from the Lasso Studio menu, and is described in the *Using Lasso Shell* chapter.

Figure 9: Lasso Shell



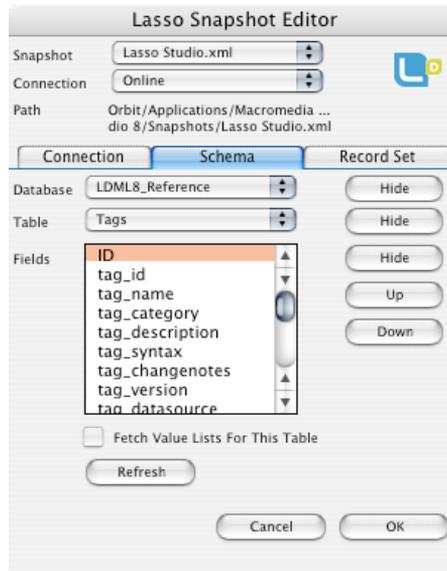
Screens and Editors

Lasso Studio includes various screens in Dreamweaver for using the features of the Lasso Studio authoring environment. Screens in Lasso Studio are for performing larger operations that cannot be handled within single panels. These screens and editors are described in this section.

Snapshot Editor

The Snapshot Editor screen allows existing database snapshot files to be updated and edited. A snapshot file is simply an XML text document created by Lasso Studio and Lasso Professional that contains information about databases being connected to, such as database, table, and field names. These snapshot files in turn give Lasso Studio information about eligible databases when writing Lasso code for data access. The Snapshot Editor is accessed by selecting Lasso Snapshot Editor from the Lasso Studio menu, and is described in the *Using Database Snapshots* chapter.

Figure 10: Snapshot Editor Screen



Snapshots can be updated by reconciling the information stored in the snapshot with the live database, or they can be manually edited just from within Dreamweaver. The Snapshot Editor screen contains three tabs for editing the properties for snapshot files.

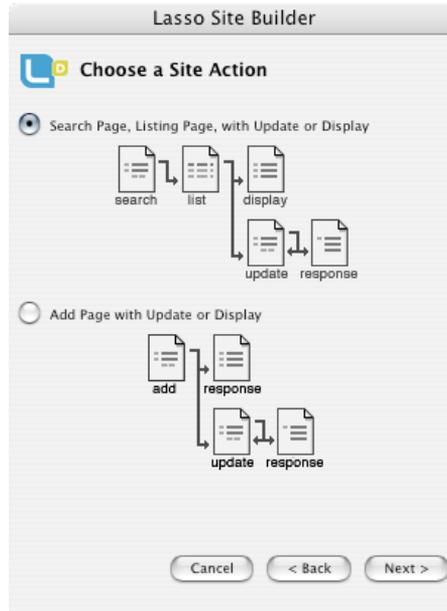
- **Connection** – Shows the Lasso Professional server connection information for the snapshot file. The database information in the snapshot comes from this server.
- **Schema** – Shows the database, tables, fields, and value lists currently stored in the snapshot file, and controls how they will be displayed throughout the Lasso Studio authoring environment interface.
- **Record Set** – Shows the record settings in the snapshot file for the Data Preview feature. This allows the sample record data stored in the snapshot to be previewed and edited. The sample record set created here is shown in the Design view in Dreamweaver whenever Data Preview is used for a particular database.

Site Builder

The Lasso Site Builder screen allows you to create an entire Lasso-based Web site using a simple dialog-driven interface. It creates various forms and

pages based on the database, table and key field selected in the Database Selector of the Lasso Studio panel.

Figure 11: Site Builder Screen



The Site Builder differs from the Form Builder in that it creates multiple pages, whereas the Form Builder only creates individual pages at a time. The Site Builder is accessed by selecting Lasso Site Builder from the Lasso Studio menu, and is described in the *Using Site Builder* chapter.

Tag Editor

The Lasso Tag Editor screen is used to edit the attributes of Lasso tags which are too complex to be fully editable in the Lasso Inspector panel. The Tag Editor is used to edit inline tags, edit complex date, file, and string operations, and more. The Tag Editor is accessed from the Lasso Inspector panel, and is described in the *Using Lasso Tag Editors* chapter.

Figure 12: Tag Editor Screen



Testing Solutions

This section describes testing and troubleshooting solutions that are built using Lasso Studio. Once data-driven pages have been built, they may be loaded in the Lasso Professional server environment and tested with a Web browser. Part of doing this includes setting up Lasso Professional database security in Lasso Site Administration for end-users, which is described below.

Deployment Note: The procedures described in this section also apply when implementing your finished Lasso solution on a production server with a multi-user version of Lasso Professional. Ideally, your single-user testing version of Lasso Professional should have the same security configuration as your production server environment.

Lasso Security

Lasso Professional includes a robust database security system for protecting databases, files, and more. This security system prevents unauthorized users from accessing data in certain databases or tables, and allows authentication systems to be set up.

So far we have been using the Lasso site administrator username and password for accessing database schema information, which is the only user

in Lasso Professional that has permission to access data after a new installation. Many users who will use the Lasso solutions being developed will not have this level of authentication, therefore the site administrator must set up groups and users in Lasso Site Administration that can access the specific data needed.

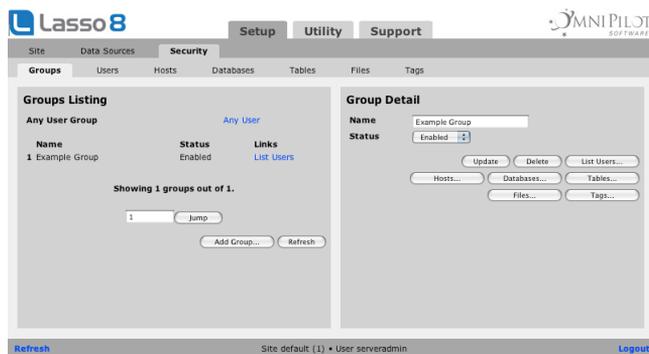
In Lasso Professional, database and table access can be granted to users without them having to submit a username or password. This is done by assigning permissions to the AnyUser group in Lasso Site Administration.

The AnyUser Group

The AnyUser group in Lasso Site Administration represents all end-users of Lasso solutions who do not have to submit a Lasso username or password to access data. The AnyUser group has no permissions assigned to it in Lasso Site Administration by default.

Once permissions for a database or table are granted to the AnyUser group, that database and table are then available to Lasso Studio developers connecting to the server environment without a username or password. No user configuration is required for the AnyUser group. For more information on the AnyUser group, see the *Setting Up Security* chapter *Groups and Users* section in the Lasso Professional 8 Setup Guide.

Figure 13: Groups Page



To assign database permissions to the AnyUser group:

- 1 In Lasso Site Administration, visit *Setup > Security > Groups*.
- 2 In the Groups Listing panel, select the AnyUser group. The Group Detail panel for the AnyUser group should appear to the right.
- 3 In the Groups Detail panel, select the Databases... button. This will take you to the Databases page with the AnyUser group selected.

- 4 In the Databases Listing panel, select the database you wish to allow under the appropriate data source connector and host. This should show the Database Detail panel for the database to the right.
- 5 Select Allow to grant permissions for specific actions in the Show Database Schema, Search Records, Update Records, Add Records, Delete Records, Execute SQL and Execute Scripts pull-down menus. These permissions are individually described in the *Setting Up Security* chapter in the Lasso Professional 8 Setup Guide.
- 6 Select Assign.

Next, we need to assign appropriate permissions for the tables in the database. Lasso Professional features table-level security, and access can be granted explicitly to individual tables in a databases while restricting others.

All Tables Note: To instantly grant table-level access for all tables in all allowed databases, see the documentation for the **All Tables** setting in the *Setting Up Security* chapter of the Lasso Professional 8 Setup Guide. Otherwise, follow the procedure below for each table for which you want to allow access.

To assign table permissions to the AnyUser group:

- 1 In the Database Listing panel, select List Tables next to the database that was configured previously.
- 2 In the Table Listing panel, select a table for which to set permissions. This will show the Table Detail panel for that table to the right.
- 3 Select Allow to grant permissions for specific actions in the Show Database Schema, Search Records, Update Records, Add Records, Delete Records, Execute SQL and Execute Scripts pull-down menus. These permissions are individually described in the *Setting Up Security* chapter in the Lasso Professional 8 Setup Guide.
- 4 Select Assign.

Members of the AnyUser group may now access the allowed databases and tables using an authored Lasso solution. No additional user configuration is required for the AnyUser group.

Custom Groups and Users

If needed, custom groups and users can be created in Lasso Site Administration for varying levels of user authentication in Lasso solutions. This is described in the *Setting Up Security* chapter in the Lasso Professional 8 Setup Guide.

Loading and Testing Solutions

This section describes loading and testing your developed Lasso solution in a Lasso Professional server environment. This section assumes that your server environment, database, and Lasso Security have all been set up as described earlier in this chapter.

Error Reporting Note: For easy debugging when testing custom solutions created using Lasso Studio, it is strongly recommended that the page error reporting setting be changed to **Full**. This is set in the **Setup > Global > Settings** section of Lasso Site Administration.

To test your solution:

- 1 Check to make sure that the Web server using Lasso Professional 8 is running, as described in the *Configuring Lasso Professional 8* chapter in this guide.
- 2 If not already, make sure your solution files and folders have been copied to a location in the Web serving folder of your server environment. This is typically `/Library/WebServer/Documents` for Apache on Mac OS X, or `C:\inetpub\wwwroot` for IIS on Windows.
- 3 Load your solution start page in a Web browser. The following example assumes you are using the `search.lasso` page in the Tutorial folder created in the *Quick Start Tutorial* section of this chapter. Replace `www.example.com` with your domain name, IP address, or `127.0.0.1` if on the local machine.
`http://www.example.com/Tutorial/search.lasso`
- 4 Navigate through your solution to make sure it is working. All data-driven pages should dynamically display information from the database you are using, and you should be able to add, update, or delete records if applicable to your solution.

Troubleshooting

This section provides a list of some of the most basic problems that can be encountered when troubleshooting a Lasso solution. These are mostly configuration issues, and many of the points listed here are covered previously in this guide. If you experience any problems while testing a Lasso solution, then check the following.

Note: This section is provided as a basic guide for solving common configuration issues and does not describe how to troubleshoot specific Lasso solutions or Lasso code. Troubleshooting code and solution functionality is always the responsibility of the developer.

General

- Make sure Lasso Service is running in the Lasso Professional server environment. Refer to the installation chapters of the Lasso Professional 8 Setup Guide for instructions.
- Make sure your Web server is running, and that Lasso Professional 8 has been correctly installed as described in the *Installing Lasso Studio* chapter. Verifying that Lasso Professional 8 has been correctly installed is described in the installation chapters of the Lasso Professional 8 Setup Guide.
- Make sure the pages are being served by your Web server and not opened as files. In other words, make sure `http://` is in the URL of your Web browser, as shown below.
`http://127.0.0.1/Tutorial/search.lasso`
- Make sure the pages you are loading in a Web browser are in the correct folder in the Web serving folder from which they are being called.
- Make sure the pages you are loading in a Web browser have a file extension that is configured in your Web server to be processed by Lasso Professional. Installing Lasso Professional 8 sets up the `.lasso` extension by default, which is the recommended file extension to be used by Lasso Web pages.
- Make sure that your database is configured in Lasso Site Administration as described in the *Configuring Lasso Professional 8* chapter *Data Source Configuration* section.
- Make sure that security for your database has been set up in Lasso Site Administration as described in the *Testing Solutions > Lasso Security* section of this chapter.

Learning Resources

There are many resources available to help you quickly learn how to build data-driven Web sites powered by Lasso.

To learn more about building sites powered by Lasso Professional 8:

- **Lasso 8 Language Guide** – The documentation of Lasso (Lasso Dynamic Markup Language), the language used to access data sources, specify programming logic, and much more.
- **Lasso Reference** – Provides detailed documentation of each tag in Lasso 7. This is the definitive reference to the language of Lasso 8. This

reference is provided as a LassoApp installed with Lasso Professional 8, and is also available as an online resource from the OmniPilot Web site.

<http://dml.omnipilot.com>

- **Lasso Professional 8 Setup Guide** – This book describes the features of Lasso Site Administration, and describes using Lasso Security.
- **LassoTraining.com** – A Web resource that lists Lasso training resources to help build data-driven Web sites powered by Lasso.
<http://www.lassotraining.com>
- **Lasso Talk Mailing List** – A community of Lasso developers who help to answer each other's questions about using Lasso products. If you can't find an answer to your question then posting to the list will often result in an answer in a short period of time. Information about subscribing to the list and searchable list archives can be found at the following address.
<http://listsearch.omnipilot.com/lassotalk.lasso>
- **Lasso Studio Talk Mailing List** – An email discussion forum similar to the Lasso Talk mailing list, but dedicated to Lasso Studio users.
<http://listsearch.omnipilot.com/lassostudiotalk.lasso>

Studio information about eligible databases when writing Lasso code for data access.

How do I use snapshot files?

Snapshot files are used via the Database Selector section of the Lasso Studio panel, which allows you to select which of your available databases and tables in the snapshot you want to use for authoring pages in Lasso Studio. This is described in the *Using Snapshots* section of this chapter.

Can snapshot files be easily updated if my database configuration changes?

Yes. The Database Selector features a Refresh button which queries the appropriate server environment and automatically updates the snapshot file with new information. This is described in the *Using Snapshots* section of this chapter.

Can I manually edit snapshot files inside the Lasso Studio authoring environment?

Yes. The Snapshot Editor in Lasso Studio allows you to edit snapshot files manually without connecting to your server environment. This is described in the *Editing Snapshots* section of this chapter.

Can snapshot files contain information about more than one Lasso Professional server environment?

No. Snapshot files reflect the database information for one server environment only. It is recommended that a new snapshot file be created for each individual server environment.

Can I trade snapshot files with other Lasso Studio developers?

Yes. Snapshot files are stored in the Snapshots folder inside the Lasso Studio 8 folder by default. These files can be copied and distributed to other Lasso Studio developers if desired. More on this is described in the *Using Snapshots* section of this chapter.

Do snapshot files contain value lists for my fields?

By default, Lasso Studio does not download value lists into the current snapshot. This helps keep the size of the snapshot file more manageable and increases the performance of Lasso Studio. However, the Fetch Value Lists for this Table option in the Snapshot Editor or Lasso Studio Panel can be used to instruct Lasso Studio to fetch value lists for any particular table or FileMaker layout.

Creating Snapshots

This section describes connecting the Lasso Studio authoring environment to a Lasso Professional server environment in order to create a database snapshot file. A snapshot file can be created in two ways:

- **Initialization** – Lasso Studio creates an initial snapshot during the authoring environment initialization of Lasso Studio. This file is stored in the Snapshots folder inside your Lasso Studio 8 application folder, and contains the database schema information for all data sources currently registered in Lasso Site Administration. For more information, see the installation chapters of this guide.
- **Snapshot Editor** – In addition to viewing and editing snapshot files, the Snapshot Editor can also be used to create new snapshot files. Snapshot files created using the Snapshot Editor are stored in the Snapshots folder inside your Lasso Studio 8 application folder. The Snapshot Editor can be launched using the Lasso Studio menu, and using it to create new snapshot files is described below.

Creating a new snapshot file after initialization is usually not necessary if you are only using the one server environment that was configured during the initialization of the Lasso Studio. For information on updating your existing snapshot with new information from your server environment, see the *Using Snapshots* section.

Creating New Snapshot Files

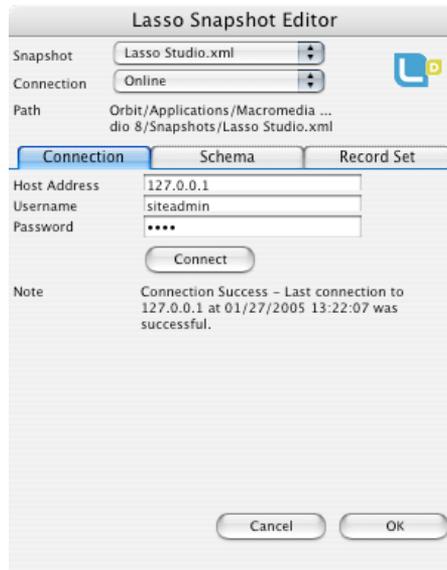
In order to take a snapshot of your database configuration schema using the Snapshot Editor, you need several pieces of information: the static IP address or domain name of your Web serving machine, and a Lasso Site Administration username and password with which to connect. See the installation chapters of this guide for more information about installing and configuring Lasso Studio to communicate with your server environment.

The following conditions should be met before you run the Snapshot Editor to take a snapshot.

- Your Web server should be running.
- The Lasso Professional server environment should be installed with data sources properly configured, as described in the *Configuring Lasso Professional 8* chapter.
- The databases you want to add to your snapshot file should be enabled in Lasso Site Administration, as described in the *Configuring Lasso Professional 8* chapter.

Once these requirements are met, use the procedure below to create a new snapshot file.

Figure 1: Snapshot Editor



To create a new snapshot:

- 1 From the Lasso Studio menu, select Lasso Snapshot Editor.
- 2 In the Snapshot pull-down menu, select New Snapshot.... This will prompt for a file name and location for a new snapshot file.
- 3 In the Save As field, enter a file name for the new snapshot (e.g. New_Snapshot.xml).
- 4 In the Location field, select the folder to which the snapshot will be saved. This is the Snapshots folder in the Lasso Studio 8 application folder by default, which is the recommended location for storing all snapshot files.
- 5 Select the Save button to save the new snapshot file.
- 6 In the Host Address field, enter the IP address or host name of the machine on which the Lasso Professional server environment is installed. Use 127.0.0.1 if the server environment is on the machine you are currently using.
- 7 In the Username field, enter the Lasso Professional 8 username you want to connect as, such as that of the Lasso site administrator. For more information, see the *Configuring Lasso Professional 8* chapter *User Configuration* section. Your username will be stored in an encrypted

format within the snapshot file, and will only be readable by Lasso Studio.

- 8 In the Password field, enter the password for the username specified above. Your password will be stored in an encrypted format within the snapshot file, and will only be readable by Lasso Studio.
- 9 Select the Connect button. This causes Lasso Studio to connect to the Lasso Professional server environment and write the database schema information retrieved to the snapshot file.

If there is an error accessing the network, your Web server, or Lasso Professional 8, the error will be reported. You will need to correct the URL and check your network settings before selecting the Connect button again.

When you create a snapshot file using the Snapshot Editor it is automatically loaded in the Database Selector, which specifies the current snapshot in use by Lasso Studio. For information on how to use your new snapshot file, see the *Using Snapshot Files* section below.

Using Snapshot Files

Database snapshots are used to supply Lasso Studio with database schema information and sample record sets so that the data access features of Lasso Studio work, such as Form Builder, Site Builder, data access Lasso Objects, and Data Preview. The following sections describe how to use, update, and manipulate database snapshots in Lasso Studio.

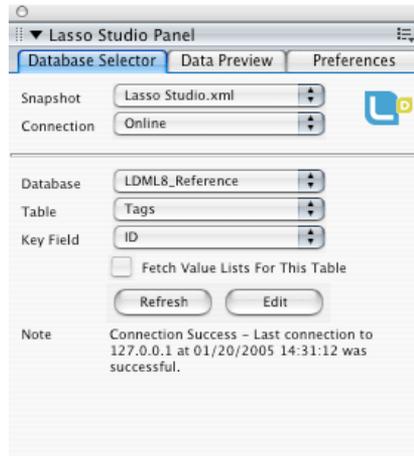
- *Database Selector* describes how to select the current snapshot, database, and table that is being used in Lasso Studio.
- *Updating Snapshot Files* describes how to update snapshot files with new database information from the server environment.
- *Loading Existing Snapshot Files* describes how to load existing snapshot files in Lasso Studio.
- *Copying and Distributing Snapshot Files* describes how to copy snapshot files and distribute them to other Lasso Studio developers.

Database Selector

The Database Selector allows you to choose the current snapshot, database, table and key field with which you want to work. To show the Database Selector, select Lasso Studio Panel from the Lasso Studio menu.

When developing pages, Lasso Studio can only actively use one snapshot at a time. The current snapshot in use is selected primarily via the Database Selector.

Figure 2: Database Selector



The points below describes the interface elements in the Database Selector and how to use them.

- **Snapshot** – Allows the current snapshot in use by Lasso Studio to be selected. A single snapshot represents the database information available from a single server environment. The Snapshot pull-down menu also contains snapshot commands for performing actions on the snapshot file, which are described in this chapter.
- **Connection** – Specifies whether the connection to the server environment of the current snapshot is online or offline. If Online is selected, then the server will be checked for any database schema changes each time a database or table is selected in the Database Selector (recommended). If Offline is selected, then only the database information already contained in the snapshot file can be used.
- **Database** – Allows you to select the current database you want to work with. The information for this database will be used whenever using a data access feature of Lasso Studio, such as Form Builder, Site Builder, data access Lasso Objects, and Data Preview.
- **Table** – Allows you to select the current table (or FileMaker layout) you want to work with. The information for this table will be used whenever using a data access feature of Lasso Studio, such as Form Builder, Site Builder, data access Lasso Objects, and Data Preview.
- **Key Field** – Allows you to select the current key field you want to work with. The key field is a field that contains a unique value for each unique record in the table, and is required for database actions that add,

- 4 In the Database pull-down menu, select the name of the database you wish to use. This will populate the Table pull-down menu with all tables currently in the database.
- 5 In the Table pull-down menu, select the name of the table (or FileMaker layout) in the database you wish to use. This will populate the Key Field pull-down menu with all fields currently in the table.
- 6 In the Key Field pull-down menu, select the name of the key field for the table.

The data access features of Lasso Studio will now be customized based on the choice of database, table and key field you made. Every time you create a new page, you should check the Database Selector to make sure that the database, table and key field are appropriate to the page in which you are working.

Updating Snapshot Files

This section describes updating a snapshot file with any new database information configured in the server environment. This includes adding new databases, tables, and fields not currently in the snapshot, and removing old database, tables, and fields no longer in the server environment.

To update a snapshot file:

- 1 In the Lasso Studio Panel, select the Database Selector tab.
- 2 Select the Refresh button. This contacts the Lasso Professional server environment and updates the snapshot with any new database schema information. Once this is done, we will verify that the new database information was added.
- 3 In the Database pull-down menu, select the name of the database you wish to use. Any databases recently configured in Lasso Site Administration should now appear here.
- 4 In the Table pull-down menu, select the name of the table (or FileMaker layout) in the database you wish to use. Any tables recently configured in Lasso Site Administration should now appear here.
- 5 In the Key Field pull-down menu, select the name of the key field for the table. Any fields recently configured in Lasso Site Administration should now appear here.

If you were able to successfully select a database, table, and key field that was recently configured, then you have successfully updated the snapshot file. The database information currently selected in the Database Selector

list of available snapshots within Lasso Studio. Snapshot files may also be saved to new locations with new names using the Database Selector or Snapshot Editor, as described below.

To copy and rename an existing snapshot file:

- 1 Select the Database Selector tab in the Lasso Studio panel.
- 2 In the Snapshot pull-down menu, select Save Snapshot As.... This will prompt for a file name and location for the new snapshot file.
- 3 In the Save As field, enter a file name for the new snapshot (e.g. MySnapshot.xml).
- 4 In the Location field, select the folder in which the snapshot will be saved. This is the Snapshots folder in the Lasso Studio 8 application folder by default, which is the recommended location for storing all snapshot files.
- 5 Select the Save button to save the new snapshot file.

Snapshot Files can be renamed anything you like, so long as the file name ends in a .xml extension and can be recognized as a file by Dreamweaver.

Editing Snapshots

The Snapshot Editor in Lasso Studio allows you to manually edit the connection information, schema information, and sample record set information contained in a snapshot file. The Snapshot Editor is launched by selecting Lasso Snapshot Editor... from the Lasso Studio menu, or by selecting Edit Snapshot... in the Snapshot pull-down menu in the Database Selector. The Snapshot Editor is divided into three sections, which are described below.

- *Editing Connection Information* describes how to use the Connection section of the Snapshot Editor to edit the server environment connection information in a snapshot file.
- *Editing Schema Information* describes how to use the Schema section of the Snapshot Editor to edit the database schema information in a snapshot file.
- *Editing Record Sets* describes how to use the Record Set section of the Snapshot Editor to edit sample record sets in the snapshot file

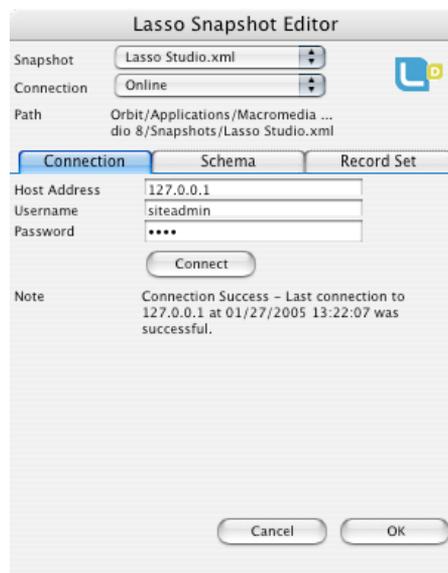
Editing Connection Information

The Connection section of the Snapshot Editor allows you to edit the server environment connection information for the snapshot file. There can only be one set of connection information per snapshot file.

This section describes editing the connection information in an existing snapshot file. It is only desirable to do this if the domain name or IP address of the server environment has changed, and the snapshot needs to be updated accordingly. The server address used should only correspond to the server using the databases contained in the snapshot file.

Note: A new snapshot file should be created for each individual server environment that is being connected to. Changing the connection information in an existing snapshot can invalidate the snapshot if the database information it contains is not reflected in the server environment.

Figure 3: Edit Connection Screen



To edit the connection information in a snapshot:

- 1 From the Lasso Studio menu, select Lasso Snapshot Editor.
- 2 In the Snapshot pull-down menu, select the name of the snapshot file you wish to edit. Note that the same snapshot commands available in the Snapshot pull-down menu in the Database Selector are also available here (with the exception of Edit Snapshot...).
- 3 In the Connection pull-down menu, select Online. Selecting Offline will still allow you to change the connection information in the snapshot file, but you will not be able to connect if your server environment is unavailable.

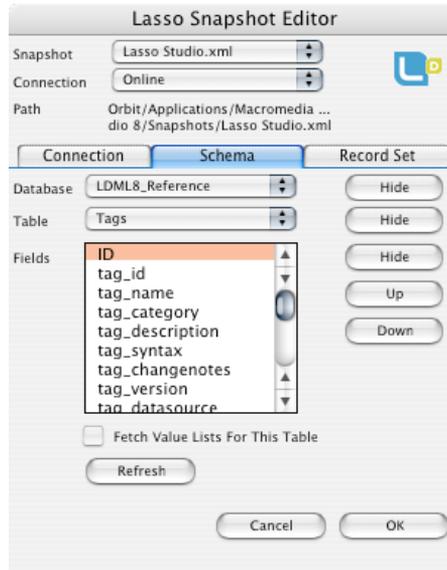
- 4 Select the Connection tab.
- 5 In the Host Address field, enter the IP address or host name of the machine on which the Lasso Professional server environment is installed. Use 127.0.0.1 if the server environment is on the machine you are currently using.
- 6 In the Username field, enter the Lasso Professional 8 username you want to connect as, such as that of the Lasso site administrator. For more information, see the *Configuring Lasso Professional 8* chapter *User Configuration* section. Your username will be stored in an encrypted format within the snapshot file, and will only be readable by Lasso Studio.
- 7 In the Password field, enter the password for the username specified above. Your password will be stored in an encrypted format within the snapshot file, and will only be readable by Lasso Studio.
- 8 Select the Connect button. This causes Lasso Studio to connect to the Lasso Professional server environment and reconcile the database schema information received with that of the existing snapshot file.

Editing Schema Information

The Schema section of the Snapshot Editor allows you to edit the database schema information in the snapshot file without connecting to your server. This affects the database information available in the Database Selector.

The Schema section works by allowing you to hide selected databases, tables, and fields from the Database Selector, Site Builder, and Lasso Inspectors making it so these schema objects are not available when developing in Lasso Studio. This is useful if there are a large number of databases, tables, or fields in a snapshot that will not be used for Web publishing. The Schema section also allows you to change the order of fields as they appear in the authoring environment interface and pages generated by the Form Builder and Site Builder.

Figure 4: Schema Editor



The points below describe how to show and hide databases, tables, and fields in a snapshot file, how to rearrange field order, and how to refresh the snapshot file.

- **Database** – A database can be hidden in the snapshot file by selecting the name of the database to be hidden from the Database pull-down menu, and then selecting the Hide button immediately to the right. This puts parentheses (...) around the database name to indicate that it is hidden.

A hidden database can be shown again by selecting the name of the database in parentheses from the Database pull-down menu, and then selecting the Show button immediately to the right.

Selecting the Show All... or Hide All... option in the Database pull-down menu automatically shows or hides all databases in the snapshot.

- **Table** – A table (or FileMaker layout) in a shown database can be hidden by selecting the name of the table to be hidden from the Table pull-down menu, and then selecting the Hide button immediately to the right. This puts parentheses (...) around the table name to indicate that it is hidden.

A hidden table can be shown again by selecting the name of the table in parentheses from the Table pull-down menu, and then selecting the Show button immediately to the right.

Selecting the Show All... or Hide All... option in the Table pull-down menu automatically shows or hides all tables in the snapshot.

- **Fields** – Individual fields can be hidden by selecting the name of the field in the Field list, and then selecting the Hide button immediately to the right. This puts parentheses (...) around the field name to indicate that it is hidden.

A hidden field can be shown again by selecting name of the database in parentheses from the Field list, and then selecting the Show button immediately to the right.

The field order may be changed by selecting a field, and then selecting the Up or Down button. This changes the order in which the fields will appear in pages built in Form Builder and Site Builder, and also in what order the fields will appear in pull-down menus in the Lasso Studio authoring environment interface.

If a value list is defined for a field then the field name will be appended with either (Y) or (N). (Y) means that the value list for the field has been downloaded into the snapshot. (N) means a value list is available for the field, but has not been downloaded into the snapshot.

- **Fetch Value Lists for this Table** – If the currently selected table has any value lists available then this option is available. When checked Lasso Studio will download all of the available value lists for the current table (or FileMaker layout). When unchecked, Lasso Studio will not download any further value list, but will also not delete any value lists already downloaded.
- **Refresh** – Selecting the Refresh button contacts the Lasso Professional server environment and updates the snapshot with any new database schema information. This adds new schema objects not currently in the snapshot, and removes schema objects that are no longer available in the server environment. This does not reset schema objects previously hidden or shown.

The procedure below describes the basic steps for editing database schema information within an existing snapshot file.

To edit database schema information in a snapshot:

- 2 From the Lasso Studio menu, select Lasso Snapshot Editor.
- 3 In the Snapshot pull-down menu, select the name of the snapshot you wish to edit.
- 4 In the Connection pull-down menu, select Online. Select Offline only if your server environment connection is unavailable, and you know the schema information you want is already in the snapshot file.
- 5 Select the Schema tab.

- 6 In the Database pull-down menu, select the name of the database you wish to hide, and select the Hide button next to it. This will put parentheses around the database name to indicate it is hidden. Repeat this step for a databases you wish to hide.
- 7 In the Table pull-down menu, select the name of a table in the current database you wish to hide, and select the Hide button next to it. This will put parentheses around the table name to indicate it is hidden. Repeat this step for all tables you wish to hide.
- 8 In the Fields list, select the name of a field in the current table you wish to hide, and select the Hide button next to it. This will put parentheses around the field name to indicate it is hidden. Repeat this step for all fields you wish to hide.
- 9 Select the OK button. This saves the changes to the snapshot file.

Editing Record Sets

The Record Set section of the Snapshot Editor allows you to edit the sample record set stored for each database in the snapshot file. These sample record sets are used whenever the Data Preview feature of Lasso Studio is used.

For information on Data Preview and editing record sets, see the *Data Preview* section.

Data Preview

Data Preview is a new feature in Lasso Studio that allows you to instantly view and edit the visually-displayed properties of stored data in a database in the Design and Live Data views in Dreamweaver. This feature allows you to develop and edit dynamic data content while seeing what the end-user will see when the page is published.

- *How Data Preview Works* describes the basic fundamentals of the Data Preview feature in Lasso Studio.
- *Data Preview Settings* describes how to enable and disable Data Preview in Lasso Studio
- *Using Data Preview* describes how to use the Data Preview feature in Lasso Studio.
- *Editing Record Sets* describes how to edit the sample record sets used by the Data Preview feature in Lasso Studio.

How Data Preview Works

Data Preview works by replacing [Field] Lasso Objects  in the Design view or Live Data view in Dreamweaver with sample data stored for that field in the snapshot file. A small set of sample data called a **Record Set** is stored for each database in the snapshot file when it is created or updated.

A default record set which consists of the first 10 records from a database and table is created automatically if none is stored. Record sets only store values for non-hidden fields from the current snapshot.

Note: Data Preview only works with text data, and cannot display images or binary files that are stored in a database within Dreamweaver. Data Preview also does not work with [Field] Lasso Objects that are inside HTML form elements, such as text areas and text fields.

[Field] Lasso Objects  are automatically placed within the Web pages created by the Form Builder or Site Builder wherever a field needs to be displayed. [Field] Lasso Objects may also be inserted into pages manually from the Lasso Data Access section of the Insert panel.

[Field] Lasso Objects will only display data from fields in a single record unless the [Records] ... [/Records] Lasso Objects  are present, which loop the [Field] Lasso Objects they contain for each record. For more information on [Field] and [Records] ... [/Records] Lasso Objects, see the *Using Objects and Tags* chapter.

Note: When using Data Preview to display multiple records, the [Records] ... [/Records] Lasso Objects will remain around the first record, and all subsequent records will be displayed below.

Figure 5: Before Data Preview

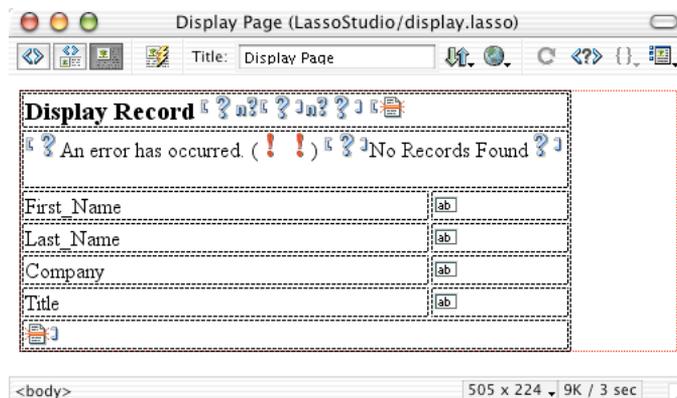


Figure 6: After Data Preview

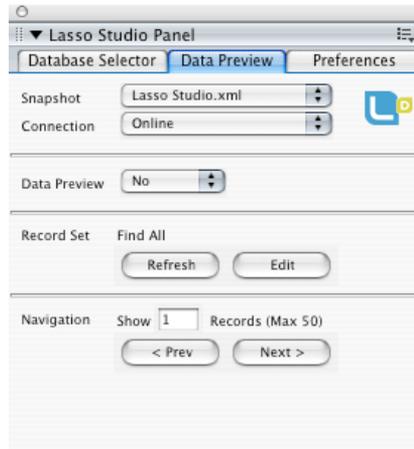


Data Preview is used as a visual aid to ease the process of designing and customizing data-driven pages. Data Preview simply displays sample data from a snapshot file, and does not dynamically test or parse Lasso code in the page you are working on. There is one sample record set stored per table in the snapshot file, and each sample record set may be customized using the Record Set section of the Snapshot Editor.

Data Preview Settings

Data Preview settings are managed from the Data Preview section of the Lasso Studio panel, as described in the *Using Lasso Studio* chapter. Here, Data Preview can be enabled or disabled, and the sample data may be both edited and navigated through.

Figure 7: Data Preview Settings



The following describes the options available in the Data Preview section and how they are used.

- **Snapshot** – Allows the current snapshot being used to be selected. This determines what database schema and what sample data will be used.
- **Connection** – This allows you to specify whether the current connection to the server environment is online or offline. If it is online, then Lasso Studio will dynamically query the server environment for new data whenever the sample record set is edited. If the connection is offline, then only existing information in the snapshot file can be used.
- **Data Preview** – Selecting Yes enables Data Preview, which will populate all [Field] Lasso Objects with sample data. Selecting No disables Data Preview.
- **Record Set** – Shows the current record set being used for Data Preview, and allows the sample record set to be edited by selecting the Edit button. The default record set is produced by a FindAll action performed in the database currently selected in the Database Selector. For more information on editing record sets, see the following *Editing Record Sets* section.
- **Navigation** – Allows the current sample record(s) shown to be changed in a Web page while using Data Preview. If viewing a page with a single record shown, selecting the < Prev or Next > button will show the previous or next record in the record set in the page. If viewing a page with multiple records shown (using the [Records] ... [Records] Lasso Objects), entering an integer between 1 and 50 in the Show Records field shows that many records in the page.

Use the procedure below to enable data preview for the database and table currently selected in the Database Selector.

To enable Data Preview:

- 1 In the Lasso Studio panel, select the Data Preview tab.
- 2 Next to Data Preview, select Yes.

Data Preview is now enabled. If [Field] Lasso Objects are not replaced with data, try selecting the Refresh button under Record Set. Editing the current record set is described the *Editing Record Sets* section.

Using Data Preview

When Data Preview is enabled, all [Field] Lasso Object icons in a page will be replaced by data for that field stored in the sample record set in the snapshot file. The sections below describe how to use Data Preview when authoring Web pages or Lasso pages.

Viewing Record Sets

When Data Preview is enabled, you should see sample field data for all [Field] Lasso Objects. The values shown in these fields may be changed using the Navigation functions of the Data Preview section of the Lasso Studio panel.

The Navigation functions allow the sample record set to be browsed in a Web page while using Data Preview. If viewing a page with a single record shown, selecting the < Prev or Next > button will show the previous or next record in the record set in the page. If viewing a page with multiple records shown (using the [Records] ... [Records] Lasso Objects), entering an integer between 1 and 50 in the Show Records field shows that many records in the page.

Editing Visual Properties of Fields

While using Data Preview, the visually-displayed HTML properties of all fields shown may be edited. For example, the font properties of a sample field may be changed by selecting the object and modifying the HTML properties using the tools in Dreamweaver. All HTML properties for each field are properly created in the source code when edited, and the field will appear in Lasso Studio exactly as the end-user will see it when the page is published.

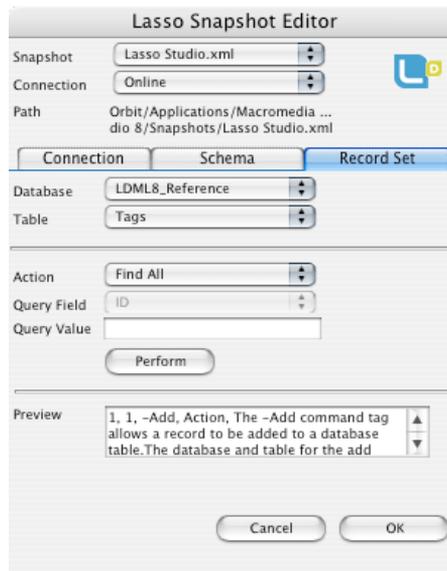
Note: HTML properties apply to an entire field. As field contents are dynamic and subject to change, individual parts of a field cannot be edited within Dreamweaver.

Editing Record Sets

The sample record set for any database and table in the current snapshot can be edited via the Record Set section of the Snapshot Editor. This section can be instantly accessed for the current database and snapshot by selecting the Edit button in the Data Preview section of the Lasso Studio panel.

A default record set consists of the first 10 records from a database and table, and is created automatically upon snapshot creation. Record sets only store values for non-hidden fields from the current snapshot.

Figure 8: Edit Record Set



The following describes the options available in the Record Set section and how they are used.

- **Database** – Specifies the database which contains the record set being edited.
- **Table** – Specifies the table (or FileMaker layout) which contains the record set being edited.
- **Action** – Specifies the database action which produces the record set. This can be FindAll to find all records in the table, or Search to find a set of records with a specific field value.
- **Query Field** – Specifies the field being searched in a query defined below. Used only for producing a specific record set via a Search action.

- **Query Value** – Specifies the value being searched for in the field defined above. Used only for producing a specific record set via a Search action.
- **Perform** – Selecting the Perform button executes the query defined in the Define Query box and shows the results in the Data Preview field. The new record set is not saved until the OK button is selected.

Note: The Perform button is disabled if the connection is Offline.

- **Data Preview** – Previews the current record set data for inspection (display only). The values for each field are shown in a tab-delimited format. Any new record set shown is not saved until the OK button is selected.

Use the procedure below to edit a sample record set for a database and table in a snapshot file.

To create or edit a record set:

- 1 Open the Lasso Snapshot Editor either from the Lasso Studio menu, or by using the Edit button in the Data Preview section of the Lasso Studio panel.
- 2 In the Snapshot pull-down menu, select the snapshot you wish to use.
- 3 In the Connection pull-down menu, select Online.
- 4 Select the Record Set tab.
- 5 In the Database pull-down menu, select the name of the database from the snapshot you wish to use.
- 6 In the Table pull-down menu, select the name of the table (or FileMaker layout) from the snapshot you wish to use.
- 7 In the Action pull-down menu, select the action which creates the sample record set. This can be FindAll or Search. If FindAll is selected, skip to step 10.
- 8 If Search was selected as the action, select a field from the Query Field pull-down menu to search.
- 9 If Search was selected as the action, enter a value to search for in the Query Value field. The value entered here should exist in the database for the field specified above in one or more records.
- 10 Select the Perform button to preview your new record set in the Data Preview field.
- 11 Select the OK button the save the record set to the snapshot file.

This new sample record set will now be used for this database and table when in Data Preview mode.

6

Chapter 6

Using Form Builder

The Form Builder is a panel in Lasso Studio that allows you to instantly create forms and response pages which have all the necessary attributes to perform common database actions using Lasso Professional, such as searching, adding, updating, and deleting records. It creates database actions based on the database you selected in the Database Selector.

- *Introduction* provides conceptual information about the Lasso code generated by the Form Builder.
- *Using Form Builder* provides an overview of using the Form Builder panel.
- *Searching Records* describes creating forms for searching records in a database, and describes creating pages that list and display records.
- *Adding Records* describes creating forms and response pages for adding records to a database.
- *Updating and Deleting Records* describes creating forms and response pages for updating and deleting records in a database.

Note: The code generated by the Form Builder obeys the global syntax preference for parentheses syntax or colon syntax. This preference can be modified in the Lasso Studio panel.

Introduction

Lasso is a flexible tool so there is often more than one way to perform a desired database action. There are some basic concepts which are important to understand before you begin building forms for use with Lasso Professional.

Inline Method

Most interaction with a user on the Web is done via HTML forms. The form may provide text inputs, pop-up menus, check boxes, etc. to the user. The user enters their data and selects a button to submit the data and then go on to a response page. The response page then presents results based on the data the user entered.

Lasso Studio uses the Inline Lasso method to make this type of interaction possible. An Inline-based action is one in which all the functions necessary for Lasso Professional to perform the action are included in the response page using Lasso tags. For more information on inlines, see the Lasso 8 Language Guide. The form which is presented to the user need only contain the inputs necessary for the user to customize the user-defined attributes. The actual Lasso action is generally performed in an Inline tag on the response page.

The Form Builder helps you to create both forms and response pages for use in an Inline-based Lasso site. The Form Builder creates forms and response pages based on the database and table you selected in the Database Selector to perform common Lasso actions, such as searching, adding, updating, and deleting records. It also allows you to create simple record listing and display response pages.

Serving Pages With Lasso tags

The Form Builder works by inserting Lasso tags into your HTML code that perform the desired database action when served by a Lasso Professional server. Lasso tags include substitution and container tags which are delimited by square brackets, e.g. [Field], and command tags which are designated by a leading hyphen, e.g. -Search.

Lasso tags should only be included in Web pages that are processed by Lasso. This means these tags need to be on a page that ends in a suffix that is set in the Web server to be processed by Lasso by, e.g. .lasso. When Lasso Professional processes a .lasso page, it dynamically replaces all Lasso syntax with calculated data and results from the database.

Lasso Professional can only process .lasso pages by default, therefore it is recommended that all pages created using Form Builder end in a .lasso extension unless your server environment is configured otherwise. For instructions on how to configure Lasso Professional to process pages with other extensions (e.g. .html), see the installation chapters of the Lasso Professional 8 Setup Guide.

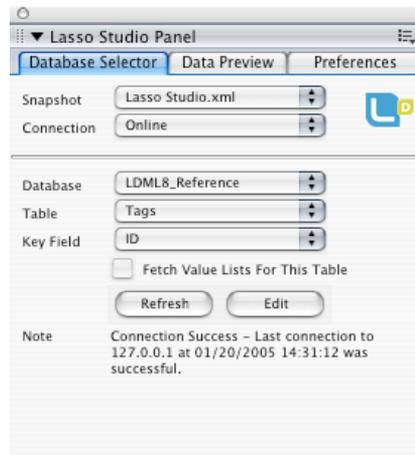
Using the Form Builder

This section describes all the different forms and pages that can be created using the Form Builder palette.

Database Selector

The Database Selector is used to select the database, table, and key field that you wish to use for any code created by the Form Builder. The Database Selector should always be consulted to make sure the right database and table are selected before the Form Builder is used.

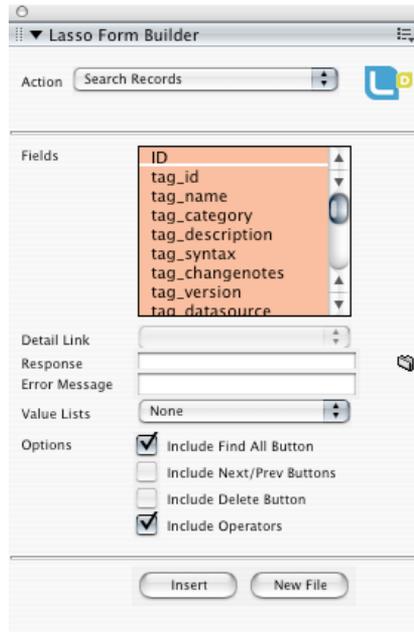
Figure 1: Database Selector



Form Builder Palette

The Form Builder is launched by choosing Show Lasso Form Builder from the Lasso Studio menu in Dreamweaver. The Form Builder panel allows you to customize and create the form which is going to be inserted into your document window. The options are different depending on what type of form you are inserting.

Figure 2: Lasso Form Builder



Form Builder Panel

The interface elements of the Form Builder panel are described below.

- **Action** – Specifies the action of the page you will create. The options are described below.

Search Records – Allows you to create a form for searching a database.

Add Record – Allows you to create a form for adding a record to a database.

Update Record – Allows you to create a form for updating or deleting a record in a database. This form should be contained in a page which is the response for a detail link from a listing page or for an add or update form.

List Records – Allows you to create code that displays a list of records from a database, and contains a Lasso `[[inline]]` tag that performs a search action. This page may be preceded by a page which contains a form for searching records.

Display Record – Allows you to create code that displays a single record from a database, and contains a Lasso `[[inline]]` tag that performs a search action. This page may be preceded by a page which contains a form for

searching records, or by a page that lists records and provides links to detail.

Add Response – Allows you to create code that adds a record to a database and displays the results. This page contains a Lasso [Inline] tag that performs an add action, and should be preceded by a page which contains a form for adding records.

Update Response – Allows you to create code that updates or deletes a record in a database and displays the results. This page contains a Lasso [Inline] tag that performs an update or delete action, depending on the form button selected. This page should be preceded by a page which contains a form for updating records.

- **Fields** – Allows you to select the fields that will be displayed in the form or response page being created. One or more fields may be selected. What fields are displayed here depends on the database and table selected in the Database Selector.
- **Detail Link** – Allows you to specify the file name of the detail page that the user will be taken to after selecting a link for a record in a listing page. Applicable for the List Records action only.
- **Response Page** – Allows you to specify the file name of the response page for a form. Applicable for the Search Records, Add Record, and Update Record actions only.
- **Error Message** – Allows you to specify a custom error message that will be displayed if an error occurs when using the page. This message will be displayed next to a generated Lasso error message and error code, which are included in the page by default even if no custom error message is specified.
- **Value Lists** – Allows you to select whether you would like fields with value lists in your database to use pull-down menus, radio buttons, or check boxes in the search and update forms. Dynamic means that the page will populate the HTML selection items from the database, whereas Static means the HTML selection values will be hard-coded in the page.

MySQL Note: Value list fields in MySQL include ENUM and SET types.

- **Options** – Allows you to specify various options for the code you will create. These options are described below.

Include FindAll Button – Allows you to include a Find All Records button in your search form that finds all records in a table. Applicable for search forms only.

Include Next/Prev Buttons – Allows you to include < Prev and Next > buttons in your listing page for navigating through record sets generated by a search action. Applicable for record listing and display pages only.

Include Delete Button – Allows you to include a Delete Record button in your update form that deletes the record shown. Applicable for update forms only.

Include Operators – Allows you to include search operators for each field in your search form, such as Begins With, Equals, or Contains. Applicable for search forms only.

- **Insert** – Selecting the Insert button inserts the code in the current document at the location initially selected in the document.
- **New File** – Selecting the New File button automatically creates a new document that contains the code instead of inserting it into the current document.

The Form Builder panel is used by selecting and entering information about the code to be generated (as defined below), and then selecting the Insert or New File button. See the following sections for step-by-step instructions on creating various forms and response pages using the Form Builder.

Tips For Using Form Builder

This section provides various tips for using Form Builder when building Lasso solutions.

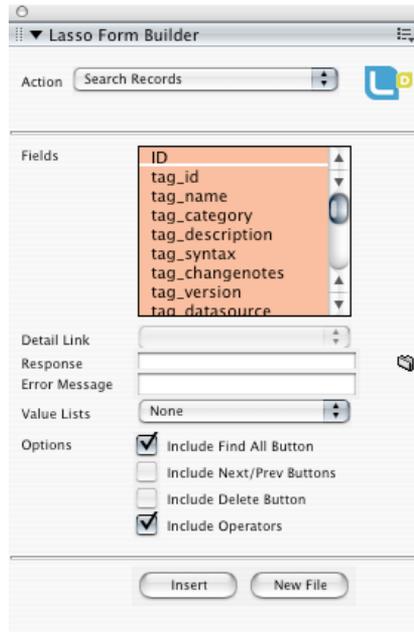
- Always make sure your form has an appropriate response page created for it. In other words, if your form is intended to perform a search action, then make sure it submits to an appropriate response page that lists or displays records.
- If desired, both the form and response code can be inserted in the same page instead of different pages. This is common for search pages, where the resulting record listing appears under the used search form after a submission. When doing this, make sure the Response Page is set to be the same page that contains the form.

Searching Records

The Form Builder allows you to easily create forms that submit search queries to a Lasso response page, which searches and displays records in a database depending on the form values submitted. The Form Builder also allows you to create the search response pages that list or display records from a database.

- *Creating Search Forms* describes creating a form that submits a search query to a response page, which searches a database.

Figure 2: Lasso Form Builder Search Records



4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys on Macintosh.

5 In the Response Page field, specify the file name of the response page you want the user to see after submitting the form. You can either select the folder icon to choose a page in your site, or you can enter a file name into the text field manually.

Note: While the search response page is typically a record listing page, it can also be a record display page or update form depending on your purpose. Record listing, record display, and update forms are described later in this chapter.

6 Select whether you would like values lists in the database to be displayed as pop-up menus, check boxes, or radio buttons. Using Dynamic value lists will insert additional Lasso code that dynamically extracts the value list information from the database when the page is loaded (via an [Inline] Lasso Object).

7 Next to Options, select Include FindAll Button if you wish to include a button that instantly finds all records regardless of search criteria.

- 8 Next to Options, select Include Operator, if you want to include selectable search operators next to each field. Search operators include Begins With, Equals, Contains, and more.
- 9 Select the Insert button to insert the search form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create a search form that contains an HTML text input for each field specified in the Form Builder panel. The <Form> HTML tag will contain the name of the response page (e.g. Response.lasso) for the Action attribute, meaning the form will submit to that page. For information on how Lasso code is used within a search form, see the *Searching and Displaying Records* chapter in the Lasso 8 Language Guide.

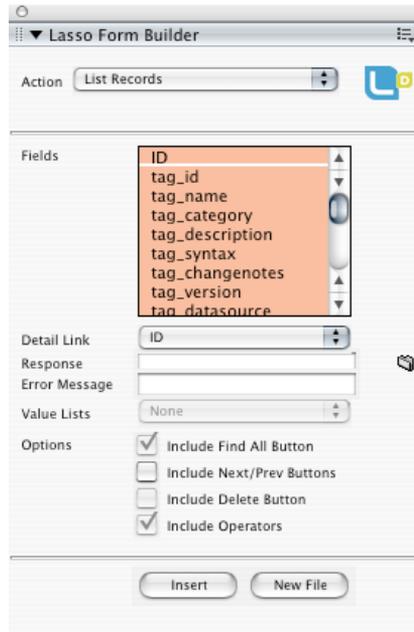
Creating Record Listing Pages

This section describes creating a record listing page to be used as a response for a search form. This page requires a Lasso search form to submit data to the page in order to display data.

To create code for listing multiple records:

- 1 First, use the Database Selector to ensure the same database, table, and key field values are selected as used in your search form. This will cause the Form Builder to insert code that searches the database and table specified.
- 2 If you are using Form Builder to insert code into an existing page, then select the location in your page in which you want to insert the code. This can be done simply by placing the text insertion tool in Dreamweaver at the location desired.
- 3 In the Form Builder panel, select List Records from the Action pull-down menu.

Figure 3: Lasso Form Builder List Records



4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys on Macintosh.

5 In the Detail Link pull-down menu, select the name of one field which will be linked to a detail page. The end-user will be able to click on this link to be taken to the detail page for the record shown.

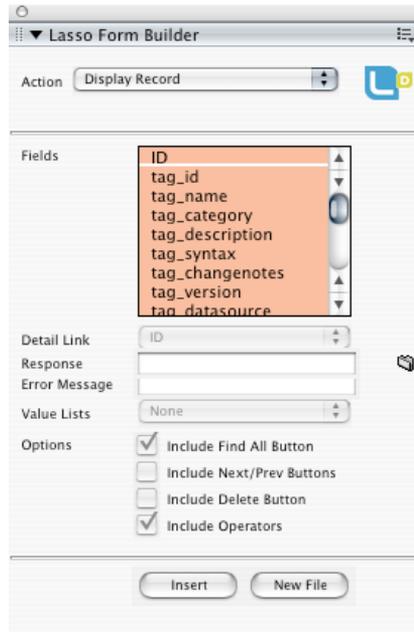
6 In the Response Page field, specify the file name of the response page you want the user to see after clicking on the detail link. You can either select the folder icon to choose a page in your site, or you can enter a file name into the text field manually.

Note: This can be a record display page as described in the next section, or can be an update page as described later in this chapter.

7 In the Error Message field, enter an error message you want displayed if an error occurs while using the page. This message will be displayed next to a generated Lasso error message and error code, which are included in the page by default even if no custom error message is specified.

8 Next to Options, select Include Next/Prev Buttons to include < Prev and Next > buttons for navigating through record sets. This is only necessary if the

Figure 4: Lasso Form Builder Display Records



4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys in Mac OS X.

5 Select the Insert button to insert the add form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create code that contains an [Inline] tag which performs a search action on the database. The [Action_Params] tag used in the beginning of the [Inline] tag collects the field information passed from the previous search form and passes it to the [Inline] tag. All data in the record returned is displayed via the [Field] tags, which are contained inside the [Inline] ... [Inline] tags. For information on how Lasso code is used within a display pages, see the *Searching and Displaying Records* chapter in the Lasso 8 Language Guide.

Creating Standalone Listing and Display Pages

This section describes creating listing and display pages that show records instantly upon being loaded in a browser. These pages do not require a

form to be submitted, and will perform the same database query every time they are loaded.

Converting post-form listing and display pages to standalone pages simply involves editing the [Inline] tag in the page so that the database query is hard-coded, rather than dependent upon the field information submitted in a form.

To create a standalone listing or display page:

- 1 Create a record listing or display page, as described previously. The procedures following are identical for either listing or display pages.
- 2 In the Design view of the page created, select the left-most [Inline] Lasso Object (shown below). This will display properties for the object in the Lasso Inspector panel, which can be accessed from the Lasso Studio menu.



- 3 Follow the instructions in the [Inline] tag example in the *Using Objects and Tags* chapter *Using Lasso Objects* section to customize your [Inline] tag to display the values you want. This can be a search for one or more records that contain specific field values, or can be a FindAll action.

Note: Customizing [Inline] tags to display records is also described in the *Searching and Displaying Records* chapter in the Lasso 8 Language Guide.

- 4 Save the listing or display page.

Adding Records

The Form Builder allows you to easily create forms and response pages for adding records to a database. Both an add form and response page must be created as described in this section in order to successfully add a record.

- *Creating Add Forms* describes creating a form that submits new record data to a response page, which adds the record to a database.
- *Creating Add Response Pages* describes creating an add response page which adds a record to a database based on information submitted in an add form.

Creating Add Forms

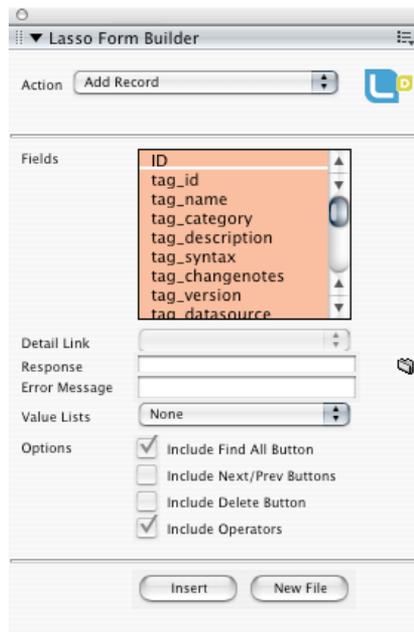
This section describes creating a form that adds a new record to a database, and displays the results in a response page. The add form itself does not perform the add action, but submits data to a response page which

contains a Lasso [Inline] tag that adds the record to the database with the values specified in the form.

To create a form for adding a record:

- 1 First, use the Database Selector to select the database, table, and key field that you wish to use for your search form. This will cause the Form Builder to insert code that searches the database and table specified.
- 2 If you are using Form Builder to insert code into an existing page, then select the location in your page in which you want to insert the code. This can be done simply by placing the text insertion tool in Dreamweaver at the location desired.
- 3 In the Form Builder panel, select Add Record from the Action pull-down menu.

Figure 5: Lasso Form Builder Add Records



- 4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys in Mac OS X.

Key Field Note: Most data sources auto-generate a key field value during an add operation. In these cases, the key field should not be included among the fields you wish the end-user to add (e.g. the ID field in MySQL).

- 5 In the Response Page field, specify the file name of the response page you want the user to see after submitting the form as described in the next section). You can either select the folder icon to choose a page in your site, or you can enter a file name into the text field manually.
- 6 Select whether you would like value lists in the database to be displayed as pop-up menus, check boxes, or radio buttons. Using Dynamic value lists will insert additional Lasso code that dynamically extracts the value list information from the database when the page is loaded (via an [Inline] Lasso Object).
- 7 Select the Insert button to insert the add form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create an add form that contains an HTML text input for each field specified in the Form Builder panel. The <Form> HTML tag will contain the name of the response page (e.g. response.lasso) for the Action attribute, meaning the form will submit to that page. For information on how Lasso code is used within an add form, see the *Adding and Updating Records* chapter in the Lasso 8 Language Guide.

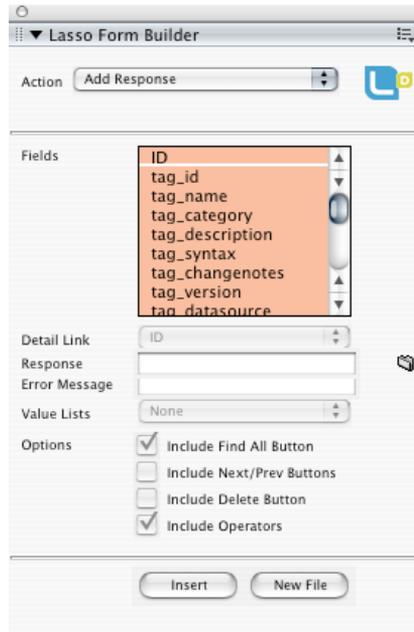
Creating Add Response Pages

This section describes creating an add response page for an add form, which contains the necessary code to add a record to the database based on the values submitted in the form.

To create code for adding and displaying a record:

- 1 First, use the Database Selector to ensure the same database, table, and key field values are selected as used in your add form. This will cause the Form Builder to insert code that searches the database and table specified.
- 2 If you are using Form Builder to insert code into an existing page, then select the location in your page in which you want to insert the code. This can be done simply by placing the text insertion tool in Dreamweaver at the location desired.
- 3 In the Form Builder panel, select Add Response from the Action pull-down menu.

Figure 6: Lasso Form Builder Add Response



- 4 In the Fields list, select the fields from the current database that should be included on the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys in Mac OS X.
- 5 In the Error Message field, enter an error message you want displayed if an error occurs while using the page. This message will be displayed next to a generated Lasso error message and error code, which are included in the page by default even if no custom error message is specified.
- 6 Select the Insert button to insert the add form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create code that contains an `[[Inline]]` tag which performs an add action on the database. The `[Action_Params]` tag used in the beginning of the `[[Inline]]` tag collects the field information passed from the previous form and passes it to the `[[Inline]]` tag, which performs the data source action. For information on how Lasso code is used within add pages, see the *Adding and Updating Records* chapter in the Lasso 8 Language Guide.

Updating and Deleting Records

The Form Builder can also be used to create forms that update or delete records in a database. Update forms are a combination of a form and a display page, in that data must be drawn from a database via an [Inline] Lasso Object and displayed in a form that can be edited. Both an update form and response page must be created as described in this section in order to successfully update or delete a record.

- *Creating Update Forms* describes creating a form that submits updated record data to a response page, which updates the record in a database.
- *Creating Update Response Pages* describes creating an update response page which updates a record to a database based on information submitted in an update form.

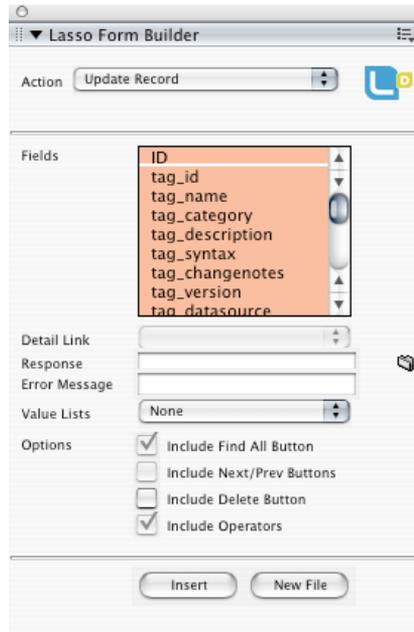
Creating Update Forms

This section describes creating an update page which displays a record and allows it to be updated or deleted. This page must be preceded by a Lasso search form or listing page in order to display record data, as described previously in this chapter.

To create a form for updating or deleting a record:

- 1 First, use the Database Selector to select the database, table, and key field that you wish to use for your search form. This will cause the Form Builder to insert code that searches the database and table specified.
- 2 If you are using the Form Builder to insert code into an existing page, then select the location in your page in which you want to insert the code. This can be done simply by placing the text insertion tool in Dreamweaver at the location desired.
- 3 In the Form Builder panel, select Update Record from the Action pull-down menu.

Figure 7: Lasso Form Builder Update Records



4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys in Mac OS X.

Key Field Note: Most data sources require that the key field remain constant during an update operation. In these cases, the key field should not be included among the fields you wish the end-user to update (e.g. the ID field in MySQL).

- 5 In the Response Page field, specify the file name of the update response page you want the user to see after submitting the form (as described in the next section). You can either select the folder icon to choose a page in your site, or you can enter a file name into the text field manually.
- 6 In the Error Message field, enter an error message you want displayed if an error occurs while using the page. This message will be displayed next to a generated Lasso error message and error code, which are included in the page by default even if no custom error message is specified.
- 7 Select whether you would like value lists in the database to be displayed as pop-up menus, check boxes, or radio buttons. Using Dynamic value lists will insert additional Lasso code that dynamically extracts the value

list information from the database when the page is loaded (via an [Inline] Lasso Object).

- 8 Next to Options, select Include Delete Button to include a button for deleting the record shown.
- 9 Select the Insert button to insert the add form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create an update form that contains an HTML text input for each field specified in the Form Builder panel. The <Form> HTML tag will contain the name of the response page (e.g. response.lasso) for the Action attribute, meaning the form will submit to that page.

Each text field input contains a [Field] tag, which displays the value for each field to be edited. The [Field] tags are populated by an [Inline] search tag, which uses the [Action_Params] tag to collect record information passed from the previous page and display the appropriate record from the database.

For information on how Lasso code is used within an update form, see the *Adding and Updating Records* chapter in the Lasso 8 Language Guide.

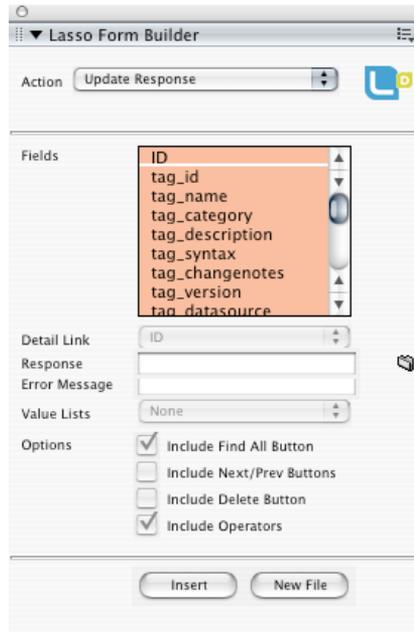
Creating Update Response Pages

This section describes creating an update response page for an update form, which contains the necessary code to update or delete a record in the database based on the values submitted in the form.

To create code for updating or deleting a record:

- 1 First, use the Database Selector to ensure the same database, table, and key field values are selected as used in your update form. This will cause the Form Builder to insert code that searches the database and table specified.
- 2 If you are using the Form Builder to insert code into an existing page, then select the location in your page in which you want to insert the code. This can be done simply by placing the text insertion tool in Dreamweaver to the location desired.
- 3 In the Form Builder panel, select Update Response from the Action pull-down menu.

Figure 8: Lasso Form Builder Update Response



- 4 In the Fields list, select the fields from the current database that should be included in the form. You can choose more than one field by holding down the Shift or Control keys in Windows or the Shift or Command keys in Mac OS X.
- 5 In the Error Message field, enter an error message you want displayed if an error occurs while using the page. This message will be displayed next to a generated Lasso error message and error code, which are included in the page by default even if no custom error message is specified.
- 6 Select the Insert button to insert the add form code in the page selected previously. Alternately, select the New File button to create the search page code in a new file in Dreamweaver.

This will create code that contains an [Inline] tag which performs an update or delete action on the database, depending on whether the Update or Delete button was selecting in the previous form. The [Action_Params] tag used in the beginning of the [Inline] tag collects the field information passed from the previous form and passes it to the [Inline] tag. For information on how Lasso code is used within an update page, see the *Adding and Updating Records* chapter in the Lasso 8 Language Guide.

7

Chapter 7

Using Site Builder

The Site Builder allows you to create Lasso-based sites using a simple dialog-driven interface. A site created via the Site Builder is a set of pages that perform database actions on a single database.

The methodology behind the pages created by the Site Builder is the same as that used by the Form Builder. Please see the *Using Form Builder* chapter for more information about the Lasso programming methodology employed.

- *Overview* provides an overview of using the Site Builder.
- *Search Site Creation* describes creating a site for searching records in a database, and describes creating pages that list and display records.
- *Add Site Creation* describes creating a site for adding records to a database.
- *Site Builder Pages* describes the Lasso code and structure generated in each individual page created by the Site Builder.
- *Testing Your Site* describes testing the sites created by Site Builder.
- *Site Builder Templates* describes creating and using HTML templates in which to insert Lasso code generated by the Site Builder.

Note: The code generated by the Site Builder obeys the global syntax preference for parentheses syntax or colon syntax. This preference can be modified in the Lasso Studio panel.

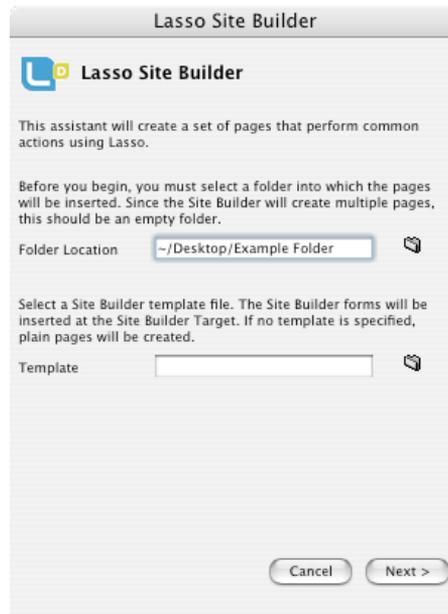
Overview

The Site Builder is launched by choosing Lasso Site Builder... from the Lasso Studio menu in Dreamweaver. The Site Builder presents a series of screens that allow you to customize a site that will be created. The options are different depending on what type of site you are creating. The site types and options are described in the sections below.

Site Builder Interface

The start page of the Site Builder is shown in *Figure 1: Lasso Site Builder* and contains two fields. The upper Site Folder field allows you to select the folder where you would like your site to be stored. You can choose an existing folder or create a new, empty folder.

Figure 1: Lasso Site Builder



The lower Template field allows you to optionally select a template to use as the basis for each of the files which are created by the site builder. Select the button to bring up a dialog box which allows you to choose a file to use as the template.

The file you choose must have a Site Builder Target Lasso Object placed in it, which can be found in the Lasso Form Insert panel. For more information about creating Site Builder templates, see the *Site Builder Templates* section at the end of this chapter.

Selecting the Next > button takes you to the Choose a Site Action screen, which is described next.

Site Types

Two basic sites can be created using the Site Builder; namely, a site to search, list, display, and update records; and a site to add, display and update records.

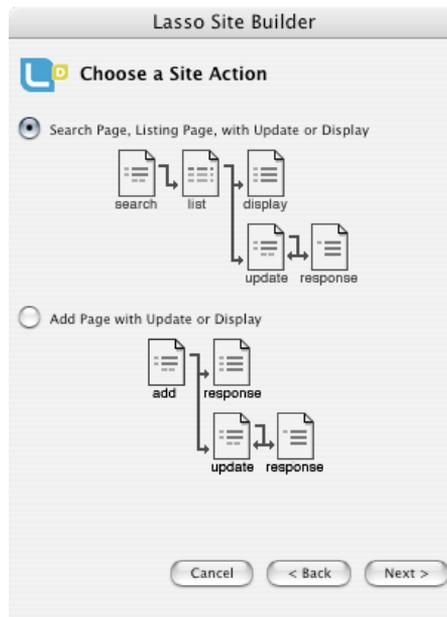
Search Page with Listing Results and Display or Update Page

The site consists of a search page and listing results page. One field from the listing page is a link to either a display page or an update form. The update form can have an optional delete button.

Add Page with Display or Update Page

The site consists of an add page with a response display page or update form. The update form can have an optional delete button.

Figure 2: Site Types



The Choose a Site Action screen allows you to select whether you want to build one of the sites listed above. The Site Builder will guide you through the screens appropriate to the site you choose to create.

For information on creating a “search” site, see the *Search Site Creation* section of this chapter. For information on creating an “add” site, see the *Add Site Creation* section of this chapter.

Customizing Your Site

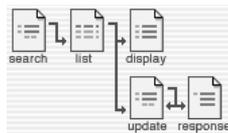
The pages which are generated by the Site Builder contain all the programming necessary to perform the selected Lasso actions. All individual pages and Lasso code generated by the Site Builder are summarized in the *Site Builder Pages* section of this chapter. However, these pages may require customization in order to fit with the other pages in your Web site. Dreamweaver’s built-in HTML editing tools can be used to rapidly transform a site created using the Site Builder to any appearance.

Lasso Studio also provides the Site Builder Target Lasso Object as a method of inserting code generated by the Site Builder into pre-developed HTML pages. For more information, see the *Site Builder Templates* section of this chapter.

Search Site Creation

This section describes building a basic data-driven site using the Site Builder. The Site Builder presents a series of screens that allow you to customize the site which is going to be created. In this section, we will show you how to use the Site Builder to create a set of pages to search your database, display the results of the search, and allow individual records to be updated.

Figure 3: Search Site



Creating a Search Site

Every search site contains a search form followed by a results listing page. Each item in the results listing links either to a display page with more

information about the particular item or an update page which allows the client to modify the item. The update page can have an optional delete button which takes the user to a feedback page when the item is successfully deleted from the database. Finally, every site has an error page which is returned to the client in case of any errors.

To create a basic search site:

- 1 From the Lasso Studio menu, select Lasso Site Builder.... This launches the Lasso Site Builder screen.
- 2 For the Site Folder field, select the folder where you would like your site to be stored, such as in an empty folder inside the root of your Web server. For instance, create a folder called MySite at the root level of your Web server so pages within it can be loaded with the following URL:
`http://127.0.0.1/MySite/`

Warning: Several individual documents will be created in the destination you select. It is recommended that you use an empty folder for the destination to avoid any files being overwritten.

- 3 Leave the Template field blank to have the Site Builder create files based on the default template. Otherwise, it is possible to use an HTML template file to which the code for each page created is inserted and named appropriately. For more information on using templates, see the *Site Builder Templates* section of this chapter.
- 4 Select the Next button to go on to the next screen.
- 5 For the site type, select the Search Page, Listing Page, with Update or Display radio button.
- 6 Select the Next button to go on to the next screen.
- 7 In the Database pull-down menu, select the name of the database you wish to use for the site. The default database shown will be the one selected in the Database Selector.
- 8 In the Table pull-down menu, select the name of the table (or FileMaker layout) you wish to use.
- 9 In the Key Field pull-down menu, select the name of the key field for the table. This is typically the ID field for MySQL databases. The key field must contain a unique value for every record in the database.

Figure 1: Lasso Site Builder Database Selector

The screenshot shows a dialog box titled "Lasso Site Builder" with a subtitle "Lasso Database Selector". It features four dropdown menus for configuration:

- Database:** LDML8_Reference
- Table:** Tags
- Key Field:** ID
- Value Lists:** None

At the bottom of the dialog are three buttons: "Cancel", "< Back", and "Next >".

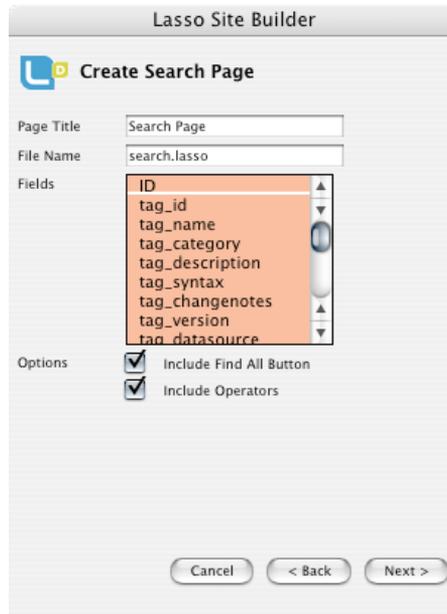
FileMaker Note: An Internal Record ID option appears for Filemaker databases, which represents Filemaker’s internal record ID field. Filemaker databases will use this regardless of what field is selected.

- 10** In the Value Lists pull-down menu, select whether you would like fields with value lists in your database to use dynamic or static pull-down menus, radio buttons, or check boxes in the search and update forms. Dynamic means that the page will populate the HTML selection items from the database (via an [Inline] Lasso Object), whereas Static means the HTML selection values will be hard-coded in the page.

MySQL Note: Value list fields in MySQL include ENUM and SET types.

- 11** Select the Next button to go on to the next screen to set options for the search page.

Figure 5: Lasso Site Builder Create Search Page



- 12 Enter a title for the search page in the Page Title field (e.g. Search Page).
- 13 Enter a file name for the search page in the File Name field (e.g. Search.lasso). It is important that this page end with a .lasso extension.
- 14 In the Fields list, select all fields that you want to appear on your pages.
- 15 Next to Options, check the Include FindAll Button check box if desired. This will allow the search page to find all records in the database upon selecting this button.
- 16 Next to Options, check the Include Operators check box if desired. This will include selectable search operators (Equals, Begins With, Contains, etc.) for each field in the form. The operators available for each field may vary depending on the data type of the field (e.g. numeric vs. alphanumeric). For more information on search operators, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.
- 17 Select the Next button to continue on to the Listing Page screen.
- 18 Enter a title for the listing page in the Page Title field (e.g. Listing Page).
- 19 Enter a file name for the listing page in the File Name field (e.g. Listing.lasso). It is important that this page end with a .lasso extension.
- 20 In the Fields list, select all fields that you want to appear in your pages.

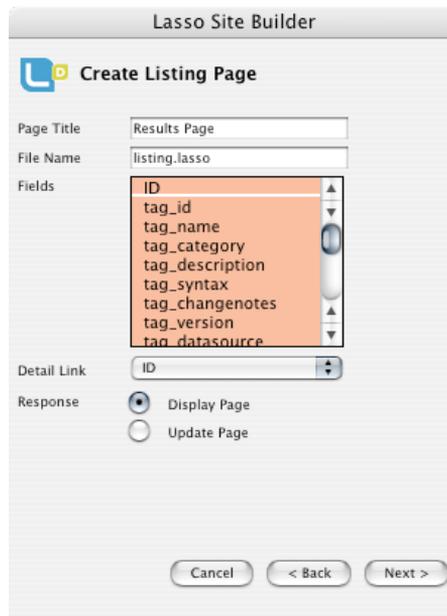
- 21 In the Detail Link pull-down menu, select the name of the field that you want users to be able to click on in the listing page to view the details for that record.

One field from each line in the results listing will link to either an update page which allows the client to update the data in the database or a display page that provides more information from the particular record in the database.

- 22 Finally, you need to decide whether the link should link to an update page or a display page.

Refer to the following *Display Detail Page* section to complete this procedure with a detail-only page. Otherwise, refer to the following *Update Detail Page* section to complete this procedure with an update page.

Figure 6: Lasso Site Builder Create Listing Page

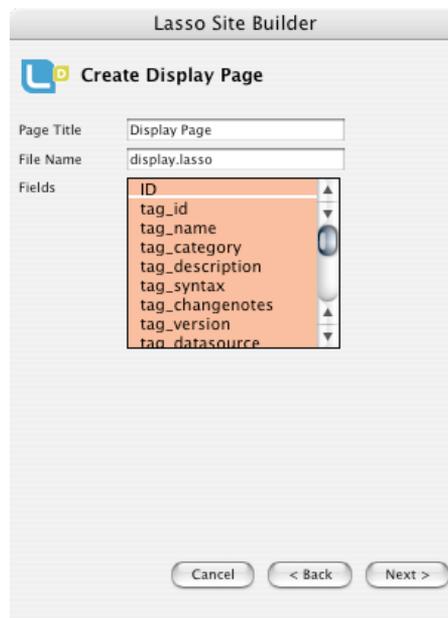


Display Detail Page

This section describes creating a record detail page that allows the record to be viewed only, and should follow the steps in the *Creating a Search Site* section.

To create a display detail page:

- 22 Next to Response, select the Display Page radio button. This will only allow the user to view the record.
- 23 Select the Next button to continue on to the Display Page screen.
- 24 Enter a title for the display page in the Page Title field (e.g. Display Page).
- 25 Enter a file name for the display page in the File Name field (e.g. Display.lasso). It is important that this page end with a .lasso extension.
- 26 In the Fields list, select all fields that you want the user to be able to view in the display page.

Figure 7: Lasso Site Builder Create Display Page

- 27 Select the Next button to view a summary of your Site Builder selections.
- 28 Select the Finish button to have Lasso Studio create the site. The process of creating the site can take a moment. You will see each page created and then saved to the destination folder.

The Site Builder has created three pages: Search.lasso, Listing.lasso, and Display.lasso. For an overview of the contents and structure of each page created, see the *Site Builder Pages* section of this chapter. For instructions on how to test your site, see the *Testing Your Site* section.

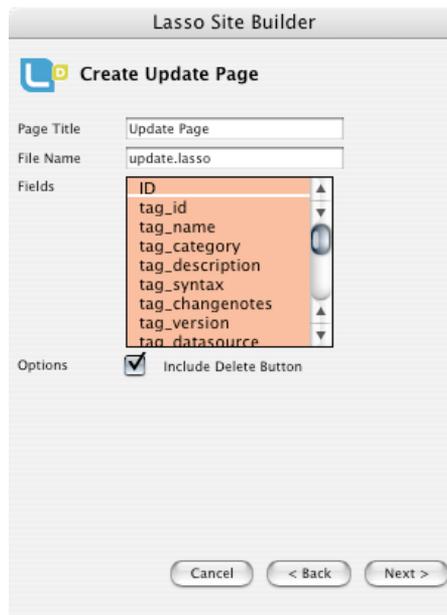
Update Detail Page

This section describes creating a record detail page that allows the record to be updated, and should follow the steps in the *Creating a Search Site* section.

To create an update detail page:

- 22 Next to Response, select the Update Page radio button. This will allow the user to update the record when viewing record detail.
- 23 Select the Next button to continue on to the Update Page screen.
- 24 Enter a title for the update page in the Page Title field (e.g. Update Page).
- 25 Enter a file name for the update page in the File Name field (e.g. Update.lasso). It is important that this page end with a .lasso extension.
- 26 In the Fields list, select all fields that you want the user to be able to update in the update page.

Figure 8: Lasso Site Builder Create Update Page



Key Field Note: Most data sources require that the key field remain constant during an update operation. In these cases, the key field should not be included among the fields you wish the end-user to update (e.g. the ID field in MySQL).

- 27 Next to Options, check the Include Delete Button check box. This will allow the user to delete the record.
A delete by a client cannot be undone, so make sure you want to allow users to delete records from your database before checking this box.
- 28 Select the Next button to continue on to the Update Response Page screen.
- 29 Enter a title for the update response page in the Page Title field (e.g. Update Response Page).
- 30 Enter a file name for the update response page in the File Name field (e.g. UpdateResponse.lasso). It is important that this page end with a .lasso extension.
- 31 The text area allows you to enter a message which will be displayed to the user upon the successful update or deletion of the selected record from the database.
- 32 Select the Next button to view a summary of your Site Builder selections.
- 33 Select the Finish button to have Lasso Studio create the site. The process of creating the site can take a moment. You will see each page created and then saved to the destination folder.

The Site Builder has created four pages: Search.lasso, Listing.lasso, Update.lasso, and UpdateResponse.lasso. For an overview of the contents and structure of each page created, see the *Site Builder Pages* section of this chapter. For instructions on how to test your site, see the *Testing Your Site* section.

Add Site Creation

This section describes how to use the Site Builder to create a set of pages to add records to your database, and then view or update the records once they are added.

Figure 9: Add Site



Creating an Add Site

Every add site contains an add form followed by either a display page with feedback that the item has been added to the database or an update page which allows the client to modify the added item. The update page can

have an optional delete button which takes the user to a feedback page when the item is successfully deleted from the database. Finally, every site has an error page which is returned to the client in case of any errors.

To create a basic add site:

- 1 From the Lasso Studio menu, select Lasso Site Builder.... This launches the Lasso Site Builder screen.
- 2 For the Site Folder field, select the folder where you would like your site to be stored, such as in an empty folder inside the root of your Web server. For instance, create a folder called MySite at the root level of your Web server so pages within it can be loaded with the following URL:
`http://127.0.0.1/MySite/`

Warning: Several individual documents will be created in the destination you select. It is important to choose an empty folder for the destination to avoid any files being over written.

- 3 Leave the Template field blank to have the Site Builder create files based on the default template. Otherwise, it is possible to use an HTML template file to which the code for each page created is inserted and named appropriately. For more information on using templates, see the *Site Builder Templates* section of this chapter.
- 4 Select the Next button to go on to the next screen.
- 5 For the site type, select the Add Page with Update or Display radio button.
- 6 Select the Next button to go on to the next screen.
- 7 In the Database pull-down menu, select the name of the database you wish to use for the site. The default database shown will be the one selected in the Database Selector.
- 8 In the Table pull-down menu, select the name of the table (or FileMaker layout) you wish to use.
- 9 In the Key Field pull-down menu, select the name of the key field for the table. This is typically the ID field for MySQL databases. The key field must contain a unique value for every record in the database.

Figure 9: Lasso Site Builder Database Selector

The screenshot shows a dialog box titled "Lasso Site Builder" with a subtitle "Lasso Database Selector". It features four pull-down menus for configuration:

- Database:** LDML8_Reference
- Table:** Tags
- Key Field:** ID
- Value Lists:** None

At the bottom of the dialog are three buttons: "Cancel", "< Back", and "Next >".

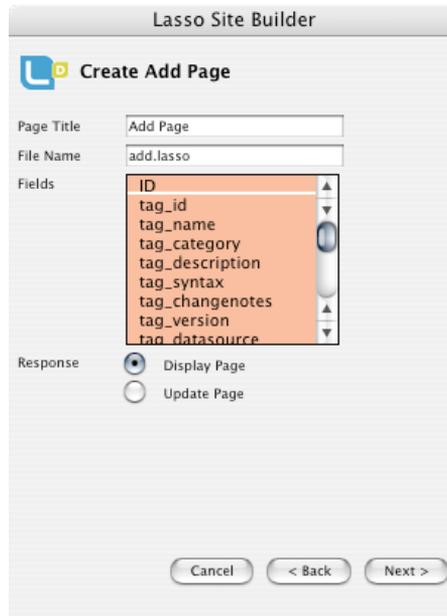
FileMaker Note: An Internal Record ID option appears for Filemaker databases, which represents Filemaker’s internal record ID field. Filemaker databases will use this regardless of what field is selected.

- 10** In the Value Lists pull-down menu, select whether you would like fields with value lists in your database to use dynamic or static pull-down menus, radio buttons, or check boxes in the search and update forms. Dynamic means that the page will populate the HTML selection items from the database (via an [Inline] Lasso Object), whereas Static means the HTML selection values will be hard-coded in the page.

MySQL Note: Value list fields in MySQL include ENUM and SET types.

- 11** Select the Next button to go on to the next screen to set options for the add page.

Figure 11: Lasso Site Builder Create Add Page



- 12 Enter a title for the add page in the Page Title field (e.g. Add Page). This page contains the form that allows users to input field values for a record they wish to add to a database.
- 13 Enter a file name for the add page in the File Name field (e.g. Add.lasso). It is important that this page end with a .lasso extension.
- 14 In the Fields list, select all fields that you want to appear on the add page.

Key Field Note: Most data sources auto-generate a key field value during an add operation. In these cases, the key field should not be included among the fields you wish the end-user to add (e.g. the ID field in MySQL).

- 15 Finally, you need to decide whether the link should link to an update page or a display page.
Refer to the following *Display Detail Page* section to complete this procedure with a detail-only page. Otherwise, refer to the following *Update Detail Page* section to complete this procedure with an update page.

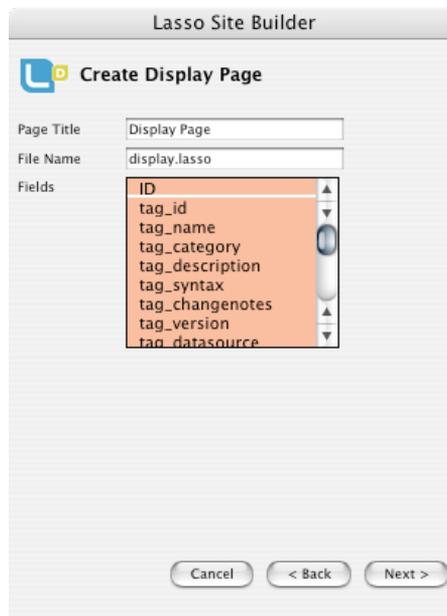
Display Detail Page

This section describes creating a record detail page that allows the record to be viewed only, and should follow the steps in the *Creating an Add Site* section.

To create a display detail page:

- 15 Next to Response, select the Display Page radio button. This will allow the user to view the record they just added, and also contains the [Inline] Lasso Object that adds the record based on the values submitted in the form.
- 16 Select the Next button to continue on to the Add Response Page screen.
- 17 Enter a title for the add response page in the Page Title field (e.g. Add Response Page).
- 18 Enter a file name for the add response page in the File Name field (e.g. AddResponse.lasso). It is important that this page end with a .lasso extension.
- 19 In the Fields list, select all fields that you want the user to be able to view in the display page.

Figure 12: Lasso Site Builder Create Display Page



- 20 Select the Next button to view a summary of your Site Builder selections.

- 21 Select the Finish button to have Lasso Studio create the site. The process of creating the site can take a moment. You will see each page created and then saved to the destination folder.

The Site Builder has created two pages: `Add.lasso` and `AddResponse.lasso`. For an overview of the contents and structure of each page created, see the *Site Builder Pages* section of this chapter. For instructions on how to test your site, see the *Testing Your Site* section.

Update Detail Page

This section describes creating a record detail page that allows the record to be updated, and should follow the steps in the *Creating an Add Site* section.

To create an update detail page:

- 15 Next to Response, select the Update Page radio button. This will allow the user to update the record when viewing record detail.
- 16 Select the Next button to continue on to the Update Page screen.
- 17 Enter a title for the update page in the Page Title field (e.g. Update Page).
- 18 Enter a file name for the update page in the File Name field (e.g. Update.lasso). It is important that this page end with a .lasso extension.
- 19 In the Fields list, select all fields that you want the user to be able to update in the update page.

Figure 13: Lasso Site Builder Create Update Page

Key Field Note: Most data sources require that the key field remain constant during an update operation. In these cases, the key field should not be included among the fields you wish the end-user to update (e.g. the ID field in MySQL).

- 20 Next to Options, check the Include Delete Button check box. This will allow the user to delete the record.

A delete by a client cannot be undone, so be sure you want to allow users to delete records from your database before checking this box.

- 21 Select the Next button to continue on to the Update Response Page screen. The update response page contains the [Inline] Lasso Object that updates a record in the database based on the values submitted in the form, and displays the newly-updated record.
- 22 Enter a title for the update response page in the Page Title field (e.g. Update Response Page).
- 23 Enter a file name for the update response page in the File Name field (e.g. UpdateResponse.lasso). It is important that this page end with a .lasso extension.

- 24 The text area allows you to enter a message which will be displayed to the user upon the successful update or deletion of the selected record from the database.
- 25 Select the Next button to view a summary of your Site Builder selections.
- 26 Select the Finish button to have Lasso Studio create the site. The process of creating the site can take a moment. You will see each page created and then saved to the destination folder.

The Site Builder has created three pages: `Add.lasso`, `Update.lasso`, and `UpdateResponse.lasso`. For an overview of the contents and structure of each page created, see the *Site Builder Pages* section of this chapter. For instructions on how to test your site, see the *Testing Your Site* section.

Site Builder Pages

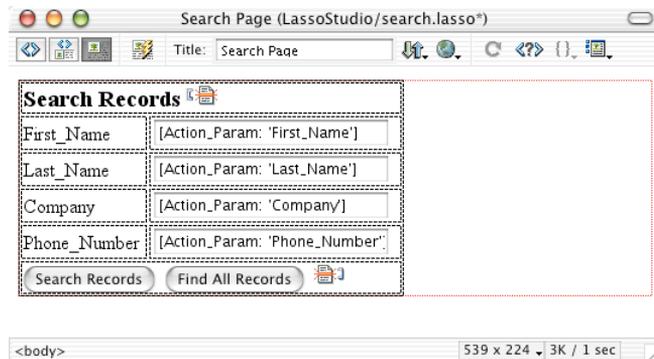
The Site Builder has the ability to create seven unique pages depending on the site options selected, which include `Search.lasso`, `Listing.lasso`, `Display.lasso`, `Add.lasso`, `Update.lasso`, `AddResponse.lasso` and `UpdateResponse.lasso`. This section describes the contents of each of these pages.

Note: The code shown throughout this section is in colon syntax. If the global syntax preference in the Lasso Studio Panel is set to parentheses syntax then the code generated will be equivalent to what is shown here except for the minor differences between the two syntaxes.

Search.Lasso

This page presents a form to the user with text inputs for each field in the database. When the user selects the Search button a search is performed in the database and the results are returned on the listing page.

Figure 14: Search.lasso



The Search.lasso page is for the most part a straight-forward HTML form, but contains several elements that allow it to communicate with an [Inline] tag on the response page. The response page is designated in the Action attribute of the <Form> tag.

```
<form action="listing.lasso" method="post">
```

In this case, when the user submits the form, the user will be taken to the listing.lasso page. The listing.lasso page will contain an [Inline] tag which processes and displays results based on the information submitted in the form.

If the Include Field Operators option was selected in the Site Builder, each field input will be preceded by an -Operator command tag with selectable values in a pull-down menu. For more information on operators, see the *Searching and Displaying Records* chapter in the Lasso 8 Language Guide.

```
<select name="-Operator">
  <option value="bw" selected>begins with</option>
  <option value="eq">equals</option>
  <option value="cn">contains</option>
  <option value="ew">ends with</option>
</select>
<input type="text" name="Last_Name" value="">
```

If a Dynamic value list option was selected in the Site Builder, then the HTML form will be surrounded by [Inline] ... [/Inline] tags that perform a -Show operation which extracts dynamic value list information out of the database. For more information on this, see the *Searching and Displaying Records* chapter in the Lasso 8 Language Guide.

Finally, the button names contain named -Token tags which will inform the response page what action needs to be performed. Each button will

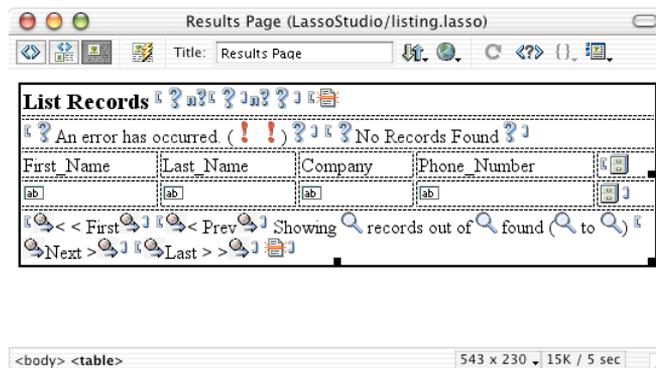
submit a different token value, which is specified in the Value attribute of the <Input> tag.

```
<input type="submit" name="-Token.Builder_Action" value="Search">
<input type="submit" name="-Token.Builder_Action" value="Find All">
```

Listing.Lasso

This is the response page to the search form. The response is a listing of all the records in the database in a columnar display.

Figure 15: Listing.lasso



The icons at the top of the page contain [If] Lasso Objects for determining what action needs to be performed based on the token value submitted with a form button, and contain [Inline] Lasso Objects for performing a -Search or -FindAll database action depending on the results of the [If] statement. For more information on [If] tags, see the *Conditional Logic* chapter in the Lasso 8 Language Guide.

```
[If: (Token_Value: 'Action') >> 'FindAll']
  [Variable: 'Action' !='-FindAll']
[Else]
  [Variable: 'Action' !='-Search']
[/If]
[Inline: -InlineName='FormBuilder', (Action_Params),
  -Database='Example', -Table='Example',
  -KeyField='ID', -MaxRecords=10,
  $Action]
...
[/Inline]
```

In between the [Inline] ... [/Inline] tags, the row with the field icons is repeated for each record in the database. The code underlying this portion of the file looks like this.

```
[Records]
<tr>
  <td><a href="display.lasso?-KeyField=[KeyField_Value]&-Token=[Token_Value:
'Action']">[Field:'Company']</a></td>
  <td>[Field:'Address']</td>
  <td>[Field:'City']</td>
  <td>[Field:'State']</td>
  <td>[Field:'Zip']</td>
</tr>
[/Records]
```

Everything between the container tags [Records] ... [/Records] will be repeated for each record in the found set. The field that is linked to the record detail page contains -Keyfield and -Token command tags in the URL that pass necessary information to the display page so that the correct record is displayed.

The final row of the table contains automatically generated links to the next group of found records and the previous group of found records.

```
[Link_FirstGroup: -NoDatabase, -NoTable, -NoKeyField, -NoAction] First [/Link_
FirstGroup]
[Link_PrevGroup: -NoDatabase, -NoTable, -NoKeyField, -NoAction] Previous [/Link_
PrevGroup]

Showing [Show_Count] records out of [Found_Count] found ([Shown_First] to
[Shown_Last])

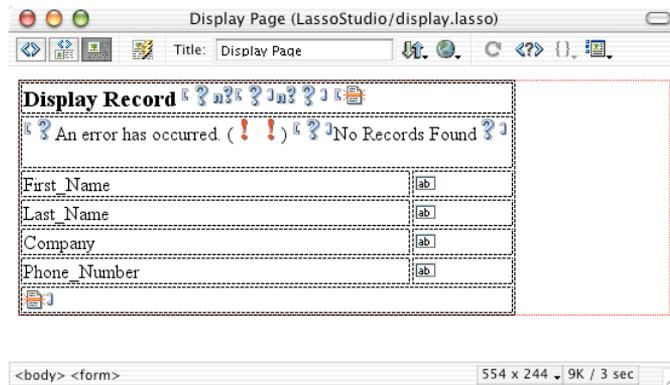
[Link_NextGroup: -NoDatabase, -NoTable, -NoKeyField, -NoAction] Next [/Link_
NextGroup]
[Link_LastGroup: -NoDatabase, -NoTable, -NoKeyField, -NoAction] Last [/Link_
LastGroup]
```

By default, 10 records will be shown on the first screen due to the -MaxRecords=10 parameter in the [Inline] tag. These links will be displayed only if more than 10 records were found that matched the search criteria.

Display.Lasso

When the user selects one of the detail links from the listing page, Lasso serves this page to provide more detail about the particular record they selected.

Figure 16: Display.lasso



The icons at the top of the page contain an [Inline] Lasso Object which performs a -Search or -FindAll action on the database in order to display record data. The [Inline] tag will display record data based on information submitted by a previous page. The [Action_Params] tag transfers all information submitted from a previous page.

```
[If: (Token_Value: 'Action') >> 'FindAll']
  [Variable: 'Action' = '-FindAll']
[Else]
  [Variable: 'Action' = '-Search']
[/If]
[Inline: -InlineName='FormBuilder', (Action_Params),
  -Database='Example', -Table='Example',
  -KeyField='ID', -MaxRecords=1,
  $Action]
...
[/Inline]
```

An [If] statement is used to display an error message if an error occurs when displaying the record, or if there are no records found. The [Error_CurrentError] tag displays the current Lasso error message. For more information on error tags, see the *Error Control* chapter in the Lasso 8 Language Guide.

```
[If: ((Error_CurrentError) != (Error_NoError))]
  An error has occurred.
  [Error_CurrentError: -ErrorCode] [Error_CurrentError]
[Else: (Found_Count) == 0] No Records Found [/If]
```

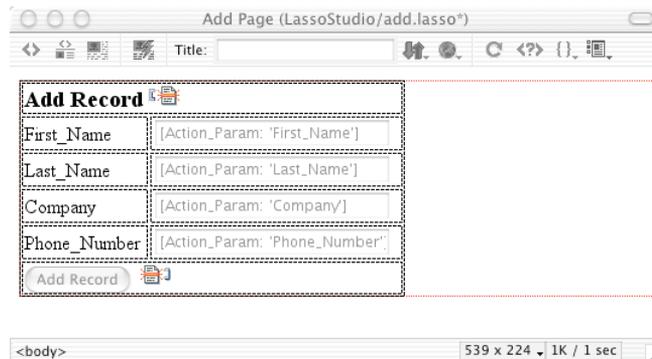
Record data is displayed via [Field] Lasso Objects. All [Field] Lasso Objects are contained within the [Inline] ... [/Inline] tags, which populates the [Field] objects with the record data returned from the database action.

[Field: 'Company']

Add.Lasso

This page presents a form to the user with text inputs for each field in the database. When the user selects the Add button, the form information is submitted to a response page that adds a record to the database with the values the user specified.

Figure 17: Add.lasso



The Add.lasso page is for the most part a straight-forward HTML form, but contains several elements that allow it to communicate with an [Inline] tag on the response page. The response page is designated in the Action attribute of the <Form> tag.

```
<form action="addressresponse.lasso" method="post">
```

In this case, when the user submits the form, the user will be taken to the addressresponse.lasso page. The addressresponse.lasso page will contain an [Inline] tag which adds the record and displays results based on the information submitted in the form.

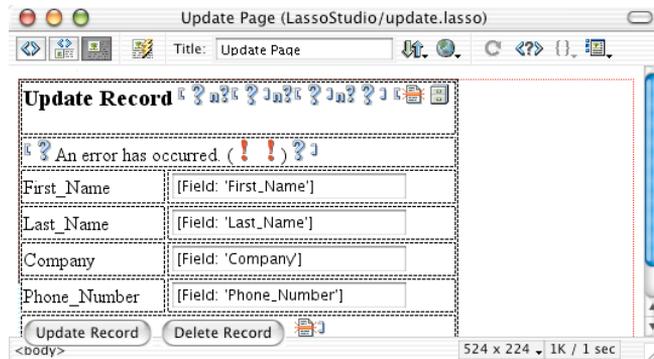
Finally, the Name attribute of the Add button contains a named -Token tag, which informs a conditional [If] tag in the response page what action needs to be performed. The token value is specified in the Value attribute of the <Input> tag.

```
<input type="submit" name="-Token.Builder_Action" value="Add Record">
```

Update.Lasso

This page is a form with text inputs for each field in the database. When the page is loaded, values from the appropriate fields in the database are retrieved and automatically inserted into the text input fields. These values can be edited and when the user selects the Update button the values will be updated in the fields for the current record in the database.

Figure 18: Update.lasso



The icons at the top of the page contain [If] and [Inline] Lasso Objects which perform a -Search or -Add action on the database (depending on the previous page) to populate the text fields with record data. The [Inline] tags will populate the fields with record data based on information submitted by a previous page.

```
[If: (Token_Value: 'Action') >> 'Add']
  [Variable: 'Action' = '-Add']
[Else]
  [Variable: 'Action' = '-Search']
[/If]
[Inline: -InlineName='FormBuilder', (Action_Params),
  -Database='Example', -Table='Example',
  -KeyField='ID', -MaxRecords=1,
  $Action]
...
[/Inline]
```

Another [If] statement is used to display an error message if an error occurs when loading or using the page. The [Error_CurrentError] tag displays the current Lasso error message. For more information on error tags, see the *Error Control* chapter in the Lasso 8 Language Guide.

```
[If: ((Error_CurrentError) != (Error_NoError))]
  An error has occurred.
  [Error_CurrentError: -ErrorCode] [Error_CurrentError]
[/If]
```

The row with the field inputs is repeated for each record returned. The values from the fields in the current database record are placed in the text inputs. In the example below, the substitution tag [Field: 'Company'] places the value for the field Company from the record which was just added to the database into the text input so the user can make changes.

```
<input type="text" name="Company" value="[Field: 'Company']">
```

Finally, the button names contain named -Token tags which will inform the response page what action needs to be performed. Each button will submit a different token value, which is specified in the Value attribute of the <Input> tag.

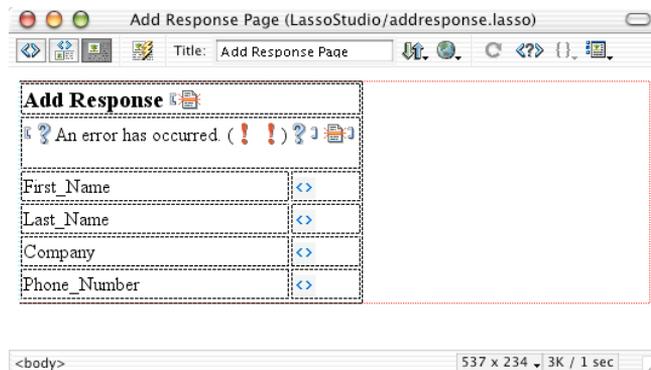
```
<input type="submit" name="-Token.Action" value="Update">
<input type="submit" name="-Token.Action" value="Delete">
```

The Update.lasso page has an option to delete the record which is currently being shown. If the Delete button is selected, then the record will be deleted in the UpdateResponse.lasso page instead of updated.

AddResponse.Lasso

The AddResponse.lasso page is similar to the Display.lasso page, but performs an -Add action instead of a -Search action in the opening [Inline] tag so that a record is added based on information submitted in a preceding form.

Figure 19: AddResponse.lasso



The icons at the top of the page contain an [Inline] Lasso Objects which performs an -Add action on the database. The [Action_Params] tag transfers all information submitted from a previous page.

```
[Inline: (Action_Params), -Database='Example', -Table='Example', -KeyField='ID',
-Add]
...
[/Inline]
```

An [If] statement is used to display an error message if an error occurs when adding the record. The [Error_CurrentError] tag displays the current Lasso error message. For more information on error tags, see the *Error Control* chapter in the Lasso 8 Language Guide.

```
[If: ((Error_CurrentError) != (Error_NoError))]
  An error has occurred.
  [Error_CurrentError: -ErrorCode] [Error_CurrentError]
[/If]
```

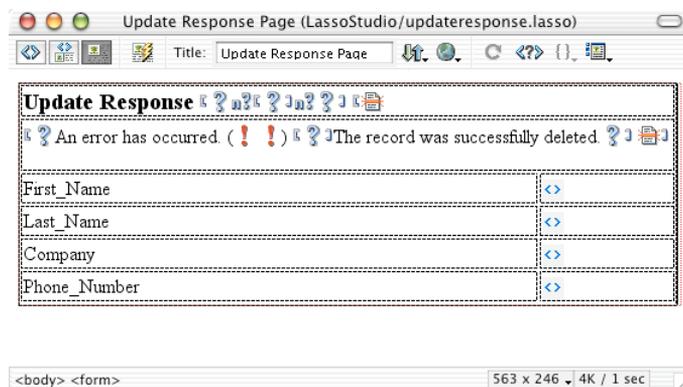
Record data is displayed via [Action_Param] Lasso Objects, which display the values submitted in the preceding form that were added to the database.

```
[Action_Param: 'Company']
```

UpdateResponse.Lasso

The UpdateResponse.lasso page is similar to the AddResponse.lasso page, but performs either an -Update or -Delete action in the opening [Inline] tag depending on the button that was selected in a preceding update form.

Figure 20: UpdateResponse.lasso



The icons at the top of the page contain [If] and [Inline] Lasso Objects which perform an -Update or -Delete action on the database depending on the button selected in a preceding Update.lasso page.

```
[If: (Token_Value: 'Action') >> 'Delete']
  [Variable: 'Builder_Action'=-Delete]
[Else]
  [Variable: 'Action'=-Update]
[/If]
  [Inline: (Action_Params), -Database='Example', -Table='Example', -KeyField='ID',
  $Action]
```

An [If] statement is used to display an error message if an error occurs when updating or deleting the record, or a confirmation message if the record is deleted. The [Error_CurrentError] tag displays the current Lasso error message.

```
[If: ((Error_CurrentError) != (Error_NoError))]
  An error has occurred.
  [Error_CurrentError: -ErrorCode] [Error_CurrentError]
[Else: (Lasso_CurrentAction) == 'Delete']
  The record has been succesfully deleted.
[/If]
```

Record data is displayed via [Action_Param] Lasso Objects, which display the values submitted in the preceding form that were updated in the database.

```
[Action_Param: 'Company']
```

Testing Your Site

The site builder will create a series of pages in the selected site folder. In order to test the site, you need to copy this folder into your Web serving folder or upload it to your Web server.

Depending on whether you created a search site or an add site the instructions for testing the site are different. In all cases, refer to the *Using Lasso Studio* chapter *Testing Solutions* section for instructions on how to set up and test your site.

Search Site

Test the search site by loading the search page in a Web browser. The URL of this page should be the following:

```
http://www.example.com/MySite/search.lasso
```

Replace www.example.com with the static IP address or domain name of your Web server running Lasso Professional. Replace MySite with the path to the site folder. Finally, search.lasso is the default name of the search page in

the site folder. You should enter the actual name of the file if you selected another name.

Try performing a Search or a Find All action (if available) and ensure that the other pages of the created site work as expected.

If you receive no results or an error message when attempting to search or update a record in the database, then your Lasso Site Administration settings for your database might not be properly configured. See the *Using Lasso Studio* chapter *Testing Solutions* section for more information.

Add Site

Test the add site by loading the add page in a Web browser. The URL of this page should be the following:

```
http://www.example.com/MySite/add.lasso
```

Replace `www.example.com` with the static IP address or domain name of your Web server running Lasso Professional 8. Replace `MySite` with the path to the site folder. Finally, `add.lasso` is the default name of the add page in the site folder. You should enter the actual name of the file if you selected another name.

Try adding a record to the database and ensure that the other pages of the created site work as expected.

If you receive an error message when attempting to add or update a record, then your Lasso Site Administration settings for your database might not be properly configured. See the *Using Lasso Studio* chapter *Testing Solutions* section for more information.

Site Builder Templates

The Site Builder features the option to use a template file as the basis for each of the files it creates. A Site Builder template file is any HTML file which has a special marker tag inserted that instructs Lasso Studio where to place the Site Builder content.

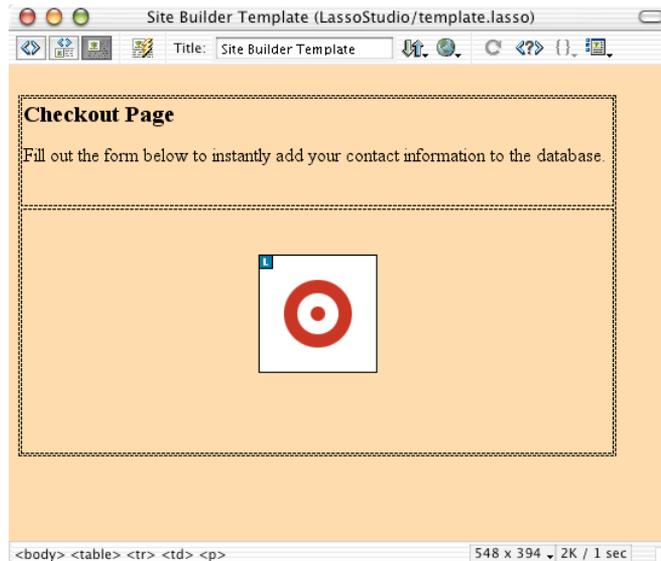
The marker tag is inserted by dragging and dropping the Site Builder Target object from the Lasso Form Objects section of the Insert panel. The marker tag is represented by a target icon in the Design view of Dreamweaver, and inserts the following in the source code of the page.

```
<ls4gl_builder_target></ls4gl_builder_target>
```

Upon running Site Builder, the marker tags and any content within them will be replaced by the code generated by the Site Builder. The template

file itself is not modified, but is used as the basis for each new file the Site Builder creates.

Figure 21: Site Builder Template



To insert a Site Builder target into an HTML template:

- 1 In Dreamweaver, open the template file that you wish to use for your site.
- 2 From the Lasso Form Objects section of the Insert panel, drag and drop the Site Builder Target object to the location in your page where you want Site Builder to insert the Lasso code.
- 3 Save the file.

Once a template file has been saved with a Site Builder Target object, it may then be used with the Site Builder.

To use the Site Builder with a template file:

- 1 From the Lasso Studio menu, select Lasso Site Builder.... This launches the Lasso Site Builder screen.
- 2 For the Site Folder field, select the folder where you would like your site to be stored.
- 3 In the Template field, select the HTML template file you wish to use. The template file should contain a Site Builder Target object, as described previously.

4 Select the Next button and continue using the Site Builder as described in the *Search Site Creation* or *Add Site Creation* sections of this chapter.

When the Site Builder finishes, each of the pages generated will contain the same code elements of the template file, but will contain different Lasso code inserted where the Site Builder Target object was.

Once a page has been created from a template by the Site Builder, further customization must be performed using the tools of Dreamweaver or Lasso Studio. You cannot use the Site Builder to update the code within a page which was previously created using the Site Builder.

8

Chapter 8

Using Objects and Tags

This chapter describes using the Lasso tag language and Lasso Objects in Lasso Studio.

- *Overview* provides conceptual information about using the Lasso tag language and Lasso Objects.
- *Lasso Data Access Objects* describes using the Lasso Data Access objects in the Insert panel in Dreamweaver.
- *Lasso Programming Objects* describes using the Lasso Programming objects in the Insert panel in Dreamweaver.
- *Lasso Form Objects* describes using the Lasso Form objects in the Insert panel in Dreamweaver.
- *Using Lasso Objects* provides an example of using Lasso Objects in a Web page.
- *Lasso Display Preferences* describes changing how Lasso Objects are displayed in the Design view of Dreamweaver.

Overview

This section describes the basics of using Lasso' tag language which is inserted into documents by Lasso Studio and interpreted by Lasso Professional when served.

When a document containing Lasso tags is opened or created in Lasso Studio for Dreamweaver, the Lasso tags appear differently in the Design and Code views in Dreamweaver. These differences are described below.

- **Design View** – The Design view of the main window presents a visual preview of how the page is going to appear in a Web browser. In the

Design view, Lasso tags are represented by objects (icons) which can be found in the Lasso Data Access, Lasso Programming, and Lasso Form sections of the Dreamweaver Insert panel. Display preferences can be set for Lasso Objects in Lasso Studio, which is described in the *Lasso Display Preferences* section of this chapter.

Note: Certain Lasso Objects such as Field objects will be displayed differently if the Data Preview feature is enabled. See the *Using Database Snapshots* chapter for more information.

- **Code View** – The Code view in Dreamweaver shows you the underlying code of the page. Just as you can edit the raw HTML in this window, you can edit the raw Lasso code as well. Lasso can appear in several different forms of syntax, which are described in the following *Lasso Tag Types* section.
- **Split Design/Code View** – The Split Design/Code view in Dreamweaver shows both the Design view and Code view in split screens, allowing each to be viewed and edited simultaneously.
- **Live Data View** – Lasso Studio operates in Dreamweaver’s Live Data view in the same fashion as the Design view, but does not interact with Dreamweaver’s Live Data view features. Lasso Studio comes with its own Data Preview feature, which is controlled via the Lasso Studio panel. For more information, see the *Using Database Snapshots* chapter

Lasso Objects Overview

Lasso Studio adds three sections to the Insert panel in Dreamweaver. These sections contain Lasso Objects, which are icons that represent one or more Lasso tags.

Figure 1: Lasso Objects



Lasso Objects operate in a similar fashion to regular Dreamweaver objects, and can be dragged and dropped into a document in order to insert Lasso code. Each Lasso Object in a panel represents a category of Lasso tags, from which one tag may be selected for use in a page using the Lasso Inspector panel.

Inserting Lasso Objects in the Design view in Dreamweaver is the primary focus of this chapter. Each of the Lasso Objects are described in the *Lasso Data Access*, *Lasso Programming*, and *Lasso Form* sections of this chapter, and general Lasso Object usage instructions are provided in the

Using Lasso Objects section. Lasso Objects are used in concert with the Lasso Inspector panel and Lasso Tag Editor, which are described in the subsequent chapters in this guide.

In order to successfully use Lasso Objects in Lasso Studio, it is important to understand a few basic concepts about the Lasso language, such as the Lasso Tag Types. An overview of the Lasso Tag Types and how they are used in Lasso Studio is provided below.

Lasso Tag Types

Among individual Lasso tags, there are several different tag and syntax types which are used in the source code of a document. Understanding the basic Lasso Tag Types is important before beginning to program in Lasso Studio. Lasso tag and syntax types include substitution tags, process tags, container tags, member tags, Lasso expressions, and LassoScript tags, which are all described individually in this section. Each Lasso tag available in each Lasso Object in Lasso Studio is one of the following tag types.

Note: Please consult the Lasso 8 Language Guide and Lasso Reference for details about each of the Lasso tags and tag types described in this chapter.

Substitution Tags

Substitution tags return a value which is substituted in place of the tag within the Lasso page being served to a browser. Most of the tags in Lasso are substitution tags. Substitution tags are used to return field values from a database query, return the results of calculations, or to display information about the state of Lasso Professional 8 and the current page request.

Substitution tags will be replaced by its value when the page is served to a browser. For example, the following [Field] tags will be replaced by the company's information from the database.

```
[Field: 'Company_Name'] → OmniPilot
[Field: 'Company_URL'] → http://www.omnipilot.com
```

In Lasso Studio, substitution tags exist as icons in the Lasso Data Access and Lasso Programming sections of the Insert panel. The example below shows the [Field] Lasso Object, which outputs data from a field in a database.



Substitution tags may be dragged and dropped from the panel into the current page being developed, and are represented by a single icon in the Design view of the page. Properties for substitution tags are viewed and

modified in the Lasso Inspector panel after selecting the substitution Lasso Object in a page.

Lasso Studio inserts substitution tags in source code using square bracket syntax. The square brackets serve to distinguish Lasso tags from markup tags in the Lasso page, such as HTML and XML tags which are delimited by angle brackets. Lasso tags can be used on their own, or within HTML or XML tags. Lasso will not disturb the markup tags, but will replace the square bracketed tag by its value in place when the Lasso page is served. For example, the [Field] tag can be used to specify the value of the src attribute for an HTML tag:

```
 → 
```

Process Tags

Process tags are similar to substitution tags, but perform an action which does not return a value. This means that the tag will perform an action, but does not appear visually in the Web page when it is served. Process tags can be used to alter the HTTP header of an HTML file being served, to store values, to schedule tasks for later execution, to send email messages, and more.

For example, the following [Session] tag begins a session that stores a cookie in a client's Web browser and logs the session information to the Site database in Lasso Professional 8, but returns no value in the Web page being served.

```
[Session: -Name='My_Session', -Expires='60', -UseCookie]
```

In Lasso Studio, process tags exist as icons in the Lasso Data Access and Lasso Programming sections of the Insert panel. The example below shows the [Email] Lasso Object, which is used to send email messages but does not output anything to the page.



Process tags may be dragged and dropped from the panel into the current page being developed, and are represented by a single icon in the Design view of the page. Properties for process tags are viewed and modified in the Lasso Inspector panel after selecting the process Lasso Object in a page.

Container Tags

Container tags are a matching pair of tags which enclose a portion of a Lasso page or LassoScript and either alter the enclosed contents or change the behavior of tags within the enclosed contents. The opening tag uses the same syntax as a substitution or process tag. The closing tag has the same

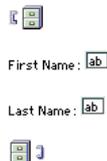
name as the opening tag, but the closing tag is specified with a leading forward slash. This is similar to how HTML markup tags are paired.

Commonly used container tags in Lasso include the [Inline] ... [/Inline] tags and [Records] ... [/Records] tags. For example, the following [Inline] tag specifies the parameters of a search on the Contacts database to be performed, and contains a pair of [Records] ... [/Records] tags. The [Records] ... [/Records] tags loop a pair of [Field] substitution tags for each record found by the search, displaying field data for each record.

```
[Inline: -FindAll, -Database='Contacts', -Table='People', -MaxRecords='All']
  [Records]
    [Field: 'First_Name'] <br> [Field: 'Last_Name']
  [/Records]
[/Inline]
```

In Lasso Studio, container tags exist as single icons in the Lasso Data Access and Lasso Programming sections of the Insert panel, and when inserted in a page they become multiple icons that represent opening and closing tags. These opening and closing tags are placed before and after other tags and objects to which they apply. For example, the image below shows a pair of [Records] ... [/Records] tags around two fields from a database.

Figure 2: Records Container Tag Objects



Note that the icons of a matched pair of container tags have a pair of square brackets around them so you can see what portion of the page they surround. Properties for container tags are viewed and modified in the Lasso Inspector panel after selecting the opening container Lasso Object in a page.

Note: When modifying the properties of container tags, select only the opening tag. Lasso Studio automatically inserts or removes the closing tag if the opening tag is changed using the Lasso Inspector panel.

Command Tags

Command tags specify an action and the parameters of that action. The elements that make up an action can be embedded within HTML form inputs, URLs, or within the [Inline] tag. Command tags are usually used to specify a database action and its parameters, but can be used to specify

other actions as well. Command tag names start with a hyphen (e.g. -Search), which helps to differentiate them from other tags.

The command tags are commonly-used to specify the database action which an [Inline] tag is to perform. The contents of the [Inline] ... [/Inline] tags will be affected by the results of this action. In the following example, the command tags -Database, -Table, -KeyField are set to pre-defined values. The -FindAll command tag specifies what action the [Inline] will perform.

```
[Inline: -FindAll, -Database='Contacts', -Table='People', -KeyField='ID']
  [Records]
    [Field: 'First_Name'] <br> [Field: 'Last_Name']
  [/Records]
[/Inline]
```

In Lasso Studio, command tags exist as icons in the Lasso Form section of the Insert panel. The example below shows the Action Lasso Object, which is used to specify a database action command tag (e.g. -Search, -Add, -Update, -Delete).



Command tags may be dragged and dropped from the panel into the current page being developed, and are represented by a single icon in the Design view of the page. Properties for command tags are viewed and modified in the Lasso Inspector panel after selecting the command Lasso Object in a page.

Member Tags

Member tags modify or return data from a value of a specific data type (Integer, String, Date, etc.). Each data type in Lasso has different member tags. Member tags can either be used in the fashion of process tags to alter a value or they can be used in the fashion of substitution tags to return a value.

Member tags differ from substitution and process tags in that they must be called using the member symbol -> and a value from the appropriate data type. For example, the [String->Get] member tag requires a value of type string. Member tags are always written in this fashion in the documentation: the data type followed by the member symbol and the specific tag name. The following code fetches the third character of the specified string literal:

```
[Output: 'Text'-(Get: 3)] → x
```

In Lasso Studio, member tags may only be used by selecting a substitution tag object in a page and inserting member tag code in the Source window

of the Lasso Inspector panel. For more information on member tags and their applications, see the *Programming Fundamentals* chapter in the Lasso 8 Language Guide.

LassoScript

A LassoScript object allows multiple Lasso tags to be inserted within a single `<?LassoScript ... ?>` tag. Multiple substitution tags, process tags, container tags and member tags can all be used within a single `<?LassoScript ... ?>` tag. LassoScript uses a syntax type different from square-bracket syntax, and is described in the *LassoScript* chapter in the Lasso 8 Language Guide.



The entire LassoScript is replaced by the result of all the tags included in the LassoScript when the Lasso page is served to a client. For example, the following LassoScript will return the value of the included [Field] tag.

```
<?LassoScript
  Field: 'Image_URL';
?>
```

→ /Images/Portrait.gif

In Lasso Studio, LassoScript may be used by dragging and dropping the LassoScript Lasso Object from the Lasso Programming section of the Insert panel. LassoScript expressions must be manually entered in the Source window of the Lasso Inspector panel. For more information on LassoScript and its applications, see the *LassoScript* chapter in the Lasso 8 Language Guide.

Lasso Expressions

A Lasso Expression object provides a single pair of square brackets in which any Lasso string or math expression may be inserted. Lasso expressions provide an easy way to perform string and math operations by simply using values and symbols without using named Lasso tags.



The following examples perform a math operation and string concatenation operation, respectively.

```
[4 * 4 + 4 - 5]
```

→ 15

```
['Omni' + ' ' + 'Pilot']
```

→ OmniPilot

In Lasso Studio, Lasso expressions may be used by dragging and dropping the Lasso Expression Lasso Object from the Lasso Programming section of the Insert panel. Lasso expressions must be manually entered in the Source window of the Lasso Inspector panel. For more information on Lasso expressions, see the *Lasso Programming Fundamentals* chapter *Symbols* section in the Lasso 8 Language Guide.

Important: When Lasso Professional parses files, anything contained within a pair of square brackets is considered a Lasso tag or Lasso expression. To display left and right square brackets in HTML, use the `[` and `]` characters.

Lasso Data Access Objects

The Lasso Data Access objects are used to add database actions or return the results of a database action in a Lasso page. They can also be used to insert links to the next or previous groups of found records, and can be used to display information about the client's browser.

Figure 3: Lasso Data Access Objects



Each Lasso Data Access object is described below along with a list of the tags available for the object. Once the object has been inserted in a document, it is represented by an icon and can be modified using a related property inspector. The property inspectors for each object are described in the *Using Lasso Inspectors* chapter. The use of each of the Lasso tags is outside the scope of this manual. Please consult the Lasso Reference or the Lasso 8 Language Guide for more information about programming in Lasso.

Action Param

Used to retrieve and output field data submitted from a form on a preceding page. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

[Action_Param]

[Action_Params]

Client 

Tags which provide information about the client's browser. For more information, see the *HTTP/HTML Content and Controls* chapter and the *Wireless Devices* chapter in the Lasso 8 Language Guide.

[Client_Address]	[Client_Browser]
[Client_ContentLength]	[Client_ContentType]
[Client_CookieList]	[Client_Cookies]
[Client_FormMethod]	[Client_GetArgs]
[Client_GetParams]	[Client_Headers]
[Client_IP]	[Client_Password]
[Client_PostArgs]	[Client_PostParams]
[Client_Type]	[Client_Username]
[WAP_IsEnabled]	[WAP_MaxButtons]
[WAP_MaxColumns]	[WAP_MaxHorzPixels]
[WAP_MaxRows]	[WAP_MaxVertPixels]

Database 

Tags that return information about the database being used. Some tags can also be used to create MySQL tables and fields. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

[Database_ChangeField]	[Database_CreateField]
[Database_CreateTable]	[Database_Nameltem]
[Database_Names]	[Database_Name]
[Database_RealName]	[Database_RemoveField]
[Database_RemoveTable]	[Database_TableNameltem]
[Database_TableNames]	[FieldName]
[KeyField_Name]	[KeyField_Value]
[Layout_Name]	[Operator_LogicalValue]
[Op_LogicalValue]	[Checked]
[Option]	[Portal]
[RecordID_Value]	[Records]
[Repeating]	[RepeatingValueItem]
[Rows]	[Selected]
[Table_Name]	[Table_RealName]
[Value_ListItem]	[Value_List]

Email 

The [Email_Send] tag is used for automatically sending email messages. For more information, see the *Email* chapter in the Lasso 8 Language Guide.

[Email_Send]

Encoding

Tags for encoding or decoding strings. For more information, see the *Encoding* chapter in the Lasso 8 Language Guide.

[Decode_Base64]	[Decode_URL]
[Decode_UTF8]	[Encode_Base64]
[Encode_Breaks]	[Encode_HTML]
[Encode_ISOToMac]	[Encode_MacToISO]
[Encode_Set]	[Encode_Smart]
[Encode_SQL]	[Encode_StrictURL]
[Encode_URL]	[Encode_UTF8]
[Encode_XML]	

Field

A tag which displays values from a field in a database or data source. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

[Field]

Image

Tags which can manipulate image files. For more information, see the *Images and Multimedia* chapter in the Lasso 8 Language Guide, and also the *FileMaker Data Sources* chapter for information on using images stored in FileMaker Pro databases.

[Image]	[IMG]
[Image_URL]	

Include

Tags which include the contents of a file from the local hard drive or from a URL on another Web server. For more information, see the *Files and Logging* chapter in the Lasso 8 Language Guide.

[Include]	[Include_Raw]
[Include_URL]	[Library]

Inline

Container tag which allow you to perform Lasso actions. This is the core tag for database interaction in Lasso 7. Inserting an Inline Lasso Object also inserts Database Lasso Objects that represent the [Records] ... [Records] tags in between the [Inline] ... [/Inline] tags. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

[Inline]

Link 

Tags which allow you to link to related sets of found records, e.g. the next or previous group of found records. Most link tags are container tags, and insert an opening and closing tag. For more information, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.

[Link_CurrentAction]	[Link_CurrentActionURL]
[Link_Detail]	[Link_DetailURL]
[Link_FirstGroupURL]	[Link_FirstRecordURL]
[Link_FirstRecord]	[Link_LastGroup]
[Link_LastGroupURL]	[Link_LastRecord]
[Link_LastRecordURL]	[Link_NextGroup]
[Link_NextGroupURL]	[Link_NextRecord]
[Link_NextRecordURL]	[Link_PrevGroup]
[Link_PrevGroupURL]	[Link_PrevRecord]
[Link_PrevRecordURL]	[Referrer]
[Referrer_URL]	

Output 

Tags which output various values to a Web page. These tags also allow you to manipulate low-level HTML and HTTP characteristics. For more information, see the *HTTP/HTML Content and Controls* chapter.

[Content_Type]	[File_Serve]
[Header]	[HTML_Comment]
[Output]	[Output_None]
[Redirect_URL]	[Server_Push]

Results 

Tags which provide information about the current Lasso action and about the results of the current action. For more information, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.

[Found_Count]	[MaxRecords_Value]
[Lasso_CurrentAction]	[Req_Field]
[Required_Field]	[Response_FilePath]
[Response_LocalPath]	[Response_Path]
[Response_Realm]	[Search_Arguments]
[Search_FieldItem]	[SearchOperatorItem]
[Search_ValueItem]	[Shown_Count]
[Shown_First]	[Shown_Last]
[SkipRecords_Value]	[Sort_Arguments]
[Sort_FieldItem]	[Sort_OrderItem]
[Total_Records]	

Utility

Tags that allow utilitarian operations to be performed. This includes tags which allow you to set and retrieve cookies, write to log files, validate string types, and more. For more information about these various tags, see the Lasso 8 Language Guide or Lasso Reference.

[Event_Schedule]	[FTP_GetFile]
[FTP_GetListing]	[FTP_PutFile]
[HTTP_GetFile]	[Inline]
[Lasso_UniqueID]	[Log]
[Log_Critical]	[Log_Detail]
[Log_Deprecated]	[Log_SQL]
[Log_Warning]	[NSLookup]
[Server_Name]	[Server_Port]
[Valid_CreditCard]	[Valid_Email]
[Valid_URL]	

Lasso Programming Objects

The Lasso Programming objects are used to add string or math calculations, file manipulations, perform administrative tasks, use conditional logic, display error messages, and more. They can be used to store and retrieve data from tokens, cookies and variables. They can also be used to insert formatted dates and times into a page, parse XML data, or create PDF documents on the fly.

Figure 4: Lasso Programming Objects



Each Lasso Programming object is described below along with a list of the tags available for the object. Once the object has been inserted in a document, it is represented by an icon and can be modified using a related property inspector. The property inspectors for each object are described in the *Using Lasso Inspectors* chapter. The use of each of the Lasso tags is outside the scope of this manual. Please consult the Lasso Reference or the Lasso 8 Language Guide for more information about programming in Lasso.

Administration

Tags that allow authentication and Lasso Professional 8 administrative tasks to be performed. For more information, see the *Control Tags* chapter in the Lasso 8 Language Guide.

[Admin_ChangeUser]	[Admin_CreateUser]
[Admin_GroupAssignUser]	[Admin_GroupListUsers]
[Admin_GroupRemoveUser]	[Admin_ListGroups]
[Auth]	[Auth_Admin]
[Lasso_DatasourcesFileMaker]	[Lasso_DatasourcesLassoMySQL]
[Lasso_DatasourcesMySQL]	[Lasso_DatasourceModuleName]
[Lasso_TagExists]	[Lasso_TagModuleName]
[Lasso_Version]	[Log_SetDestination]
[Tags]	

Array

Tags which allow you to manipulate array, pair, and map data types. For more information, see the *Arrays and Maps* chapter in the Lasso 8 Language Guide.

[Array]	[List]
[Map]	[Pair]
[PriorityQueue]	[Queue]
[Set]	[Stack]
[TreeMap]	[Iterator]
[Reverseliterator]	

Condition

Tags which allow you to implement conditional logic. Condition tags are container tags, and insert opening and closing tag objects. For more information, see the *Conditional Logic* chapter in the Lasso 8 Language Guide.

[Abort]	[Case]
[Else]	[If]
[If_False]	[If_True]
[Iterate]	[Loop]
[LoopAbort]	[LoopCount]
[Repetition]	[Select]
[While]	

Cookie

Tags which allow you to set and retrieve cookies in a Web page. For more information, see the *HTTP/HTML Content and Controls* chapter in the Lasso 8 Language Guide. See also the Session Lasso Object.

[Cookie]	[Cookie_Set]
----------	--------------

Custom Tag

Tags which allow custom Lasso tags to be created. For more information, see the *Custom Tags* chapter in the Lasso 8 Language Guide.

[Define_Tag]	[Define_Type]
[Params]	[Params_Up]
[PostCondition]	[PreCondition]
[Return]	[Return_Value]
[Run_Children]	[Self]
[Tag_Name]	

Date

Tags which allow you to manipulate and format date values. For more information, see the *Date and Time Operations* chapter in the Lasso 8 Language Guide.

[Date]	[Date_Add]
[Date_Difference]	[Date_Format]
[Date_GetCurrentDate]	[Date_GetDay]
[Date_GetDayOfWeek]	[Date_GetHour]
[Date_GetLocalTimeZone]	[Date_GetMinute]
[Date_GetMonth]	[Date_GetSecond]
[Date_GetYear]	[Date_GMTToLocal]
[Date_LocalToGMT]	[Date_Subtract]
[Duration]	

Encryption

Tags which allow data to be encrypted and decrypted. For more information, see the *Encoding* chapter in the Lasso 8 Language Guide.

[Cipher_Decrypt]	[Cipher_Encrypt]
[Cipher_List]	[Decrypt_BlowFish]
[Decrypt_BlowFish2]	[Encrypt_BlowFish]
[Decrypt_BlowFish2]	[Encrypt_MD5]

Error

Tags which allow you to check, display and set error codes. For more information, see the *Error Control* chapter in the Lasso 8 Language Guide.

[Error_AddError]	[Error_CurrentError]
[Error_DatabaseConnection]	[Error_DatabaseTimeout]
[Error_DatabaseTimeout]	[Error_DeleteError]
[Error_FieldRestriction]	[Error_FileNotFound]
[Error_InvalidDatabase]	[Error_InvalidPassword]
[Error_InvalidUsername]	[Error_ModuleNotFound]
[Error_NoError]	[Error_NoPermission]
[Error_NoRecordsFound]	[Error_OutOfMemory]

[Error_RequiredFieldMissing]	[Error_SetErrorCode]
[Error_SetErrorMessage]	[Error_UpdateError]
[Fail]	[Fail_If]
[Handle]	[Protect]

Expression

Inserts a custom Lasso expression. For more information, see the section on Lasso expressions in the *Using Lasso Inspectors* chapter.

```
[ ]
```

File

Tags which allow you to read, write and manipulate files on the Web server. For more information, see the *Files and Logging* chapter in the Lasso 8 Language Guide.

[Directory_Lister]	[DirectoryNameItem]
[File]	[File_Copy]
[File_Create]	[File_CreationDate]
[File_CurrentError]	[File_Delete]
[File_Exists]	[File_GetLineCount]
[File_GetSize]	[File_IsDirectory]
[File_ModDate]	[File_Move]
[File_Read]	[File_ReadLine]
[File_Rename]	[File_SetSize]
[File_Uploads]	[File_Write]

Java

Inserts a tag that allows Java Beans to be used within Lasso.

```
[Java_Bean]
```

LassoScript

Inserts a LassoScript expression. For more information, see the *Lasso Syntax* chapter in the Lasso 8 Language Guide.

```
<? LassoScript ?>
```

Math

Tags which allow you to perform mathematical operations. For more information, see the *Math Operations* chapter in the Lasso 8 Language Guide.

[Decimal]	[Integer]
[Math_Abs]	[Math_Acos]
[Math_Add]	[Math_Asin]
[Math_Atan]	[Math_Atan2]
[Math_Ceil]	[Math_Cos]
[Math_Div]	[Math_Exp]

[Math_Floor]	[Math_Ln]
[Math_Log10]	[Math_Max]
[Math_Min]	[Math_Mod]
[Math_Mult]	[Math_Pow]
[Math_Random]	[Math_Rint]
[Math_Round]	[Math_Sin]
[Math_Sqrt]	[Math_Sub]
[Math_Tan]	[Math_ConvertEuro]

Namespace

Tags which manipulate Lasso's tag namespaces.

[Namespace_Current]	[Namespace_Delimiter]
[Namespace_Global]	[Namespace_Import]
[Namespace_Load]	[Namespace_Unload]
[Namespace_Using] ... [/Namespace_Using]	

Net

Tags which are used for network operations. For more information, see the *Networking* chapter in the Lasso 8 Language Guide.

[Net]

PDF

Tags which are used to generate and manipulate PDF documents on-the-fly. These tags may also be used for generating barcodes and graphics. For more information, see *Chapter 30: Portable Document Format* in the Lasso 8 Language Guide.

[PDF_Barcode]	[PDF_Doc]
[PDF_Font]	[PDF_Serve]
[PDF_Table]	

Session

Tags which allow custom user sessions and server-side variables to be created. For more information, see the *Sessions* chapter in the Lasso 8 Language Guide.

[Session_AddVariable]	[Session_AddVar]
[Session_End]	[Session_ID]
[Session_RemoveVariable]	[Session_RemoveVar]
[Session_Result]	[Session_Start]

String

Tags which allow you to manipulate string values. For more information, see the *String Operations* chapter in the Lasso 8 Language Guide.

[String_Concatenate]	[String_CountFields]
[String_EndsWith]	[String_Extract]
[String_FindPosition]	[String_GetField]
[String_Insert]	[String_IsAlpha]
[String_IsAlphanumeric]	[String_IsDigit]
[String_IsHexDigit]	[String_IsLower]
[String_IsNumeric]	[String_IsPunctuation]
[String_IsSpace]	[String_IsUpper]
[String_Length]	[String_Lowercase]
[String_Remove]	[String_RemoveLeading]
[String_RemoveTrailing]	[String_Replace]
[String_ToDecimal]	[String_ToInteger]
[String_Uppercase]	

Technical

Tags which allow server process operations to be performed. For more information about these various tags, see the Lasso 8 Language Guide or Lasso Reference.

[Boolean]	[Compress]
[Decompress]	[LassoApp_Create]
[LassoApp_Dump]	[LassoApp_Link]
[NoProcess]	[Null]
[Process]	[Reference]
[Sleep]	[TCP_Close]
[TCP_Open]	[TCP_Send]
[Thread_Event]	[Thread_Lock]
[Thread_Pipe]	[Thread_RWLock]
[Thread_Semaphore]	

Token

A tag which allows you to fetch a token value. For more information, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide. See also the -Token object in the Lasso Form Objects.

[Token_Value]

Variable

Tags which allow you to set and retrieve values within a page. For more information, see the *Programming Fundamentals* chapter in the Lasso 8 Language Guide.

[Global]	[Globals]
[Globals_Defined]	[Local]
[Locals]	[Locals_Defined]
[Variable]	[Var]
[Variable_Defined]	

XML 

Tags which allow XML to be parsed and served. XML-RPC actions may also be performed. For more information, see the *XML* chapter in the Lasso 8 Language Guide.

[XML]	[XML_Extract]
[XML_RPCCall]	[XML_RPC]
[XML_Serve]	[XML_Transform]

Lasso Form Objects

The Lasso Form objects are used to add user-modifiable inputs, buttons and controls and hidden inputs to Lasso forms. You can create Lasso forms using the Form Builder, described in the *Using Form Builder* chapter, or you can create them from scratch using the Lasso Form objects.

Figure 5: Lasso Form Objects



The Lasso Form objects represent Lasso command tags and HTML inputs that allow you to give additional attributes to Lasso about how the action specified in the form should be carried out.

Action 

The Action object allows you to specify a Lasso action in a form. Lasso actions only exist in forms when using the Classic Lasso method (as opposed to the Inline method), which is described in the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

-Add	-Delete
-Duplicate	-FindAll
-Image	-Nothing
-Random	-Scripts
-Search	-Show
-SQL	-Update

Button 

Inserts an HTML button, reset or submit input. The Lasso attributes can be modified in the Lasso Form Inspector.

Checkbox

Inserts an HTML checkbox input. The Lasso attributes can be modified in the Lasso Form Inspector.

```
<input type="checkbox" name="Field Name" value="[Field: 'Field Name']">
```

Database

The Database object allows you to specify information in a form using the Classic Lasso method (as opposed to the Inline method), which is described in the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

The -Required input can be inserted immediately before any user input to specify that the user must enter a value for that input. If the input is left blank, the -ResponseRequiredFieldMissingError page will be shown instead.

-Database	-Datasource
-Distinct	-KeyField
-KeyValue	-Layout
-MaxRecords	-RecordID
-Req	-Required
-ReturnField	-SkipRecords
-SortField	-SortOrder
-SortRandom	-Table
-UseLimit	

Form

The Form object inserts an HTML form with Action.lasso in the Action attribute. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

```
<form action="/Action.lasso" method="POST">
...
</form>
```

Operator

Inserts a Lasso search operator in a form. This is used ahead of specific fields to determine how they will be searched. For more information, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.

-Op	-OpBegin
-OpEnd	-OpLogical
-Operator	-OperatorBegin
-OperatorEnd	-OperatorLogical

Radio Button

Inserts an HTML radio button input. The Lasso attributes can be modified in the Lasso Form Inspector.

```
<input type="radio" name="Field Name" value="[Field: 'Field Name']">
```

Response

The Response object allows you to specify what page will be returned as a result of a form submission. This page will usually present the result of the form action, either a listing display of the results of a search or a detail display for an individual record. Different response pages can be specified for different Lasso actions or -AnyResponse can be used for any Lasso action. These hidden inputs can also be inserted or modified in the Form property inspector. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

-Response	-ResponseAdd
-ResponseAddError	-ResponseAny
-ResponseAnyError	-ResponseDelete
-ResponseDuplicate	-ResponseLassoApp
-ResponseNoResultsError	-ResponseReqFieldMissingError
-ResponseRequiredFieldMissingError	
-ResponseSecurityError	-ResponseUpdate
-ResponseUpdateError	

Select List

Inserts an HTML select list. When the HTML Size and Multiple attribute are not used, this becomes a pull-down menu. Otherwise, it is a multiple-item select list. The Lasso attributes can be modified in the Lasso Form Inspector.

```
<select name="Field Name" multiple size="5">
  <option value="[Field: 'Field Name']">[Field: 'Field Name']</option>
  ...
</select>
```

Script

The Script Object allows you to specify a Filemaker script to be run in a Filemaker Pro data source in concert with the current database action (FileMaker Pro databases only). For more information, see the *Filemaker Data Sources* chapter in the Lasso 8 Language Guide.

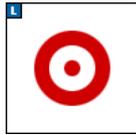
-FMScript	-FMScriptPost
-FMScriptPre	-FMScriptPreSort

Site Builder Target

The Site Builder features the ability to use a template document as the basis for each of the files it creates. Templates can be any HTML document which contains a special marker tag.

```
<ls4gl_builder_target></ls4gl_builder_target>
```

Use this object to insert this marker into an HTML document in order to use it as a Site Builder template. The Site Builder will insert Lasso-based forms at the location of the Site Builder Target icon. See the *Using Site Builder* chapter for more details about using the Site Builder.



Text Area

Inserts an HTML text area. The Lasso attributes can be modified in the Lasso Form Inspector. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

```
<textarea name="Field Name"></textarea>
```

Text Input

Inserts an HTML text input. The Lasso attributes can be modified in the Lasso Form Inspector. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

```
<input type="text" name="Field Name" value="[Field: 'Field Name']">
```

Token

A tag which allows you to set a token value that can be retrieved on the response page. See also the [Token_Value] object in the Lasso Programming Objects. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

```
-Token
```

Utility

Utility form tags include -Username and -Password for submitting browser authentication values in a form. These can be hidden inputs or text inputs so the user can enter a username and password. For more information, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.

```
-Password
```

```
-Username
```

Using Lasso Objects

This section provides an example of using Lasso Objects in a Web page. No matter what Lasso Object you insert into a page, the procedure is similar.

To insert a Lasso Object:

- 1 Choose the appropriate Lasso section from the Insert panel.
- 2 Find the Lasso Object you want to insert, and drag and drop it into a location in the document window.
- 3 In the document window, select the resulting icon if it's not already selected. This displays the properties for the object in the Lasso Inspector panel.

Performance Note: When selecting a Lasso Object, Dreamweaver may pause for up to three seconds before refreshing the panel. If this occurs, please be patient until Dreamweaver properly refreshes the panel.

- 4 Use the Tag pull-down menu in the Lasso Inspector to choose which tag you would like to use in your page, as described in the *Using Lasso Inspectors* chapter.
- 5 Then, either customize the attributes presented in the property inspector, or click the Edit... button to show the Tag Editor screen, which is described in the *Using Lasso Tag Editors* chapter.

Describing how to use each individual Lasso tag within a solution is beyond the scope of this guide. However, an example of using the [Inline] Lasso Object is provided in this section. For more information on how to use other Lasso tags and objects, refer to the Lasso 8 Language Guide and Lasso Reference, which describe the Lasso language.

Writing Lasso

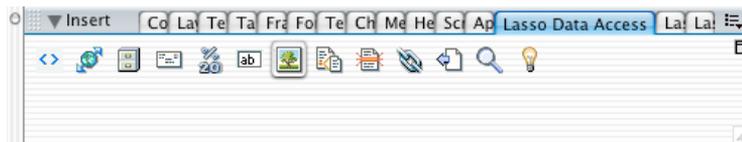
It should be noted that Lasso code may also be written directly in the Code view in Dreamweaver, and programming Lasso in Lasso Studio is not limited to using drag-and-drop Lasso Objects. Lasso Studio offers the Form Builder, Site Builder, Lasso Objects, and the Code view of Dreamweaver as methods of authoring Lasso code.

Inline Object Example

The example below describes inserting an Inline Lasso Object into a document to perform a search on a database and display the results in the same page.

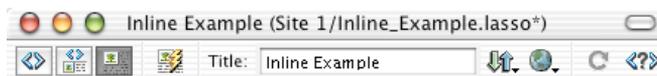
- 1 In Dreamweaver, create a new document or open an existing document in which you wish to add Lasso data access tags. The page file name should end with a .lasso extension.
- 2 In your page, select a location where you want your Lasso Objects to be inserted using the Dreamweaver text selection tool. You can insert a few carriage returns in the page so you have more than one location to insert Lasso Objects.
- 3 In the Insert panel, select the Lasso Data Access section.

Figure 6: Lasso Data Access Objects



- 4 Locate the [Inline] Lasso Object, and drag-and-drop it into a location in the document window. Four icons are inserted, which represent [Inline] ... [/Inline] tags with [Records] ... [/Records] tags contained.
- 5 In the document window, select the left-most [Inline] object. This displays the properties for the [Inline] tag in the Lasso Inspector panel. If the Lasso Inspector panel is not visible, then you can show it from the Lasso Studio menu.

Figure 7: Inline Example



```
<body> <p> 446 x 177 3K / 1 sec
```

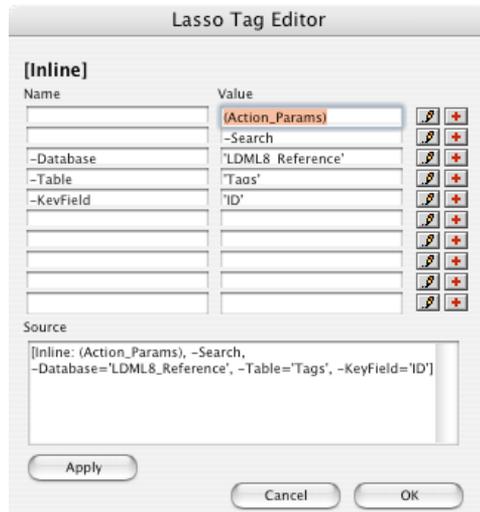
- 6 The Lasso Inspector panel will show the source of the [Inline] tag, as well as a list of parameters for the [Inline] tag that can be added, removed, and modified. For this example, leave all checkboxes checked next to each parameter as they will all be used.

Figure 8: Inline Inspector



- 7 In the Action pull-down menu, select -Search. This means the [Inline] tag will perform a search action on the database specified.
- 8 In the Database pull-down menu, select the name of the database you wish to search. This will populate the Table pull-down menu with all tables currently in the database.
- 9 In the Table pull-down menu, select the name of the table (or FileMaker layout) in the database you wish to use. This will populate the Key Field pull-down menu with all fields currently in the table.
- 10 In the Key Field pull-down menu, select the name of the key field for the table.
- 11 Finally, we wish to hard-code a search argument into the [Inline] tag rather than use [Action_Params] to retrieve search arguments from a preceding form. Hard-coding a search argument means that the same search will be performed every time the pages are loaded, and it does not require a preceding form. To change [Action_Params] to a specific search argument, select the Tag Editor... button to open the Lasso Tag Editor screen.

Figure 9: Lasso Tag Editor



- 12 In the top-most Value field, delete the (Action_Params) tag, as it will not be used.
- 13 Now, we want to add a new parameter to the [Inline] tag. Next to the first blank Value field below 'ID', select the Pencil icon. This launches the Lasso Parameter Editor screen.
- 14 In the upper Name box of the Lasso Parameter Editor screen, select the Field radio button. This means we will search a specific field in the database.
- 15 In the Field pull-down menu to the right, select a field you wish to search.
- 16 In the lower Value box, select the Literal radio button. This means we want to enter a literal value to search for in the field selected above.
- 17 In the text input next to Literal, enter the name of a field value to search for, surrounded by single-quotes. For example, if you selected a field called Company in the Field pull-down menu, then enter the name of the company to search for in the text area, e.g. 'OmniPilot'.

Figure 10: Lasso Parameter Editor

The screenshot shows the Lasso Parameter Editor dialog box. It has a title bar that says "Lasso Parameter Editor".

Name

- Field Name: 'ID'
- Variable: Variable
- Tag Selector: Select
- Parameter: -Database
- Literal: -Database
- No Name (Value Only)

Value

- Field Value: (Field: 'ID')
- Variable: Variable
- Tag Selector: Select
- Keyword: -Add
- File Name: [empty]
- Literal: 'LDML8_Reference'

Source

-Database='LDML8_Reference'

Buttons: Cancel, OK

- 18 Select the OK button to close the Lasso Parameter Editor screen.
- 18 Select the OK button to close the Lasso Tag Editor screen. We have now finished editing the [Inline] tag.
- 19 In the Lasso Inspector panel, select the Submit button to commit all changes to the [Inline] tag. The completed [Inline] tag should look similar to the example below in the Lasso Inspector.


```
[Inline: -Search,
  -Database='Contacts',
  -Table='People',
  -KeyField='ID',
  'Company'='OmniPilot']
```
- 20 Next, we need to insert tags to display the results of the search within the [Inline] ... [/Inline] tags. Click in between the two [Records] tag icons and insert a carriage return.

Figure 11: Inline Example



- 21 Using the Lasso Data Access section of the Insert panel, insert a [Field] tag represented by the `ab` icon.

Figure 12: Inline Example



- 22 Select the [Field] tag and use the Lasso Inspector to select a field to display from the Field pull-down menu.
- 23 Select the Submit button. We have now finished creating the code necessary to perform a search and display one field for each record found. The resulting code in the Code view should be similar to the following

```
[Inline: -Search,
  -Database='Contacts',
  -Table='Web',
  -KeyField='ID',
  'Company'='OmniPilot']
```

```
[Records]
  [Field: 'Last_Name']
[/Records]
[/Inline]
```

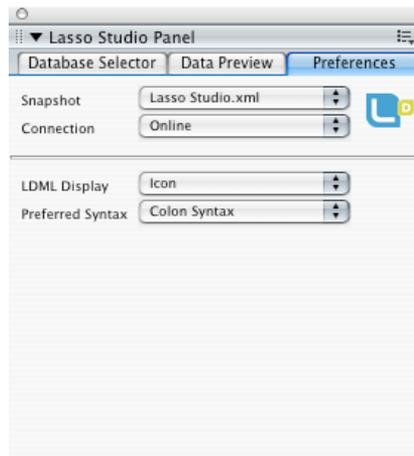
Everything between the filing cabinets [Records] tags will be repeated for each record in the found set, and all [Field] tags will be displayed on the page. To display multiple fields, simply repeat steps 21-23 for each additional field. Field objects may be formatted and delimited in any way desired, so long as they remain between the [Records] ... [/Records] tags.

Lasso Display Preferences

Lasso Studio allows you to change how Lasso Objects are displayed in the Design view of Dreamweaver. You can choose to display Lasso Objects as icons, Lasso tag names, or Lasso expressions. This section describes configuring Lasso Object display properties in Dreamweaver.

Lasso Display Preferences are configured in the Preferences section of the Lasso Studio panel. Lasso Display Preferences are changed by selecting a new Lasso display option from the Lasso Display pull-down menu. Each of the Lasso display options are described below.

Figure 13: Lasso Display Preferences



The following Lasso display options are available in the Lasso Display pull-down menu.

- **Icon** – Displays a Lasso Object icon from the Insert panel whenever a Lasso tag is used within a page. For example, the Lasso tag

[Date: '12/15/2002'] will appear as the following in the Design view of Dreamweaver.



- **Tag Name** – Displays the Lasso tag name only whenever a Lasso tag is used within a page. For example, the Lasso tag [Date: '12/15/2002'] will appear as the following in the Design view of Dreamweaver.

[Date]

- **Full Expression** – Displays the full Lasso tag expression (as it appears in the source) whenever a Lasso tag is used within a page. For example, the Lasso tag [Date: '12/15/2002'] will appear as the following in the Design view of Dreamweaver.

[Date: '12/15/2002']

- **None** – Displays nothing in the Design view of Dreamweaver whenever a Lasso tag is used within a page. This is useful for previewing how a document will look after it has been served and all Lasso tags have been parsed by the server.

To change the current Lasso display option, simply select a new option from the Lasso Display pull-down menu. When doing so, Dreamweaver automatically switches all documents to the Design view and applies the changes. Lasso Display Preferences are purely cosmetic in Dreamweaver, and do not affect any Lasso code in the page source.

Preferred Syntax

The preferred syntax popup menu controls what syntax Lasso Studio generates code in. The options include:

- **Colon Syntax** – This was the only syntax of prior versions of Lasso. Tags consist of a tag name followed by a colon and then the parameters of the tag. If you are familiar with earlier versions of Lasso then this option will be most familiar.
- **Parentheses Syntax** – In this new syntax option in Lasso 8 tags consist of the tag name followed by parentheses that contain the parameters of the tag. This syntax style will be more familiar to users of other languages like C or Java.

Lasso Studio can edit and work with tags in either syntax style. This preference only controls how syntax created by the Site Builder, Form Builder, Lasso objects, and other elements will be formatted when created from scratch.

9

Chapter 9

Using Lasso Inspectors

Lasso Studio includes a Lasso Inspector panel that allows you to view and modify the common properties of any Lasso tag, hidden input, or form element.

- *Overview* provides an overview of Lasso Inspector operation with Lasso tags.
- *Using the Lasso Inspector* describes using the various Lasso inspector interfaces for different types of Lasso Objects.
- *Substitution and Process Tags* provides examples of using the Lasso Inspector with substitution and process Lasso tags.
- *Container Tags* provides examples of using the Lasso Inspector with container Lasso tags.
- *Member Tags* provides examples of using the Lasso Inspector with member Lasso tags.
- *Form Elements* provides examples of using the Lasso Inspector with Lasso form objects.
- *LassoScript and Expressions* provides examples of using the Lasso Inspector with LassoScript and Lasso expressions.

Overview

The Lasso Inspector is a tool in Lasso Studio that makes editing Lasso code easier within the Design view of Dreamweaver. Lasso Objects in Lasso Studio are edited via the Lasso Inspector panel, which is accessed from the Lasso Studio menu. The Lasso Inspector panel provides an interface specific to each Lasso Object.

Upon selecting a Lasso Object in a page, the Lasso Inspector panel shows a list of selectable parameters specific to that object, and also allows the raw Lasso code source for the object to be viewed and edited. The Lasso Inspector panel also allows the Lasso Tag Editor to be launched for a selected object, which is described in the *Using Lasso Tag Editors* chapter.

As each Lasso Object is different, the Lasso Inspector will appear with different options for each tag. However, there are three main types of Lasso Inspector interfaces, which are as follows.

- **Lasso Tag Inspector** – This inspector interface is used for substitution tags, process tags, container tags, and member tags in Lasso, which constitute most of the tags in the Lasso language. This is the primary Lasso Inspector interface that will be seen for most tags.
- **Lasso Form Inspector** – This inspector interface is shown when a Lasso tag is embedded within an HTML form element, and allows both Lasso and HTML elements to be edited.
- **LassoScript/Expression Inspector** – This inspector interface is shown whenever a LassoScript or Lasso expression Lasso Object is selected, and allows free-form editing of the Lasso syntax.

Each of these inspector interfaces are described in the *Using Lasso Inspectors* section of this chapter. For more information on Lasso Objects and tag types, see the *Using Objects and Tags* chapter.

The Lasso Inspector is not the only way to edit Lasso code, and may be used in concert with the other editing methods such as using the Code view in Dreamweaver.

Syntax Note

The inspectors will preserve the syntax of each tag which is edited using them. They will also attempt to preserve the white space within the tag. However, new tags will be created with according to the global syntax preference for either colon or parentheses syntax. This preference can be adjusted in the Lasso Studio panel.

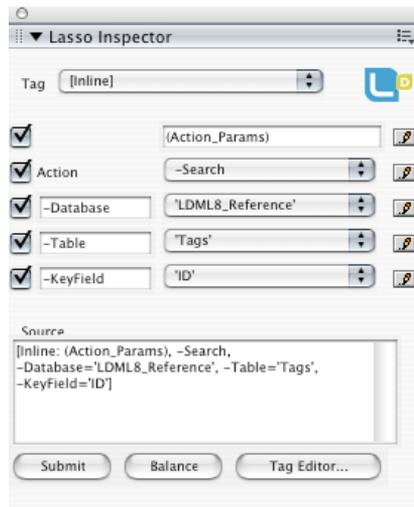
Using the Lasso Inspector

This section describes the elements of the Lasso Inspector interfaces when used to modify the attributes of Lasso tags. The Lasso Inspector panel where all interfaces are contained is shown by selecting Show Lasso Inspector from the Lasso Studio menu in Dreamweaver.

Lasso Tag Inspector

The Lasso Inspector panel becomes the Lasso Tag Inspector panel whenever a Lasso substitution, process, container, or member tag is selected. The Lasso Tag Inspector has the same basic interface for each of these tag types.

Figure 1: Lasso Tag Inspector



The interface elements of the Lasso Tag Inspector panel are described below. The interface of the Lasso Tag Inspector is customized for each of the more than a thousand tags in Lasso.

- **Tag** – The Tag pull-down menu at the top of the Inspector panel shows the current Lasso tag used by a selected object, and can be used to select a related tag from the current tag category. As described in the previous chapter, a Lasso Object often represents a group of tags, from which one must be selected in the Lasso Inspector panel.
- **Parameter Checkboxes** – The Lasso Inspector panel shows a maximum of five selectable parameters for the current tag, which allow the attributes of the selected tag to be modified. These parameters appear directly below the Tag pull-down menu, and the checkbox beside each parameter can be used to add or remove the parameter from the tag being edited. If more than five parameters are needed for a tag, then the Lasso Tag Editor should be used. For more information, see the *Using Lasso Tag Editors* chapter.

Parameter names and keywords are shown on the left side of the panel, while parameter values are shown on the right. Parameter values may be edited directly in the Lasso Inspector panel, or via the Lasso Parameter Editor

screen, which appears for a parameter by selecting the pencil icon to the right. For more information on this, see the *Using Lasso Tag Editors* chapter.

The number and types of parameters shown will vary greatly depending on the Lasso tag currently selected in the Tag pull-down menu. The Lasso 8 Language Guide or Lasso Reference should be consulted for more information about required and optional parameters for specific tags.

- **Source** – The source area at the bottom of the Inspector shows how the current tag appears in the source code. Modifications can be made to the tag by editing the text in this area and then selecting the Submit button. Any changes made manually in the Source text area will override the parameters selected above.

Note: The Tag Inspector will be immediately refreshed based upon the committed source code edit.

- **Balance** – The Balance button is active for Lasso container tags. It automatically selects the matching opening and closing tag of the current tag. If the Balance button cannot find a matching opening or closing tag, then it will display an error message.
- **Tag Editor** – The Tag Editor button is active for tags that can be edited using the Lasso Tag Editor screen. The Lasso Tag Editor allows up to 10 parameters to be viewed and edited for the current tag, and makes it easy to build or modify complex tags that contain too many parameters to be represented in the Lasso Tag Inspector. The Lasso Tag Editor screen is described in the *Using Lasso Tag Editors* chapter.
- **Submit** – Selecting the Submit button applies all changes made to the currently selected object to the Web page. No changes are applied to the page until this button is selected.

The Lasso Tag Inspector is used by selecting a tag, modifying its parameters, and then selecting the Submit button. Tag parameter modification can occur via the parameter checkboxes, Lasso Tag Editor screen, Lasso Parameter Editor screen, or the Source text area.

Lasso Form Inspector

The Lasso Form Inspector is similar in function to the interface for any HTML form element which is selected in Dreamweaver, but has pre-inserted values in the form element fields that are applicable to the Lasso form object selected. For example, selecting the -Token Lasso Object will display the HTML properties for a hidden form input, but has the -Token.Name tag pre-inserted into the Name HTML attribute of the hidden input.

Figure 2: Lasso Form Inspector



The Tag pull-down menu provides access to the other command or action tags in the same category as the tag being inspected. Selecting a new tag changes the Name attribute of the hidden input. The Value attribute of the hidden HTML input can be changed using the Value field. Selecting the pencil icon shows the Lasso Parameter Editor screen for a parameter, as described in the previous section.

The Source text area shows the complete HTML and Lasso source for the currently selected object, which may be edited manually. The Balance and Tag Editor... buttons are never available when the Lasso Tag Inspector is being used to modify a Lasso form object.

The Lasso Form Inspector is used by selecting a command tag, modifying HTML attribute values for the tag, and then selecting the Submit button. The Lasso Form Inspector works in concert with the Lasso Tag Inspector and the built-in Dreamweaver form inspector to provide access to all of the attributes of HTML form elements.

LassoScript/Expression Inspector

This inspector interface is shown whenever a LassoScript or Lasso expression Lasso Object is selected, and allows free-form editing of the Lasso syntax. For an overview of LassoScript and Lasso expressions, see the *Methodology* section of the previous chapter.

Figure 3: LassoScript/Expression Inspector



A single Source text area allows a LassoScript object or Lasso expression to be manually edited. The LassoScript/Expression Inspector is used by entering the desired LassoScript or Lasso expression code in the Source window, and selecting the Submit button.

Substitution and Process Tags

This section provides a series of examples for editing the properties of Lasso substitution and process tags in the Lasso Inspector panel. Substitution tags make up the bulk of Lasso, as described in the *Using Objects and Tags* chapter. They allow you to insert database, browser, and server information into an HTML page.

Date Example

The following example describes how to use a Date Lasso Object to dynamically display the current date in a specified format in a Web page. For more information on date Lasso tags, see the *Date and Time Operations* chapter in the Lasso 8 Language Guide.

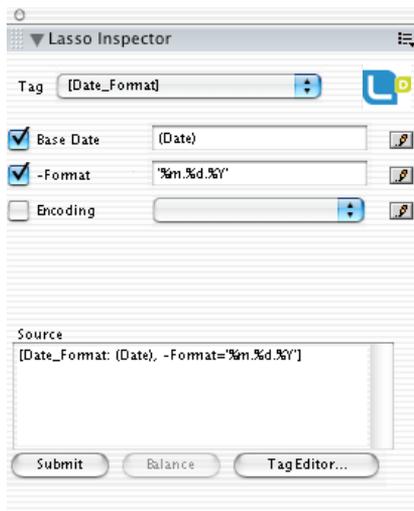
To inspect and edit the properties of a Date Lasso Object:

- 1 Drag and drop a Date Lasso Object from the Lasso Programming section of the Insert panel into your document.



- 2 In the document window, select the Date object. This displays the properties for the Date object in the Lasso Inspector panel.
- 3 In the Lasso Inspector panel, select [Date_Format] from the Tag pull-down menu. We will use the [Date_Format] tag to output a date in a format we specify.

Figure 4: Lasso Inspector



- 4 In the Base Date parameter field, replace 'Date' with the base date value you wish to format and display. For this example, simply enter (Date). The (Date) tag dynamically outputs the current date of the server when the page is loaded.

Syntax Note: Because the [Date] tag is contained within the [Date_Format] tag, it is formatted in parentheses rather than square-brackets. Per Lasso syntax conventions, all sub-tags are formatted in parentheses.

- 5 Next to the -Format parameter, leave the checkbox checked and enter the date format string '%m.%d.%Y' in the text field to the right. This means we want to display a date in Month.Day.Year format. Be sure to properly surround the date format string in single-quotes, which is how all string values must be formatted in Lasso.
- 6 Upon making the changes to the two parameters above, the following should be displayed in the Source text area.

```
[Date_Format: (Date), -Format='%m.%d.%Y']
```

- 7 Select the Submit button. This applies the Date object changes to your Web page.
- 8 To further customize how your Date object will appear when served, select the Date object and insert HTML text attributes such as bold, italic, or font tags using the Dreamweaver Tool bar.

```
<b><i><font face="Arial">
[Date_Format: (Date), -Format='%m.%d.%Y']
</font></i></b>
```

When the page is served, the current date will be shown in Month.Day.Year format in the font and style specified.

Email Example

The following example describes how to use an Email Lasso Object to dynamically send a pre-defined email message upon loading a Web page. For documentation on the [Email_Send] tag, see the *Email* chapter in the Lasso 8 Language Guide.

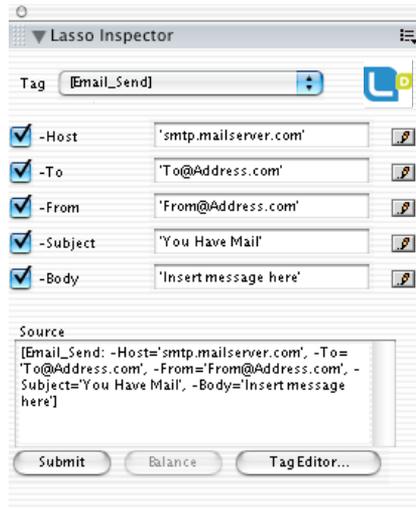
To inspect and edit the properties of an Email Lasso Object:

- 1 Drag and drop the Email Lasso Object  from the Lasso Programming section of the Insert panel into your document.



- 2 In the document window, select the Email object. This displays the properties for the Email object in the Lasso Inspector panel.
- 3 In the Lasso Inspector panel, select [Email_Send] from the Tag pull-down menu. This is the only Lasso tag used in Lasso Professional 8 to send email.

Figure 5: Lasso Inspector



- 4 Next to the -Host parameter, leave the checkbox checked and enter the IP address or host name of a configured SMTP mail server. Lasso Professional 8 requires an SMTP mail server to send email. For more information, see the Lasso Professional 8 Setup Guide. Also be sure to properly surround the server name or IP address in single-quotes, which is how all string values must be formatted in Lasso.
- 5 In the -To parameter field, insert the email address of the addressee you wish to send a message to.
- 6 In the -From parameter field, insert the email address that you want the addressee to be able to reply to.
- 7 In the -Subject field, insert the subject of the email address.
- 8 In the -Body field, insert the text you wish to appear in the email message.
- 9 Select the Submit button. This applies the Email object changes to your Web page. The email you specified will be sent whenever that particular page is loaded. This is particularly useful for response pages.

Container Tags

This section provides examples for editing the properties of Lasso container tags. Container tags are used in Lasso for performing database actions, writing conditional statements, performing extended processes, and more. Container tags do not directly display values in a Web page, but are used to determine what values are output in the substitution tags they enclose.

Container tags in Lasso always consist of an opening and closing tag. Closing tags are handled automatically by the Lasso Tag Inspector when modifications are made to an opening tag. The example below shows the [HTML_Comment] Lasso container tag.

[HTML_Comment] Opening tag on left, Closing tag on right [/HTML_Comment]

When the Tag pull-down menu is used to change a substitution tag into a container tag, the closing tag is inserted automatically in the document. The closing tag can then be moved independently. Likewise, when the Tag pull-down menu is used to change a container tag into a substitution tag, the matching closing tag is automatically deleted. If you then change the tag back to a container tag the new closing tag will immediately follow the opening tag which is being modified.

Note: If a container tag is changed to a different tag type, but a closing tag icon still appears in the Design view, simply switch to the Code view and then back to the Design view in Dreamweaver to refresh the page.

Inline Example

The [Inline] tag is one of the key container tags in the Lasso language, and is used to perform database actions. A full [Inline] tag example with the Lasso Tag Inspector is provided in the *Using Objects and Tags* chapter.

Conditional Logic Example

The following example shows how to output different results messages in a response page depending on the value of a submitted form value. For more information on using conditional tags in Lasso, see the *Conditional Logic* chapter in the Lasso 8 Language Guide.

To inspect and edit the properties of Conditional Lasso Objects:

1 Drag and drop a Conditional Lasso Object  from the Lasso Programming section of the Insert panel into your document. This adds three icons to your page, which represent an opening [!f], [Else], and closing [!f] Lasso tags.



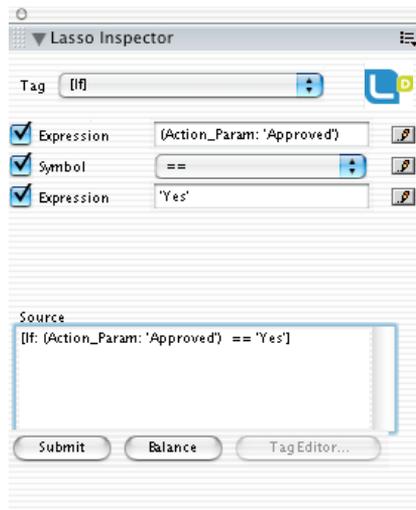
2 In the document window, select the opening Conditional object on the left. This displays the properties for the [!f] tag in the Lasso Inspector panel.

3 For our conditional expression, we want to display a message if a form value named Approved from a previous page is equal to Yes. Enter the following next to the first Expression field.

(Action_Param: 'Approved')

- 4 In the Operator pull-down menu, select == as the conditional operator, which means “equals.” For more information on conditional operators in Lasso, see the *Conditional Logic* chapter in the Lasso 8 Language Guide.
- 5 In the second Expression field, enter 'Yes'. This forms the finalized conditional statement [if: Action_Param: 'Approved'] == 'Yes' in the Source text area below.

Figure 6: Lasso Inspector



- 6 Select the Submit button to apply the changes to the [if] tag.
- 7 Next, we want to display the message This record has been approved! if the conditional expression is true. This is done by inserting the information you want to be displayed in between the [if] and [Else] Lasso Objects.

Figure 7: Code Preview



8 Next, we want to display the message This record has not been approved! if the conditional expression is false. This is done by inserting the information you want to be displayed in between the [Else] and [/If] Lasso Objects. This completes the basic conditional statement.

Figure 8: Code Preview



No modification of the [Else] object is needed as we simply want it to represent the opposite result of the [If] conditional expression. Closing container tags such as [/If] in Lasso are not modifiable.

When the page is loaded as a response page following a form, a different message will be displayed based on the value of a field named Approved in the form.

Member Tags

Member tags are often used with variables of specific data types (Integer, String, Date, etc.) to modify the data returned. Member tags must be called using the member symbol -> and a value from the appropriate data type.

In Lasso Studio, member tags may only be used by inserting member tag code manually in the Source window of the Lasso Inspector panel for an applicable tag. For more information on member tags and their applications, see the *Programming Fundamentals* chapter in the Lasso 8 Language Guide.

String Size Example

This section describes how to display the number of characters contained in a string data type output by a Lasso tag. Lasso tags that output string data types include [Action_Param], [Field], and [Token_Value], and [Variable]

when set to a string value. The example below shows how to use the `->Size` member tag to output the number of characters contained in a field.

To insert a string size member tag in a [Field] tag:

1 Find a Field Lasso Object `ab` in one of your existing documents.

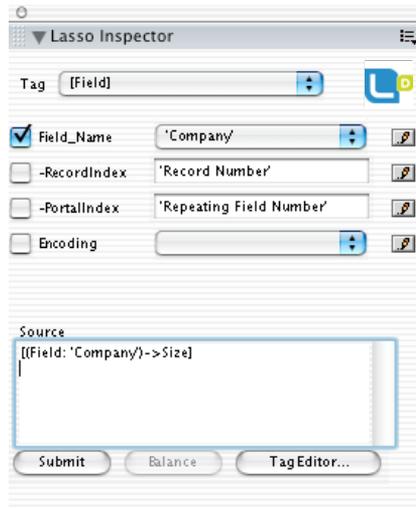


2 In the document window, select the Field object. This displays the properties for the Field object in the Lasso Inspector panel.

3 In the Source window, enclose the Field: 'Field Name' part of the `[Field: 'Field Name']` tag in parentheses. Doing so allows the field output part of the tag to be separated from the member tag, which will both exist between the same pair of brackets.

4 After the closing parentheses, insert `->Size` before the closing tag bracket. This is the member tag of a string data types that returns the number of characters in the string.

Figure 9: Lasso Inspector



5 Select the Submit button. When the Web page is served, the integer number of characters contained within the field will be displayed instead of the field content.

Form Elements

Lasso Objects in the Lasso Form section of the Insert panel show their properties in the Lasso Form inspector, as described earlier in this chapter. The Lasso Form Inspector can be used to associate an input with a specific field in the database, or to insert a calculated value. Both the HTML and Lasso attributes of a form element can be modified using the Lasso Form inspector.

Token Example

The example below shows how to insert a named Lasso token value in a form that will be passed to the response page when the form is submitted. For more information on tokens, see the *Searching and Displaying Data* chapter in the Lasso 8 Language Guide.

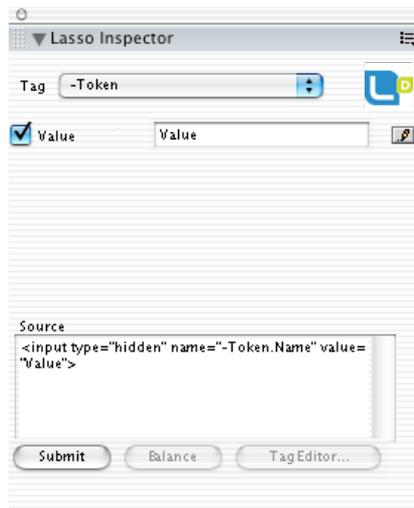
To inspect and edit the properties of a token form object:

- 1 Drag and drop a Token Lasso Object  from the Lasso Form section of the Insert panel into your document.



- 2 In the document window, select the Token object. This displays the properties for a -Token HTML hidden input in the Lasso Inspector panel.

Figure 10: Lasso Inspector



- 3 In the Value field, insert the value that you want to pass with the token name in the form. This will appear in the Value attribute of the HTML hidden input.
- 4 In the Source text area, replace -Token.Name with the name you wish to give the token, e.g. -Token.Myname. This appears in the Name attribute of the HTML input.
- 5 Select the Submit button. This will apply the changes to the page.

LassoScript and Expressions

Both LassoScript and Lasso Expression Lasso Objects use the LassoScript/Expression Inspector, which allows free-form editing of the Lasso syntax. For an overview of LassoScript and Lasso expressions, see the *Methodology* section of the previous chapter.

LassoScript Example

The following example shows how to perform an entire database query and output field data within a single LassoScript tag object. For more information on LassoScript, see the *Lasso Syntax* chapter in the Lasso 8 Language Guide.

To insert a LassoScript object:

- 1 Drag and drop a LassoScript Lasso Object  from the Lasso Programming section of the Insert panel into your document.



- 2 In the document window, select the LassoScript object. This displays the properties for the LassoScript object in the Lasso Inspector panel. Only a Source text area with `<? LassoScript ?>` should appear.
- 3 In the Source text area, enter the following LassoScript code.

Figure 11: Lasso Inspector



- 4 Select the Submit button. This will apply the changes to the page. When the page is served, the query will be performed and data in the field First Name will be output for each record in the People table of the Contacts database.

Lasso Expression Example

The following example shows how to perform a math operation using a Lasso expression. For more information on Lasso expressions see the *Using Objects and Tags* chapter *Methodology* section.

To insert a Lasso Expression object:

- 1 Drag and drop a Lasso Expression Lasso Object  from the Lasso Programming section of the Insert panel into your document.



- 2 In the document window, select the Lasso Expression object. This displays the properties for the Lasso Expression object in the Lasso Inspector panel. Only a Source text area with [] should appear.
- 3 In the Source text area, enter the following expression.

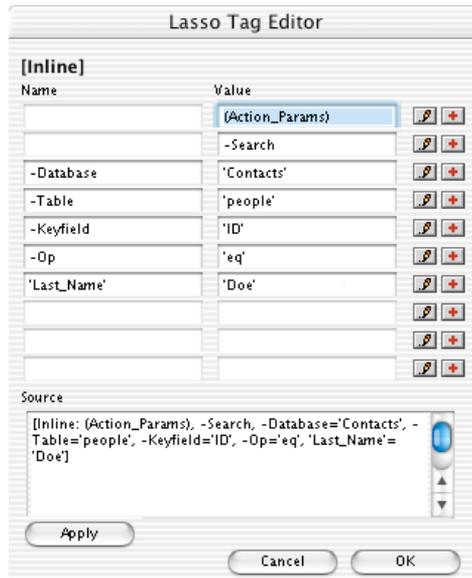
Figure 12: Lasso Inspector

- 4 Select the Submit button. This will apply the changes to the page. When the page is served, the number 6 will be output where the Lasso expression is placed.

Using the Tag Editor

The Tag Editor is launched by selecting the Tag Editor... button in the Lasso Inspector panel while a Lasso Object is selected. This shows the Lasso Tag Editor screen for the current tag selected in the Lasso Inspector panel with all parameters currently defined.

Figure 1: Tag Editor



A tag can be edited using the Lasso Tag Editor screen in one of three ways: by editing the values directly in the Name and Value fields; launching the Parameter Editor by selecting the selecting the pencil icon to the right of a name/value pair; or by modifying the tag directly in the Source area.

Each of the interface elements of the Lasso Tag Editor screen are described below.

- Parameter Name/Value Pairs** – Each parameter in a Lasso tag is represented by a single name/value pair, which represents a parameter name and value. When editing parameter names and values, the pair will be inserted separated by an equals sign in the finished tag, e.g. `-Database='Contacts'`. For more information on parameter types and usage, see the *Lasso 8 Tag Language* chapter in the Lasso 8 Language Guide.

Note: If a parameter is represented by a single keyword (e.g. `-Search`) or single value, then it will be shown in the Value field.

Proper Lasso syntax guidelines must be followed when inserting a name and value parameter information. If the value should be surrounded by single quotes—as all literal strings in Lasso syntax should be—then the single quotes must be typed in the value field. Similarly, sub-tags should be surrounded by parentheses. If there are more than 10 parameters in the tag which is being edited, the remaining parameters will appear in the Source text area at the bottom of the dialog

- **Pencil Icon** – Each parameter name/value pair has a pencil icon to the right. Selecting this icon brings up the Parameter Editor screen which allows you to select from all the available options for the Name and Value fields. Using the Parameter Editor ensures that the syntax of the sub-tags and values in the name and value fields are correct.
- **Plus Icon** – Selecting the Plus icon creates a new parameter name/value pair above a current parameter name/value pair. This is useful for inserting parameters in specific locations when parameter order is important to the tag. This does not add new lines to the Tag Editor screen, and any parameters contained in the tenth parameter name/value pair will no longer be displayed in the name/value fields but in the Code view.
- **Source** – The source code of the tag can be modified directly in the Source text area at the bottom of the screen. Selecting the Apply button repopulates the parameter name/value pairs with any changes made to the tag in the Source text area, including changes in parameter order.

All changes are applied to the Lasso source code only upon selecting the OK button. Selecting the Cancel button reverts the tag to the way it was in the Lasso Inspector panel before selecting the Tag Editor.. button.

Parameter Editor

Selecting the pencil to the right of a parameter name/value pair in the Lasso Tag Editor screen or Lasso Inspector panel will bring up the Parameter Editor screen. The Parameter Editor screen allows you to edit the name and value of a single parameter in a Lasso tag.

The values available in the Parameter Editor are specific to the tag being edited. See the Lasso Reference or Lasso 8 Language Guide for more information on how to use parameters specific to a tag.

Note: Successfully using the Parameter Editor interface requires knowledge of parameter usage in the Lasso language. Be sure to see the *Lasso 8 Tag Language* chapter and the documentation for the specific tag being used for information on using parameters.

Figure 2: Lasso Parameter Editor

The screenshot shows the 'Lasso Parameter Editor' dialog box. It is organized into three sections: 'Name', 'Value', and 'Source'.
 - **Name Section:** Features a collapsed header with a downward arrow. Below it are six radio button options: 'Field Name' (selected), 'Variable', 'Tag Selector', 'Parameter', 'Literal', and 'No Name'. Each option is followed by a text input field. 'Field Name' has 'ID' in the field. 'Variable' has 'Variable' in the field. 'Parameter' has '-Subject' in the field.
 - **Value Section:** Features six radio button options: 'Field Value', 'Variable', 'Tag Selector', 'Keyword', 'File Name', and 'Literal'. 'Field Value' has '(Field:'ID'' in the field. 'Literal' has ''Email Subject'' in the field.
 - **Source Section:** A text area containing the text '-Subject='Email Subject'.
 - **Buttons:** 'Cancel' and 'OK' buttons are located at the bottom right.

The interface elements of the Parameter Editor screen are described below.

- **Name** – The name of the parameter being edited. In Lasso code, the name of a parameter is always followed by an equals-sign and a value. The options for parameter names are shown below.

Display Note: The Name section of the Lasso Parameter Editor can be collapsed into display-only mode by selecting the triangle to the left. The Name section is display-only by default for parameters that are value-only.

Field Value – Uses a field name as the parameter name. The pull-down menu contains a list of fields currently available in the Database Selector. In an [Inline] tag, this will insert the field name surrounded in single-quotes as the parameter name (e.g. 'First_Name') per [Inline] tag conventions. Otherwise, a [Field] sub-tag will be inserted (e.g. (Field:'First_Name') or Field('First_Name')).

Variable – Uses a variable as the parameter name. The pull-down menu allows you to select the variable tag to be used, and the field to the right allows you to input the name of the variable you wish to use. The

[Action_Param], [Cookie], [Field], and [Token_Value] tags may also be selected from the pull-down menu.

Tag Selector – Uses a defined substitution Lasso tag as the parameter name. Selecting the **Select** button launches the **Tag Selector** screen allowing a substitution tag to be selected. This is described in the *Tag Selector* section of this chapter.

Parameter – Uses a keyword parameter as the parameter name. The pull-down menu contains a list of keyword parameters available for the current tag (e.g. -Subject for [Email_Send]). Note that not all tags use keyword parameters.

Literal – Uses a literal value as the parameter name. The literal value may be entered in the field provided. True literal values should be surrounded in single-quotes (e.g. 'This is a literal text string'), however any name that is entered here will be interpreted in the Lasso code how it is entered. This means that any desired Lasso syntax may be entered here, bypassing the other radio button options.

No Name (Value Only) – Specifies that a parameter name is not used for this parameter.

- **Value** – The value of the parameter being edited. Values may be solitary, or they may be preceded by a parameter name and an equals-sign depending on the parameter being edited. The options for parameter values are shown below.

Field Value – Uses a field name as the parameter value. The pull-down menu contains a list of fields currently available in the Database Selector. This inserts a [Field] sub-tag for the field name selected (e.g. (Field:'First_Name')).

Variable – Uses a variable as the parameter value. The pull-down menu allows you to select the variable tag to be used (e.g. [Var], [Variable], \$), and the field to the right allows you to input the name of the variable you wish to use. The [Action_Param], [Cookie], [Field], and [Token_Value] tags may also be selected from the pull-down menu.

Tag Selector – Uses a defined substitution Lasso tag as the parameter value. Selecting the **Select** button launches the **Tag Selector** screen allowing a substitution tag to be selected. This is described in the *Tag Selector* section of this chapter.

Keyword – Uses a single keyword parameter as the parameter value. The pull-down menu contains a list of keyword parameters available for the current tag (e.g. -Search for [Inline]). Note that not all tags use keyword parameters.

File Name – Uses a file name as a parameter value. This is used for tags such as [Include] and [File_Copy] which require file names and paths as a

parameter value. A file name and path may be inserted manually in the provided text field (e.g. /Folder/Filename.html), or the file name and path may be entered by selecting the Folder icon to the right and pointing the parameter editor to the file you wish to use.

Note: If the Folder icon is used to specify a file path, you may wish to edit the resulting value to conform to where the referenced file will exist on your Web server. For example, if a value of File:///Documents/User/MyFolder/File.html is inserted, you would need to change it to /Folder/File.html if MyFolder is in the root folder of the Web server.

Literal – Uses a literal value as the parameter name. The literal value may be entered in the field provided. True literal values should be surrounded in single-quotes (e.g. 'This is a literal text string value'), however any value that is entered here will be interpreted in the Lasso code how it is entered. This means that any desired Lasso syntax may be entered here, bypassing the other radio button options.

- **Source** – The Source text area shows the finalized parameter as it will appear in the Lasso source code. The parameter may be edited here, and all changes made will be applied to the Name and Value section above. The Parameter Editor screen will display name and value changes in the Literal field of each section if it is not able to determine what each part of the parameter is suppose to represent.

Using the Parameter Editor

This section describes the basic procedure for using the Parameter Editor. Specifics for this procedure will vary, as there are over 1100 unique parameters in over 600 Lasso tags. Refer to the Lasso Reference or Lasso 8 Language Guide for information on handling parameters for specific tags.

To edit a Lasso tag parameter:

- 1 In the Lasso Inspector or Lasso Tag Editor, select the pencil icon next to the parameter name/value pair you wish to edit. This displays the Parameter Editor screen.
- 2 In the Name section, select a radio button for the parameter name type you wish to use, as described earlier.
- 3 Next to the selected parameter name type, enter or select a parameter name. This will appear in the Source text area below on the left side of the equals sign (=).
- 4 In the Value section, select a radio button for the parameter name type you wish to use, as described earlier.

- 5 Next to the selected parameter value type, enter or select a parameter name. This will appear in the Source text area below on the right side of the equals sign (=).
- 6 Review the finalized parameter in the Source text area to ensure it is in the correct format shown in the documentation for that tag.
- 7 Select the OK button.

This will apply the changes to the Lasso code. Select the Cancel button to revert the parameter to the way it was before selecting the pencil icon.

Tag Selector

The Tag Selector screen is called in the Parameter Editor by selecting the Select button next to the Tag Selector option, and is used to select a single Lasso tag. The tag will be formatted using either colon or parentheses syntax according to the global syntax preference. The preference can be adjusted in the Lasso Studio Panel.

Figure 3: Lasso Tag Selector



The interface elements of the Tag Selector screen are described below.

- **Category** – Specifies the Lasso Object category in which the desired Lasso tag resides. This can be Data Access or Programming. For more information on these categories, see the *Object and Tags* chapter.
- **Sub-Category** – Specifies the Lasso Object sub-category in which the desired Lasso tag resides. The sub-categories contain the same groups of Lasso tags represented by individual Lasso Objects. For more information, see the *Object and Tags* chapter.
- **Tag** – Specifies the Lasso tag to be used. The properties and parameters of this tag may be edited in the Source text area below.

- **Source** – Allows the selected Lasso tag to be viewed and edited as it will appear in the source. Other than viewing the tag in the Code view of Dreamweaver, this is the only way to edit sub-tags in Lasso Studio.

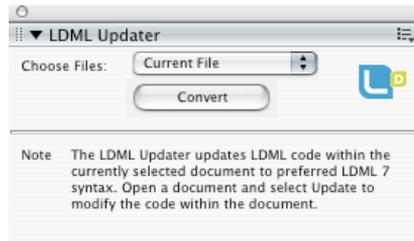
Using the Tag Selector

This section describes the basic procedure for using the Tag Selector. Refer to the Lasso Reference or Lasso 8 Language Guide for information on the tags and tag categories selected.

To select a Lasso tag as a parameter:

- 1 In the Parameter Editor, select the Select button next to Tag Selector in the Name or Value section. This displays the Tag Selector screen.
- 2 In the Category pull-down menu, select the Lasso tag category from which you wish to select a tag.
- 3 In the Sub-Category pull-down menu, select the Lasso tag sub-category from which you wish to select a tag.
- 4 In the Tag pull-down menu, select the Lasso tag you wish to use.
- 5 Edit the selected tag in the Source text area to ensure it is in the correct format shown in the documentation for that tag.
- 6 Select the OK button.

This will apply the changes to the parameter editor. Select the Cancel button to revert the parameter name or value to the way it was before selecting the pencil icon using the Tag Selector.

Figure 1: Lasso Updater

The Lasso Updater is accessed by selecting Show Lasso Updater from the Lasso Studio menu. The Lasso Updater panel has a popup menu that allows a folder to be selected or the current open document to be updated.

When the Update button is selected, the selected folder or the frontmost document in Dreamweaver is updated, and feedback about the conversion is provided in the Lasso Updater panel.

To update a page to Lasso 8 syntax:

- 1 In Dreamweaver, open a Web page you wish to update from an earlier version of Lasso to Lasso 8 syntax.
- 2 With the Web page in the foreground of Dreamweaver, select Show Lasso Updater from the Lasso Studio menu. This will show the Lasso Updater panel.
- 3 In the Lasso Updater panel, select the Current File option from the popup menu and click the Update button. This automatically updates all Lasso code in the page to Lasso 8 syntax.

To update a folder to Lasso 8 syntax:

- 1 Select Show Lasso Updater from the Lasso Studio menu. This will show the Lasso Updater panel.
- 2 Select the Select a Folder... option from the popup menu. A dialog box will appear which allows you to choose which folder you would like to update.
- 3 Click the Update button. This automatically updates all Lasso code in every file within the selected folder to Lasso 8 syntax.

After updating a page or folder, be sure to read through the feedback provided in the Lasso Updater panel for important information about tags which could not be updated or could only be partially updated.

Important: Check and test all Lasso pages after running the Lasso Updater. All attempts have been made to ensure that the code modified by the Lasso Updater will be valid Lasso syntax and will respect the original intent of the programmer.

Using FileMaker CDML Converter

The FileMaker CDML Converter updates custom Web publishing CDML Lasso pages from FileMaker solutions to Lasso 8 syntax. All CDML tags—including those in form inputs and square brackets—are updated to the newest syntax. CDML conversions include the following.

- CDML tag names are updated to their Lasso equivalents.
- Tag parameters and keywords are updated to begin with hyphens.
- Field names, related field references, and literal values in CDML tags are surrounded by single-quotes.
- Double quotes in tags are changed to single quotes.
- The operators used in conditional tags are updated to their Lasso equivalents.

Figure 2: FileMaker CDML Converter



The FileMaker CDML Converter is accessed by selecting Show FileMaker CDML Converter from the Lasso Studio menu. The FileMaker CDML Converter panel has a popup menu that allows a folder to be selected or the current open document to be updated.

When the Convert button is selected, the selected folder or the frontmost document in Dreamweaver is converted, and feedback about the conversion is provided in the FileMaker CDML Converter panel.

To convert a CDML page to Lasso 8 syntax:

- 1 In Dreamweaver, open a Web page you wish to update from a FileMaker custom Web publishing solution to Lasso 8 syntax.
- 2 With the Web page in the foreground of Dreamweaver, select Show FileMaker CDML Converter from the Lasso Studio menu. This will show the FileMaker CDML Converter panel.

- 3 In the FileMaker CDML Converter panel, select the Current File option from the popup menu and click the Converter button. This automatically converts all CDML code in the page to Lasso 8 syntax.

To convert a folder to Lasso 8 syntax:

- 1 Select Show FileMaker CDML Converter from the Lasso Studio menu. This will show the FileMaker CDML Converter panel.
- 2 Select the Select a Folder.. option from the popup menu. A dialog box will appear which allows you to choose which folder you would like to convert.
- 3 Click the Convert button. This automatically updates all CDML code in every file within the selected folder to Lasso 8 syntax.

After converting a page or folder, be sure to read through the feedback provided in the FileMaker CDML Converter panel for important information about tags which could not be updated or could only be partially updated.

Important: Check and test all Lasso pages after running the FileMaker CDML Converter. All attempts have been made to ensure that the code modified by the FileMaker CDML Converter will be valid Lasso syntax and will respect the original intent of the programmer.

12

Chapter 12

Using Lasso Shell

This chapter describes the Lasso Shell. This panel can be used to send Lasso code to Lasso Service and see the results.

Lasso Shell

The Lasso Shell panel allows Lasso code to be checked for syntax errors or executed in Lasso Professional. This tool can be invaluable for trying out code snippets or for helping find errors in Lasso code.

- **Check Syntax** – Lasso will check the syntax of code pasted into the source text area. If there are any problems, one or more errors will be reported in the results area below.

The Lasso Shell can be used to check the syntax of a short code snippet typed directly into the panel or can be used to check the syntax of an entire page pasted into the text area.

- **Execute Code** – Lasso will execute the code pasted into the source text area. The results of the code will be returned in the results area below.

The Lasso Shell can be used to quickly evaluate expressions to confirm that they work as expected, as a calculator which uses Lasso syntax to create calculations, or to execute any Lasso commands in Lasso Professional including inlines, sending email, writing files, etc.

Security

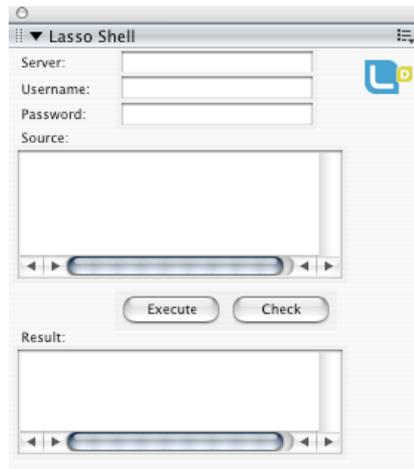
Since the Lasso Shell is executing code in Lasso Professional it is essential that only trusted users be able to submit code for processing. The Lasso Shell requires that the username and password of either the site admin-

istrator or a member of the LassoStudio group be entered into the palette before it will execute any code. See the *Configuring Lasso Professional 8* chapter for details about how to set up a LassoStudio group.

Interface

The interface of the Lasso Shell is shown below and then described in detail.

Figure 1: Lasso Shell



The top three fields allow the IP address or hostname of the server with Lasso Service to be entered and a username and password to access that server. The server must have Lasso Professional 8 installed with the latest version of LassoStudio.LassoApp. The username and password define what databases and permissions the code will be executed with.

The Source textarea allows Lasso code to be entered. This code will be either checked for errors if the Check button is selected or executed on Lasso Service if the Execute button is selected.

The code should be entered with square brackets or `<?LassoScript ... ?>` delimiters. For example, either of the following code samples would work:

```
[Var: 'Total' = 0]
[Loop: 10]
  [$Total += Loop_Count]
[/Loop]
[$Total]
```

```

<?LassoScript
  Var('Total' = 0);
  Loop(10);
    $Total += Loop_Count;
  /Loop;
  $Total;
?>

```

You can also mix HTML and Lasso code. The result will include both the HTML and the results of processing the Lasso code.

To check the syntax of Lasso code:

- 1 Open the Lasso Shell by selecting Show Lasso Shell from the Lasso Studio menu.
- 2 Enter the hostname or IP address of a machine running Lasso Professional 8. Enter a username and password to access the machine (either the Site Administrator username and password or any user defined in Lasso Site Administration).
- 3 Enter the Lasso code that you want to check into the Source textarea.
- 4 Select the Check button. The results of the syntax check will be displayed in the Results textarea below.

To execute Lasso code:

- 1 Open the Lasso Shell by selecting Show Lasso Shell from the Lasso Studio menu.
- 2 Enter the hostname or IP address of a machine running Lasso Professional 8. Enter a username and password to access the machine (either the Site Administrator username and password or any user defined in Lasso Site Administration).
- 3 Enter the Lasso code that you want to execute into the Source textarea.
- 4 Select the Execute button. The results of the Lasso code will be displayed in the Results textarea below. If any syntax errors are found in the code then an error message will be displayed instead.



Appendix A

License Agreement

PLEASE READ THIS LICENSE CAREFULLY BEFORE INSTALLING OR USING LASSO SOFTWARE. BY DOWNLOADING, INSTALLING, AND/OR USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, YOU ARE NOT AUTHORIZED TO DOWNLOAD, INSTALL, AND/OR USE THIS SOFTWARE.

1 License – The Lasso products, applications, components, modules, application programming interfaces (APIs), and programming language including Lasso Studio for Dreamweaver, Lasso Service, Lasso data source connectors, Lasso Web server connectors, Lasso modules, Lasso Site Administration interface, and Lasso Dynamic Markup Language (Lasso), whether on disk, in read only memory, or on any other media or networked storage device (the “Lasso Software”) and related documentation (the “data”) are licensed to you by OmniPilot Software (the “Author”). You own the disk on which the Lasso Software and data are recorded but the Author and/or the Author’s Licenser(s) retain title to the Lasso Software and related documentation.

This License allows you to install and use a single copy of the Lasso Software and data on a single computer and make one copy of the Lasso Software and data in machine-readable form for backup purposes only. You must reproduce on such copy the Author’s copyright notice and any other proprietary legends that were on the original copy of the Lasso Software and related documentation. You may transfer all your license rights in the Lasso Software and related documentation, the backup copy of the Lasso Software and related documentation, and copy of this License to another party, provided the other party reads and agrees to accept the terms and conditions of this License and a transfer of owner-

ship letter, listing contact information for both parties, is sent to the Author.

2 Upgrading – In order to facilitate porting existing solutions to the current version of the Lasso Software, this License allows the prior version and the current version of the software to be run concurrently for a maximum of 45 days from the purchase date of the Lasso Software.

3 Evaluation Versions – Evaluation versions are provided for one-time 30-day evaluation and initial product testing use. Evaluation versions are not licensed for use for extended development.

4 Restrictions – The Lasso Software and related documentation contains copyrighted material, trade secrets and other proprietary material and in order to protect them you may not decompose, reverse engineer, disassemble or otherwise reduce the Lasso Software to a human-perceivable form without the Author’s written permission.

5 Termination – This License is effective until terminated. You may terminate this License at any time by destroying the Lasso Software and related documentation and all copies thereof. This License will terminate immediately without notice from the Author if you fail to comply with any provision of this License. Subject to the terms of the Upgrading section above, this License will terminate upon upgrading, whether via free or paid upgrade, to a newer version of the Lasso software. Upon termination you must destroy the Lasso Software, and related documentation and all copies thereof.

6 Disclaimer of Warranty on Lasso Software – You expressly acknowledge and agree that use of the Lasso Software and related documentation is at your sole risk. The Lasso Software and related documentation are provided AS IS and without warranty of any kind and the Author EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHOR DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE LASSO SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE LASSO SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE LASSO SOFTWARE WILL BE CORRECTED.

FURTHERMORE, THE AUTHOR DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE LASSO SOFTWARE AND DATA OR RELATED DOCUMENTATION IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY THE AUTHOR OR A REPRESENTATIVE AUTHORIZED BY THE AUTHOR SHALL CREATE

A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. SHOULD THE LASSO SOFTWARE PROVE DEFECTIVE, YOU (AND NOT THE AUTHOR OR A REPRESENTATIVE AUTHORIZED BY THE AUTHOR) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

7 Limitation of Liability – UNDER NO CIRCUMSTANCES INCLUDING NEGLIGENCE, SHALL THE AUTHOR BE LIABLE FOR ANY INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES THAT RESULT FROM THE USE OR INABILITY TO USE THE LASSO SOFTWARE OR RELATED DOCUMENTATION AND DATA, EVEN IF THE AUTHOR OR AN AUTHORIZED REPRESENTATIVE OF THE AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU. In no event shall the Author's total liability to you for all damages, losses, and causes of action (whether in contract, tort (including negligence) or otherwise) exceed the amount paid by you for the Lasso Software and data.

8 Complete Agreement – This License constitutes the entire agreement between the parties with respect to the use of the Lasso Software, related documentation and data, and supersedes all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter. No amendment to or modification of this License will be binding unless in writing and signed by the Author or a duly authorized representative of the Author.

B

Appendix B

Glossary

This appendix contains definitions for many terms used in this book. Please note that many of these terms appear prepended with “Lasso” in the text of this book or in the interface of Lasso Studio.

If a term you are looking for does not appear in this glossary, see also the *Glossary* appendix in the Lasso Professional 8 Setup Guide.

Authoring Environment

The collective name for the components which are installed into the Web authoring program and the Web authoring program itself. An authoring environment is an installation of only the Lasso Studio extensions for Dreamweaver. This allows you to create Lasso-compatible Web pages within Dreamweaver, but does not allow you to create new snapshot files or to test your data-driven Web pages with live data from your database.

See also *Server Environment*.

Command Tags

Lasso command tags appear in HTML form inputs, hidden inputs, or in [Inline] tags. They define the parameters of a database action, and their names are always preceded by a hyphen.

Some command tags include:

-Search	-Add
-SQL	-Update
-Delete	-Table
-Keyfield	-Required
-AnyError	-SortOrder

See the *Using Objects and Tags* chapter for more information.

Connection

A connection is communication between a Lasso Studio authoring environment and a local or remote Lasso Professional server environment. A default connection is made to the local Lasso Professional server environment installed on the same machine as the Lasso Studio authoring environment.

See also *Online, Offline, Authoring Environment, Server Environment*.

Container Tags

Container tags are delimited by square brackets and are comprised of both an opening and a closing tag. Much like in HTML markup, the area between the opening and closing tags is modified by the container tag. Container tags in Lasso perform database actions, hide code from the Web browser, perform conditional logic, and provide error control. Some container tags include:

```
[File_Control] [/File_Control]    [If] [Else] [/If]
[Inline] [/Inline]                [Loop] [/Loop]
[While] [/While]
```

See the *Using Objects and Tags* chapter for more information.

Data Access Objects

The Data Access objects include Lasso tags for retrieving data from a database, performing database actions inline within a Lasso page, constructing links to related data, and retrieving information about the current database action.

See also *Form Objects* and *Programming Objects*.

Data Preview

Feature in the Lasso Studio authoring environment that replaces [Field] tags in the current Lasso page with data from a sample record set stored in a snapshot file. Only [Field] and [Records] ... [/Records] tags are modified by Data Preview. No other tags can be previewed.

Default Installation

A default installation of Lasso Studio is where both the Lasso Studio authoring environment (e.g. Dreamweaver extensions) and included single-user license of Lasso Professional 8 (server environment) are installed on the same machine.

Form Action

In an HTML form, the form action defines how the form will be processed on the server. In order to work with Lasso, the form action needs to specify

a valid Lasso response page that will be used as the response page for the form.

```
<form action="/ResponsePage.Lasso" method="POST">
  <input type="hidden" name="-Response" value="Response.lasso">
  ...
</form>
```

Form Builder

The Form Builder inserts a Lasso-enabled form into the current page, replacing the current selection. The Form Builder can generate a form based on the current database schema for any common database action. The Form Builder can be accessed from the Lasso Studio menu.

Form Element

HTML form elements include button, check boxes, inputs, pop-up menus, radio buttons, select lists and text areas. Lasso-specific attributes can be added to form elements using the Lasso Form Inspector.

Form Inspector

The Lasso Form Inspector allows Lasso specific attributes to be added or modified in HTML form elements.

Form Objects

The Lasso Form section of the Insert panel contains HTML form elements and hidden inputs with command tags and action tags. These objects can be dragged into the Design view of the current page.

See also *Data Access Objects* and *Programming Objects*.

Initialization

A screen in Lasso Studio which allows the Lasso Studio user to enter the product serial number, set up the initial connection to the Lasso Professional server environment, and create a snapshot file for the first time.

Inline Method

Inline-based Web sites use HTML forms and URLs simply to pass user-defined data around the Web site. All database actions are performed using [Action_Params] and [Inline] tags. Using [Inline] tags in a Web site affords great flexibility and security when creating data-driven Web sites. For more information, see the *Database Interaction Fundamentals* chapter in the Lasso 8 Language Guide.

Lasso 8

The product line produced by OmniPilot Software used to build and serve data-driven Web sites. This includes Lasso Professional 8, Lasso Studio 8, and Lasso Developer 6.

Lasso Inspector

The floating panel in Dreamweaver which provides information about the current selection if it is a Lasso Object. Includes the Tag Inspector, Form Inspector, and LassoScript Inspector.

Lasso Professional 8

The Lasso 8 product used to serve data-driven Web sites. Lasso Studio 8 includes a single-user license of Lasso Professional 8 for testing data-driven Web sites.

See also *Single-User* and *Multi-User*.

Lasso Studio 8

A Lasso 8 product used to build and test data-driven Web sites to be served by Lasso Professional 8.

Lasso Studio Extensions for Dreamweaver

The Lasso Studio extensions for Dreamweaver are the set of plug-ins and modules which enable Lasso-based Web sites to be easily authored in Macromedia Dreamweaver, and constitute the authoring environment in Lasso Studio. These include all of the objects, inspectors, menu commands, floating panels and dialog boxes described in this manual.

Lasso Studio 8 for Adobe GoLive CS

The edition of Lasso Studio which provides support for Adobe GoLive CS. Lasso Studio 8 for Adobe GoLive CS transforms Adobe GoLive into an authoring environment for dynamic, data-driven Web sites. It consists of two components: the Lasso Studio Extensions for GoLive (authoring environment) and single-user version of Lasso Professional 8 (server environment).

Lasso Studio 8 for Dreamweaver

The edition of Lasso Studio which provides support for Macromedia Dreamweaver. Lasso Studio 8 for Dreamweaver transforms Dreamweaver into an authoring environment for dynamic, data-driven Web sites. It consists of two components: the Lasso Studio Extensions for Dreamweaver (authoring environment) and single-user version of Lasso Professional 8 (server environment)

Lasso Studio Menu

The menu in the Dreamweaver menu bar which provides access to each of the components of the Lasso Studio authoring environment.

Lasso Studio Panel

The floating panel in Dreamweaver which provides access to the Database Selector, Data Preview, and Preferences control panels in Lasso Studio.

Lasso

Lasso is an abbreviation for Lasso Dynamic Markup Language. Lasso is Lasso's tag-based markup language. It consists of container, process, member, and substitution tags which are delimited by square brackets and are processed when a Lasso-based page is served by your Web Server. Lasso also consists of command and action tags which are preceded by a hyphen. Command and action tags are contained in HTML form inputs or URLs and are processed when the form is submitted or URL is loaded from your Web server. LassoScript is an alternate form of Lasso syntax that is not delimited by square brackets.

Lasso Updater

The Lasso Updater brings the Lasso syntax of a Lasso page created with previous versions of Lasso up-to-date with the newest syntax.

Local Server Environment

The local server environment consists of Lasso Professional 8 working with a Web server installed on the machine on which Lasso Studio is installed (e.g. where Dreamweaver or GoLive is installed).

Multi-User

Allows Lasso solutions to be served to unlimited IP addresses (unlimited users). The single-user version of Lasso Professional 8 provided with Lasso Studio allows the developer to test and preview solutions in a server environment, but does not allow solutions to be used or viewed by more than one user. The full version of Lasso Professional 8 is a multi-user version, and is a separate purchase from Lasso Studio 8.

See also *Lasso Professional 8* and *Single-User*.

Offline

When the authoring environment connection is offline, Lasso Studio will not attempt to access the server environment, but will work from database information and record sets stored in the current snapshot file.

Online

When the authoring environment connection is online, Lasso Studio will attempt to access the server environment defined by the current snapshot file in order to retrieve any database information or record sets which are not already stored in the current snapshot file.

Parameter Editor

The Parameter Editor screen allows for a single parameter in a complex tag to be edited. The screen provides access to the most common tags and values so they can be easily inserted. The Parameter Editor can only be accessed through the Tag Editor.

See also *Tag Editor*.

Process Tags

Process tags are delimited by square brackets and appear much like substitution tags except that they do not return any value to the page which will be shown to the user. Process tags include the following:

[Email_Send]	[HTML_Comment]
[Log]	[Session_Start]
[Variable_Set]	

See also *Container Tags* and *Substitution Tags*.

Programming Objects

The Lasso Programming section of the Insert panel contains tags which perform programming logic within your Lasso page. These include math tags, string manipulation tags, date tags, list tags, file control tags, and others.

Record Set

A record set is a collection of records that defines sample data for use in Data Preview for a particular database and table. A default record set which consists of the first 10 records from a database and table is created automatically if none is stored. Record sets only store values for non-hidden fields from the current snapshot.

Remote Server Environment

A remote server environment consists of Lasso Professional 8 installed on a different machine than that on which the authoring environment is installed. A remote server environment may be at an ISP or a shared Web server.

Response Page

The Lasso page which Lasso displays to the user after a successful action. The response page can be specified within the form action of a preceding form. The Site Builder automatically creates the appropriate response pages for each database action.

See also *Form Action*.

Server Environment

The collective name for the services which allow Lasso pages to be processed through Lasso. Includes the Web server and Lasso Professional 8. A server environment connection is required for the data access features in the Lasso Studio authoring environment to function.

Lasso Studio 8 includes a single-user license of Lasso Professional 8 that allows you to test Lasso solutions and provides the Lasso Studio authoring environment with information about available databases. This feature operates separately from Dreamweaver or GoLive, and may be installed on the same machine or a different machine from the authoring environment. Lasso Studio may connect either to this single-user version, or to any full version of Lasso Professional 8.

See also *Authoring Environment*, *Single-User*, and *Multi-User*.

Single-User

A single-user version of Lasso Professional 8 allows Lasso solutions to be served only to one IP address (one user). The single-user version of Lasso Professional 8 allows the developer to test and preview solutions in a server environment, but does not allow solutions to be used or viewed by more than one user. A single-user version of Lasso Professional 8 may be obtained by entering a Lasso Studio 8 serial number in Lasso Professional 8, which is included with Lasso Studio.

See also *Lasso Professional 8* and *Multi-User*.

Site Builder

The Site Builder allows you to create a set of Lasso pages that perform common database actions. The pages are customized based on the current snapshot file. Either a set of pages for searching a database or a set of pages for adding a record to a database can be created.

See also *Form Builder*.

Site Builder Template

The pages generated by the Site Builder can be optionally based on a template file. This is any HTML file that includes a Site Builder Target

object. The Site Builder Target object can be inserted from the Lasso Form section of the Insert panel.

Snapshot File

A snapshot file is an XML file created by Lasso Studio which contains server connection information, database schema information, and sample record sets for a configured Lasso Professional server environment. Snapshot files provide the data-access features of the Lasso Studio authoring environment with information about eligible databases, tables, and fields. Sample record sets are stored in snapshot files for offline Data Preview.

Snapshot files may be edited via the Snapshot Editor, which allows databases, tables, and fields to be hidden to create a custom set of databases for a project. Snapshot files can be shared so multiple Web developers can work with the same database schema.

See also *Snapshot Editor* and *Data Preview*.

Snapshot Editor

A screen in Lasso Studio which allows snapshot files to be edited. This includes editing the server environment connection information, database schema information, and sample records set information in a snapshot file.

Source Editor

The Source Editor area at the bottom of the Lasso Tag Editor and Lasso Inspector panel allows for a tag to be edited manually.

Substitution Tags

Substitution tags are the most common tags in Lasso's tag-based markup language Lasso. They are delimited by square brackets and are replaced by a generated value as Lasso processes a Lasso page. For instance, the following tag will be replaced by the value for the field `Company` in the current database:

```
[Field: 'Company']
```

Substitution tags can also be used as attributes for Lasso tags. For instance, the following substitution tag adds two values, each defined by a field placeholder substitution tag:

```
[Math_Add: (Field: 'Total'), (Field: 'Shipping')]
```

See also *Container Tags* and *Process Tags*.

Tag

Lasso uses a tag-based language called Lasso. Command and action tags can be inserted into HTML form inputs, URLs, or [Inline] tags:

```
<input type="hidden" name="-Database" value="[Database_Name]">
```

Substitution, container, and process tags are surrounded by square brackets and appear within the HTML source of a Web page:

```
[Field: 'Field Name']
```

See also *Action Tags*, *Command Tags*, *Container Tags*, *Process Tags*, and *Substitution Tags*.

Tag Editor

The Tag Editor allows for editing of complex tags by breaking out each of the parameters in the tag. The Parameter Editor can be accessed for any name/value pair in the tag in order to edit any attribute of the tag in a guided environment.

Tag Inspector

The Lasso Tag Inspector is shown each time a container tag, process tag, or substitution tag is selected.

The Lasso Tag Inspector allows any of the attributes of the tag to be modified and provides access to the Tag Editor for more help modifying the attributes of the tag.

Value List

Value lists are defined in FileMaker Pro data sources. They allow for a list of commonly used values to be associated with a database field.

C

Appendix C

Index

A

Acrobat 22, 23
 Action Object 162
 Action Param Object 152
 Add.Lasso 137
 Adding Records 107
 AddResponse.Lasso 139
 Administration Object 157
 AnyUser Group 67
 Apache 22
 Application Files 32
 Array Object 157
 Authoring Environment 10, 213

B

Balance 178
 Blue World Support Central 18
 BSD 22
 Button Object 162

C

Checkbox Object 163
 Client Object 153
 Code 16
 Command Tags 149, 213
 Components 10
 Conditional Logic Example 184
 Conditional Object 184
 Condition Object 157
 Connection 64, 78, 214
 Editing 82
 Container Tags 148, 183, 214
 Cookie 157

Cross References 16
 Customer Support 18
 Custom Groups and Users 68
 Custom Tag Object 158

D

Database 78, 85
 DatabaseBrowser.LassoApp 42
 Database Object 153, 163
 Database Selector 59, 77
 Database Snapshots 73
 Data Access Objects 214
 Data Preview 11, 59, 73, 87, 214
 Settings 89
 Usage 91
 Data Source Configuration 44
 Date Example 180
 Date LDML Object 180
 Date Object 158
 Default Installation 214
 Definitions 16
 Deleting Records 111
 Deployment 22, 66
 Display.Lasso 135
 Documentation 15, 18
 Documentation Conventions 16
 Documentation Folder 32
 Drag-and-Drop 11

E

Editors 63
 Email Example 182
 Email LDML Object 182
 Email Object 153

Encoding Object 154
 Encryption Object 158
 Error Object 158
 Evaluation 19, 28
 Extensions Folder 32, 33

F

Features 11
 Field 86
 Fields
 Editing Visual Properties 91
 Field Object 154
 FileMaker Pro 44
 File Object 159
 File Paths 17
 Form Action 214
 Form Builder 11, 60, 95, 215
 Creating Add Forms 107
 Creating Add Response Pages 109
 Creating Record Display Pages 105
 Creating Record Listing Pages 103
 Creating Search Forms 101
 Creating Standalone Listing and Display Pages 106
 Creating Update Forms 111
 Creating Update Response Pages 113
 Tips 100
 Usage 97
 Form Builder Palette 97, 98
 Form Elements 188, 215
 Form Inspector 215
 Form Objects 163, 215
 Full Expression 173

G

Getting Started 31
 Glossary 213

I

Icon 172
 Image Object 154
 Include Object 154
 Initialization 27, 215
 Inline Example 184
 Inline Method 96, 215
 Inline Object 154
 Inline Object Example 166
 Inspectors 11
 Installation 21
 Mac OS X 25
 Prerequisites 23

Windows 2000/XP 26
 Installation Contents 32
 Installing Lasso Professional 6 25
 Introduction 9

J

Java Virtual Machine 23
 JDBC Data Sources 49
 JRE 23

K

Key Field 78

L

Lasso 216
 LassoModules Folder 33, 34
 LassoScript 151, 159, 189
 LassoScript/Expression Inspector 176
 Usage 179
 LassoScript Object 189
 LassoTraining.com 71
 Lasso Administration 58
 Accessing 40
 Lasso Connector for Apache 14
 Lasso Connector for FileMaker Pro 14
 Lasso Connector for JDBC 14
 Lasso Connector for MySQL 14
 Lasso Database Browser 42
 Lasso Data Access Objects 152
 Lasso Expression 159
 Lasso Expressions 189
 Lasso Expression Object 190
 Lasso Form Builder Palette 60
 Lasso Form Inspector 176
 Usage 178
 Lasso Form Objects 162
 Lasso Inspector 60, 175, 216
 Usage 176
 Lasso Language Guide 15, 70
 Lasso Product Line 13
 Lasso Professional 10, 13, 14, 216
 Usage 39
 Lasso Professional Files 34
 Lasso Professional Setup Guide 15, 71
 Lasso Programming Objects 156
 Lasso Security 40, 66
 Lasso Service 10, 13
 Lasso Studio 9, 14, 55, 216
 Interface Elements 56
 Usage 55

- Lasso Studio Documentation 15
- Lasso Studio Extensions for GoLive 216
- Lasso Studio Folder 32
- Lasso Studio for Adobe GoLive 6 216
- Lasso Studio for Dreamweaver MX 216
- Lasso Studio Menu 57, 217
- Lasso Studio Palette 58, 217
- Lasso Studio User Guide 15
- Lasso Tag Inspector 176
- Usage 177
- Lasso Talk Mailing List 18, 71
- LDML 217
- LDML Display Preferences 172
- LDML Objects 61, 145
- LDML Reference 15, 70
- LDML Tags 17
- LDML Updater 12, 61, 201, 217
- Usage 201
- Learning Resources 70
- License Agreement 209
- Link Object 155
- Listing.Lasso 134
- Local Server Environment 217

M

- Mac OS X 22, 23
- Math Object 159
- Member Tags 150, 186
- Multi-User 11, 217
- MySQL 47

N

- Name/Value Pairs 194
- Name Field 196
- Navigation 90
- New Features 12

O

- Objects 145
- Offline 217
- Online 218
- Operator Object 163
- Output Object 155

P

- Palettes 58
- Parameter Checkboxes 177
- Parameter Editor 195, 218
- Usage 198

- Parameter Name/Value Pairs 194
- PDF Object 160
- Pencil Icon 195
- Plus Icon 195
- Preferences 59
- Process Tags 148, 180, 218
- Programming Objects 218

Q

- Query Field 92
- Query Value 93

R

- Radio Button 164
- Record Set 64, 90, 218
- Record Sets
- Editing 87, 92
- Viewing 91
- Refresh 79, 86
- Remote Server Environment 218
- Response Object 164
- Response Page 219
- Results Object 155
- Running Lasso Professional 6 40

S

- Schema 64
- Editing 84
- Screens 63
- Script Object 164
- Search.Lasso 132
- Searching Records 100
- Select List Object 164
- Serial Number 28
- Server Connection 11
- Server Environment 10, 219
- Server Environment Setup 39
- Serving Pages With LDML Tags 96
- Session Object 160
- Single-User 10, 219
- Site Builder 11, 64, 115, 219
- Creating an Add Site 125
- Creating a Search Site 118
- Interface 116
- Site Types 117
- Testing Your Site 141
- Site Builder Pages 132
- Site Builder Target Object 165
- Site Builder Template 219
- Site Builder Templates 142

Site Customization 118
 Snapshots 11
 Copying 81
 Creating 75
 Editing 82
 Loading 81
 Updating 80
 Using 77
 Snapshot Editor 63, 220
 Snapshot Files 73, 220
 Source Editor 178, 195, 198, 200, 220
 String Object 160
 String Size Member Tag Example 186
 Substitution Tags 147, 180, 220
 Support 18
 System Requirements 21

T

Table 78, 85
 Tag 177, 221
 Tag Editor 65, 193, 221
 Usage 194
 Tag Inspector 221
 Tag Name 173
 Tag Selector 199
 Usage 200
 Technical Object 161
 Testing Solutions 56, 66, 69
 Text Input Object 165
 Token Example 188
 Token Object 161, 165, 188
 Troubleshooting 69

U

UFS 22
 Uninstalling Lasso Studio 6 36
 Update.Lasso 138
 UpdateResponse.Lasso 140
 Updating Records 111
 Upgrading
 Mac OS X 24
 Usage Rights 18
 User Configuration 50
 Using Lasso Studio 55
 Using LDML Objects 166
 Utility Object 156, 165

V

Value Field 197
 Value List 221

Variable Object 161

W

What's New in 6.0 12
 Windows 2000 22
 Windows XP 22
 Writing LDML 166

X

XML 73
 XML Object 162