

MODX Revolution Building the Web Your Way

A Journey Through A Content Management Framework

W. Shawn Wilkerson

MODX Revolution Building the Web Your Way: A Journey Through a Content Management Framework

Published by Sanity Press, a Sanity LLC company (sanitypress.com), Daytona Beach, FL 32117. Copyright © 2012 Sanity LLC All rights reserved. The editorial arrangement, analysis, and professional commentary are subject to this copyright notice. No portion of this book may be copied, retransmitted, re posted, duplicated, photocopied, or otherwise reproduced without the express written approval of the author, except by reviewers who may quote brief excerpts in connection with a review. Any unauthorized copying, reproduction, translation, or distribution of any part of this material without permission by the publisher is prohibited and against the law.

Warning and Disclaimer: No information contained in this book should be considered as financial, tax, business, or legal advice. Your reliance upon information and content obtained by you at or through this publication is solely at your own risk. Sanity LLC, Sanity Press nor the author assume no liability or responsibility for damage or injury to you, other persons, or property arising from any use of any product, information, idea, or instruction contained in the content or services provided to you through this book. Reliance upon information contained in this material is solely at the reader's own risk. Sanity LLC, Sanity Press, and the author makes no warranties, either express or implied, regarding the content or its fitness for any particular purpose. The authors have no financial interest in and receive no compensation from manufacturers of products or web sites mentioned in this book.

Trademarks: MODX, MODX Revolution, MODX Evolution are trademarks or registered trademarks of MODX, LLC. All other trademarks are the property of their respective owners. The publisher and the author is not associated with any product or vendor mentioned in this book.

Library of Congress Control Number: 2012914262

ISBN-13: 978-0985853204 (Sanity Press)

ISBN-10: 0985853204

Printed in the United States of America

First Printing: August 2012

07 06 05 04 4 3 2 1

Bulk Sales: Sanity Press offers discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

Sanity Press

1-386-322-7999

sales@sanitypress.com

Abbreviated Contents

Foundations	2
Introducing MODX Revolution	4
Discovering The Heart of MODX Revolution: The Manager	16
Creating Project Templates	46
[[<code>\$Chunks</code>]]:Simplifying Web Projects Using (X)HTML	58
[[<code>*templateVariables</code>]]: Building Interactive Interfaces	74
[[<code>Snippets</code>]], Plug-ins and Widgets: Adding PHP Dynamic Content	98
:filters - The Art of Manipulating Element Output	160
Live Projects	176
Quick Start: Putting the Pieces Together	178
Streamlining Web Projects With SEO, FURLs, and Minify	198
Power Blogging	222
Dynamic JavaScript, CSS, and jQuery Essentials	268
AJAX Content Injection	296
Administration	310
An Overview of Grouping: Category, User and Resource	312
Access Control List Implementation	328
Contexts: Implementing Sites with Multiple Faces	354
Development	378
MODX Revolution API Concepts	380
xPDO: Foundations For Development	420
xPDO: Third-Party Applications	452
Now What?	532
Appendices	536

Contents

Part I: Foundations

Introducing MODX Revolution	4
What is MODX Revolution?	5
Example MODX Revolution Web Projects	7
What Sets MODX Revolution Apart?	8
Simple File Structure	9
Communication Facilitation	10
Streamlined Workflow	11
Project Resources	12
My Current Endeavours	13
Discovering The Heart of MODX Revolution: The Manager	16
The Evolution of MODX Managers	16
Welcome Screen	19
The MODX Revolution Manager	20
Exploring the Menu	21
Dashboard	21
Site Menu	21
Components Menu	23
Security Menu	24
Tools Menu	29
Reports Menu	34
System Menu	37
User menu	42
Support menu	43
Creating Project Templates	46
Concerning Templates	46
MODX Revolution Templates	47
Template Inheritance	47

Naming Conventions	48
Creating a Template	49
Resource Tags	50
System Settings Tags	51
(X)HTML Tags	51
Storing Templates in the Manager	52
Text Editors	52
Example XHTML 1.0 Strict Template	53
W3C Validation	55
Template Storage Alternatives	56
Templates available via Package Management	56
Using Media Sources: Filesystem	56
[[\${Chunks}]]: Simplifying Web Projects Using (X)HTML	58
<hr/>	
Introducing (X)HTML [[\${Chunks}]]	59
Creating a Chunk	59
[[\${siteHeader}]]	60
[[\${jQuery}]]	61
[[\${siteFooter}]]	61
Chunks within Chunks	63
Using Chunks to Streamline Templates	63
Showing the Pieces Together	65
Increasing Readability	67
Using Chunks With Placeholders	68
Using Chunks with Property Sets	69
Overriding a Property Set	70
Chunks Used With Snippets	70
[[\${globalNav}]]	71
[[\${wfNavBar}]]	71
Example [[\${globalNav}]] Output	72
Test Implementations with Duplicate Elements	73
[[*templateVariables]]: Building Interactive Interfaces	74
<hr/>	
What is a Template Variable?	74
Creating Template Variables	75
Place Holder Template Variables	76
Attaching the TV to a Template	77
Understanding Input and Output	77
Selecting The Type of Template Variable Input	78

Template Variable Input Types as of 2.2.2	79
Template Variable Examples	80
Auto-tag	80
Check Box	82
Date	83
Email	84
File	85
Hidden	87
Image	88
Listbox	89
Radio Option	90
Resource List	91
Rich Text Editor	91
Tag	92
Text	93
Textarea	93
URL (Uniform Resource Locator)	94
Forcing the Use of Template Variables	94
Output Template Variables	95
Delimiter	95
HTML Tag	95
Rich Text	96
Custom Forms	96
Database considerations	96
<u>[[Snippets]], Plug-ins and Widgets: Adding PHP Dynamic Content</u>	98
Snippets	99
Why is it called a Snippet?	100
The Dangerous Side to Snippets	100
Verify Snippet Functionality and Output	101
Package Example: Wayfinder	101
Exploring the Directory Structure	102
The Wayfinder Snippet	102
When the Code is All The Documentation We Have	105
Deciphering Snippet Activity	105
Finding the Required Parameters	106
When A Snippet Exists For Evolution And Revolution	107
Snippet Calls	108
Syntax and Caching Implications	108

Use Backticks (`) For Parameters	108
Wayfinder Implementation Samples	109
Global Horizontal Navigation Bar	109
Vertical Navigation	111
Building a Wiki-type Layout	112
Included Navigation Configurations	114
Getting Ready To Build Your Own Snippet	115
Coding Advice	115
MODX Revolution Coding Expectations	117
Using A Basic Structure	118
Part 1: File and Package Information	122
Part 2: Legal Declarations	122
Part 3: Snippet Description and Parameter Documentation	123
Part 4: Functions	123
Part 5: Variable Declaration	124
Part 6: The Workhorse	125
Part 7: Return Value	125
Using the Simplest Solution	126
Adding A Snippet To The Project	128
Steps to Creating a New Snippet	128
Pasting Our Snippet Code	129
Verify Snippet Output	130
Working with Placeholders	132
Simple Example	132
Why Placeholders?	133
Working With Multiple Placeholders and a Single Snippet	134
Using Chunks To Template Snippet Output	137
Create a Snippet to Collect The Data	137
Add An Optional Property Set	141
Building Snippet Frameworks	143
Create A Chunk To Use As A Template	144
Benefits Of Chunk Templates Used For Snippet Output	145
Implementing A Templated Snippet	146
Snippet output	147
Making the Workflow and Snippet Simpler	148
Working With PHP Classes	150
\$modx->runSnippet()	153
Exotic Snippet Calls	154
Plugins	155
Plugin Example: AntiSpam	155

Anti-Spam Output	158
Assigning Priority to Plugins	158
When to use Plugins instead of Snippets	159

:filters - The Art of Manipulating Element Output **160**

Filters defined	161
The Benefits of Filters	161
Implementation Examples	162
Conditional Overrides and Targeting	162
else	162
hide	162
show	162
then	162
Compare Operators	163
eq, is, equals, equalto, isequal, isequalto	163
gt, isgt, greaterthan, isgreaterthan	163
gte, isgte, eg, ge, equalorgreaterthan, greaterthanorequalto	163
input, if	163
lt, islt, lessthan, lowerthan, islessthan, islowerthan	163
lte, islte, le, el, lessthanorequalto, equaltoorlessthan	163
ne, neq, isnot, isnt, notequals, notequalto	163
Content Injection	164
cssToHead	164
htmlToBottom	164
htmlToHead	164
jsToBottom	164
jsToHead	164
toPlaceholder	164
Date Formatting	165
ago	165
date	165
fuzzydate	165
strtotime	165
Element Filters	165
select	165
tag	165
Logic	166
and	166
or	166
Math Operations	166
add, increment, incr	166
divide, div	166
math	166
modulus, mod	166
multiply, mpy	166

subtract, decrement, decr	166
String Manipulation	167
cat	167
cdata	167
default, ifempty, isempty, empty	167
esc, escape	167
ellipsis	167
htmlent, htmlentities	168
ifnotempty, isnotempty, notempty, !empty	168
length, len, strlen	168
lcase, lowercase, strtolower	168
limit	168
md5	169
nl2br	169
notags, striptags, stripTags, strip_tags	169
replace	169
reverse, strrev	169
strip	169
stripString	170
ucase, uppercase, strtoupper	170
ucfirst	170
ucwords	170
urldecode	170
urlencode	170
wordwrap	170
wordwrapcut	170
User Filters	171
isLoggedIn	171
ismember, memberof, mo	171
isnotloggedin	171
userinfo	171
Custom Filter	172
Successful Implementation Considerations	173
Sample Each Stage	173

~~~~~

***Part II: Live Projects***

~~~~~

Quick Start: Putting the Pieces Together	178
Change System Settings (Chapter 2)	179
Add A Site Template (Chapter 3)	180
Create Chunks (Chapter 4)	181
Create Template Variables (Chapter 5)	183
Install Navigation (Chapter 6)	185

Add AntiSpam Plugin	185
Add Custom PHP Snippets	186
Edit The Home Page	187
Add Additional Resources	188
Create Subdirectories / Containers	189
Add Additional Pages To A Container	189
Thinking Forward: Structure and Project Direction	190
Adding A Contact Form	191
Adding Site Search (For Static Content Only)	196
<u>Streamlining Web Projects With SEO, FURLs, and Minify</u>	<u>198</u>
Understanding The Search Engine	199
Optimizing a Site For Search Engines	199
Configuring: Friendly URLs	200
Modify .htaccess file	200
Configure MODX Revolution To Use Friendly URLs	202
Check All Templates and Header Chunks For a <base href> Tag	203
Clear the MODX Revolution cache	203
Minified JavaScript / Cascading Style Sheets	204
Installing Minify	204
Match Settings to Server Configuration	205
Testing the Minify Implementation	208
Establishing Groups	210
Forcing JavaScript To The Page Bottom	212
Basic .htaccess Optimizations	213
Strategic Utilization of System Settings and Elements	215
SEO Benefits From the Site Structure	216
Webmaster Resources	217
Creating a Site Map	217
robots.txt	218
humans.txt	218
Dynamic Content	219
<u>Power Blogging</u>	<u>222</u>
Sample Project Requirements	223
Blogging Foundations	223
Summary of Installed Packages For “Power” Blogging	225
Creating a Blog Implementation	225
General Web Project	225

User Authentication	226
Manually Building a Blog	227
Site Templates	227
Blog Related Chunks	232
Wayfinder Related Chunks	234
Taglist Related Chunks	235
Site - Wide Chunks	235
Individual Author's Resource Page	236
Custom Snippets	236
Screen Shots Taken from 1truth.net	237
Final Notes on Creating a Blog From Scratch	239
Implementing the Articles Add-on	240
Templates	241
Template Variables	252
Creating a Chunk for Category and Other Pages	253
Creating Category Pages	254
Assign Templates to the Article Container	256
Advanced Settings	257
Advanced Settings Highlights	258
Third-Party Services	259
<u>Dynamic JavaScript, CSS, and jQuery Essentials</u>	<u>268</u>
Loading JavaScript into the Resource	269
Forcing JavaScript and HTML to the Bottom of a Resource	269
Forcing CSS, HTML and JavaScript to the Page <head>	270
Using Chunks to Create Version Specific Dynamic Links	271
Laying the JavaScript Foundation	271
Testing the Foundation	272
Implementing Site Wide JavaScript	273
Using Minify's To Combine External JavaScript Files	274
jQuery Foundational Concepts	275
Default document.ready Container	275
MODX Revolution jQuery Implementations	276
Manipulating Hyperlinks	276
Limiting The Number of List Items Displayed	277
Sortable Tables and Working With Runtime Configurations	279
Toggle the Display of Page Content by Element Class	283
Extremely Simple Lightbox	286
Wayfinder Combined With jQuery	289

flashembed()	293
Final Thoughts on Content Injection	294

AJAX Content Injection **296**

AJAX (Asynchronous JavaScript And XML)	297
To AJAX Or Not To AJAX	298
Argument For Implementing AJAX	298
Argument Resistant to AJAX	298
Component Essentials	299
An AJAX Template	299
AJAX Resources	299
Assorted AJAX Implementations	299
Manipulating Snippet Presentation: Extending The Archivist Tree	300
Using \$modx->runSnippet()	304
User Details	305

~~~~~

### ***Part III: Administration***

~~~~~

An Overview of Grouping: Category, User and Resource **312**

Element Categories	313
Resource Group Considerations	315
Adding a Resource to a Resource Group	316
User Groups	317
User Groups: A World Unto Themselves	319
Using the Access Wizard for User and Resource Group Creation	320
Adding Users To A User Group	321
User Roles	322
Policy Templates	323
Policies	325
How the Pieces Work Together	326

Access Control List Implementation **328**

Implementing The Login Package	329
User Account Considerations	331
Implementing a "Trusted User" Read-Only Area	332
Create A Resource Group	332
Create a Resource and add it to the Resource Group	333

Create a User Group	333
Create the Minimum User Role	334
Add a Site Administrator to the User Group	335
Add Additional Users	336
Assign Minimum Roles to Contexts	336
Add Resource Access	337
Verify Settings	338
When Everything is Satisfactory, Click The Save Button.	341
Force All Users to Log Back Into Site	341
Testing the Read-Only Implementation	342
Extending the User Group	343
Add New User Roles to the Project	343
Place Users Within The Newly Created Roles	344
Adjust the minimum User Role for Manager Access	344
Adjust the minimum User Role to access Resources	345
Add Associated Element Categories	345
Attach a Dashboard to the User Group	346
Example Modified Dashboard	350
Targeted Application	351
Contexts: Implementing Sites with Multiple Faces	354
<hr/>	
Laying a Foundation	355
Possible Malfunctions	356
Simple Precautions	356
Understanding Context and User Group Differences	357
Creating a New Context	358
Implementing a Context “Skeleton”	359
Example Site Skeleton Components	359
Global Site Header	361
Utilizing the Site Skeleton Context	362
Prerequisites For Domain Name Specific Contexts	363
Configuring Domain Names on Shared Hosts	365
Hosting Many Web Sites	366
.htaccess Multi-domain Modifications	366
robots.txt and humans.txt Considerations	367
Create or Duplicate a New Context	368
Activating The New Context	368
Subdomain Contexts	369
Forcing the Subdomain to the Site Root	369

.htaccess subdomain Modifications	370
Create or Duplicate a Context for the Subdomain To Use	371
Implement or Edit the domainGateway Plugin	371
domainGateway with Domains and Subdomains	372
Protecting the Manager Context	373
Examples of using Snippets and Chunks with Contexts	374

~~~~~

***Part IV: Development***

~~~~~

MODX Revolution API Concepts	380
Concerning Object Oriented Programming	381
The \$modx Object	382
Benefits of utilizing the MODX Object	382
The Dangerous Side to the MODX Revolution API	383
Verify PHP Functionality and Output	383
\$modx Object Implementation - importing the API	384
Secondary MODX Revolution Classes	385
Avoid Garbage In / Garbage Out Scenarios	386
Retrieving Values From Object Attributes	387
Object Instantiation	388
get(): Directly Accessing Object Attributes	389
toArray(): Returning PHP Arrays of Object Contents	391
toJSON(): Returning JavaScript Object Notation	392
Retrieving Multiple Related Objects	393
getOne(): Accessing a 1:1/0 Relationship	394
getMany(): Accessing a 1:n Relationship	396
Creating an Object Test Bed	398
Creating a Resource Programmatically	403
Establish a Baseline	403
Building an Implementation	405
Updating the User On-The-Fly	407
Mastering the Elements	409
The Small Print	411
All Snippets are treated as functions	411
Resources are Cached as Arrays of Elements	412
The Difference Between Cached and Non Cached Elements	417
Snippets Can Serve as Functions	418

xPDO: Foundations For Development	420
xPDO Overview	421
xPDODriver	422
xPDOGenerator	422
xPDOManager	422
xPDOObject	423
xPDOSimpleObject	423
Schemata	424
Hierarchy Structure	424
The Model	425
The Object	426
The field	427
Relationships	428
Quick Start: Creating a Single Database Lookup Table	429
Step 1: Create a Simple xPDO Schema	429
Step 2: Create a Working Directory	430
Step 3: Upload the Schema to the model/schema directory	430
Step 4: Build the xPDO Model	431
Step 5: Creating Database Table(s) using the xPDO Model	432
Step 6: Instantiate an Object of the Package	434
Step 7: Populate the Table	435
Step 8: Assign a Category (Optional)	436
Packaging Components via Packman	437
Creating a Multi-Table Schema	440
Solutions Diversify As Complexity Increases	440
Lay the Primary Tables Out First	443
Table Normalization	443
Using Names and Variables to Establish Consistency	444
Use the Schema Structure to Clearly Represent Function	445
Establishing The Ground Rules For Relationships	446
Composite Relationships: Planning Cascading Deletions	447
Aggregate: Establishing Non-deleting Relationships	448
Common Solutions to Simple Mistakes	449
Best Examples: Learning from Project Schema	450
xPDO: Third-Party Applications	452
Choosing How To Load The Application	454
Laying the Foundation	455
Create The Application Directory Tree(s)	455

Initialize An xPDO Schema	455
Create a "Parent Class" For the Project	457
Populate Tables With Data	459
Small Beginnings	467
Client Sketch Describing Basic Functionality	468
Retrieving Base Components	469
A Procedural Programming Quagmire	470
discjockey xPDO Schema	471
Discjockey Main Class	476
Extended xPDO Derived Classes	489
MODX Revolution Snippets	494
jQuery Javascript Files	502
Resources	509
Chunk Templates	513
Primary Cascading Style Sheet	516
Screen Shots From discjockey Created For This Chapter	524

Now What? **532**

~~~~~  
***Appendices***  
 ~~~~~

MODX Revolution Terminology **538**

Resource and System Settings **542**

Resource Settings	542
Resource Settings	543
System Settings	544

Primary Object Cheat Sheets **554**

modChunk	555
modPlugin	557
modSnippet	559
modUser	561
modResource	563

The Document Parser And Cache Mechanism **570**

Simplified Parser Sequence of Default Operation	571
Straight From The Core - with additional comments added	572
Cache Mechanism	573
Cached Versus Uncached Snippet Calls	574
Placeholder Considerations	574
Common Errors to Avoid	575
<u>Internet Caching Technologies</u>	576
JavaScript and CSS Caching	576
HTTP Protocol Caching	577
PHP Caching	578
Internet Proxies	578
Closing Thoughts on Internet Caching Technologie	579