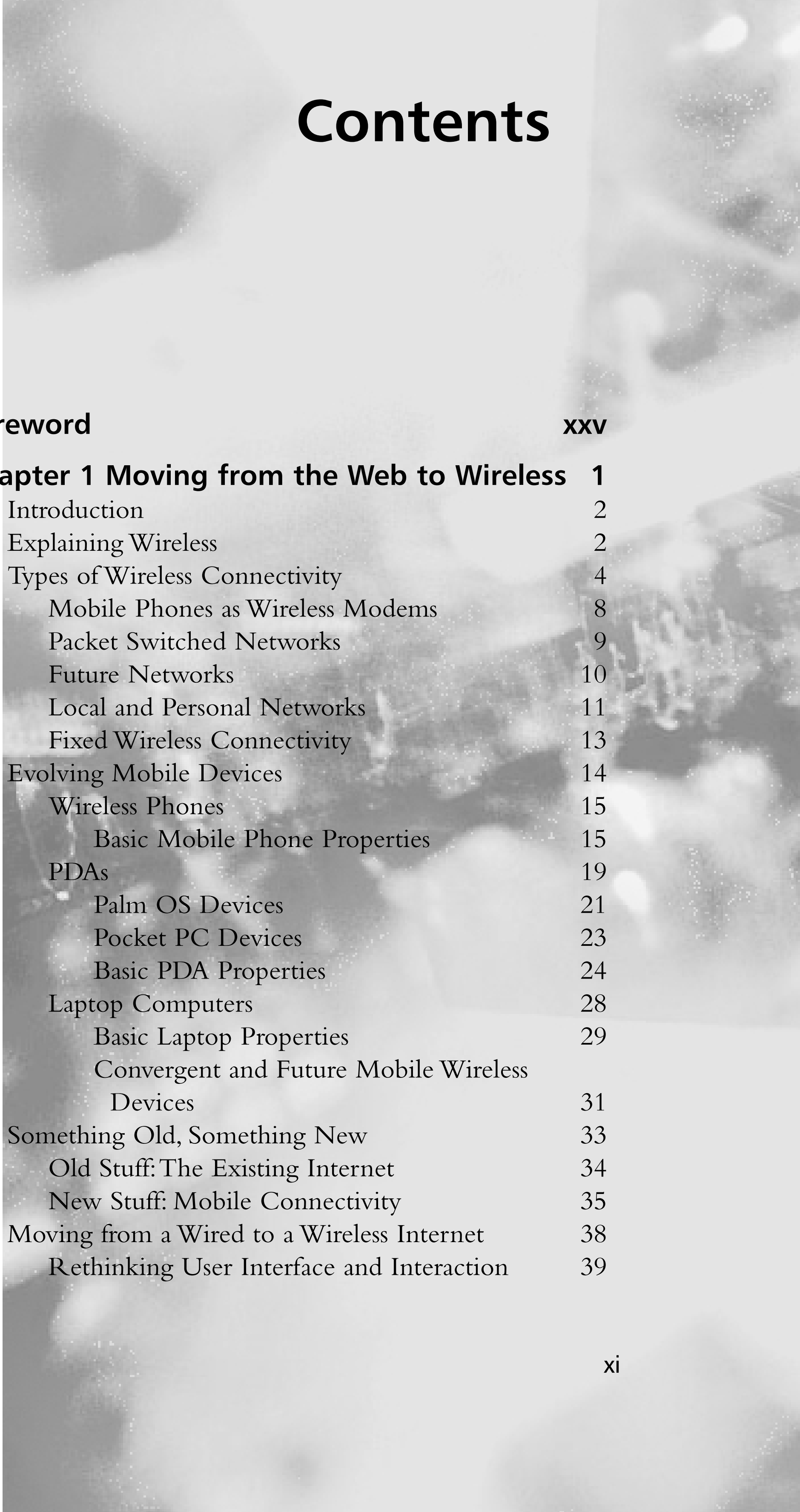
Contents



Answers to Your Wireless Questions

Q: Will I have to learn different programming if i-Mode comes to the United States?

A: i-Mode uses a subset of HTML called Compact HTML (cHTML). Anyone familiar with HTML should have no problem learning this. However, there are signs that the industry may move towards XHTML as the preferred markup language for these devices.

Foreword xxv

Chapter 1 Moving from the Web to Wireless 1

Introduction 2
Explaining Wireless 2
Types of Wireless Connectivity 4
- Mobile Phones as Wireless Modems 8
- Packet Switched Networks 9
- Future Networks 10
- Local and Personal Networks 11
- Fixed Wireless Connectivity 13

Evolving Mobile Devices 14
- Wireless Phones 15
  - Basic Mobile Phone Properties 15
- PDAs 19
  - Palm OS Devices 21
  - Pocket PC Devices 23
  - Basic PDA Properties 24
- Laptop Computers 28
  - Basic Laptop Properties 29
  - Convergent and Future Mobile Wireless Devices 31

Something Old, Something New 33
- Old Stuff: The Existing Internet 34
- New Stuff: Mobile Connectivity 35

Moving from a Wired to a Wireless Internet 38
- Rethinking User Interface and Interaction 39

Recognizing Device Limitations 40
Adding Personalization 41
Summary 43
Solutions Fast Track 45
Frequently Asked Questions 48
Chapter 2 Wireless Architecture 51
Introduction 52
Components of a Wireless Network 52
The WAP Browser 53
The WAP Gateway 54
Corresponding WAP Protocols 54
Understanding Information Flow through the Gateway 54
The Web Server 55
Adjusting the Metaphor for the Wireless Internet 56
Considering the Mobile User 57
Complementing Your Web Offering 57
Accepting the Challenge of WAP-Enabled Devices 57
Determining Device Capabilities 58
Testing Your Application on Various Devices 59
Adopting Wireless Standards 60
Options in Markup Languages 61
Wireless Markup Language 62
Compact HTML 62
Web Clipping 62
Handheld Device Markup Language 62
Using Wireless Networks and Their Evolving Generations 62
Noting the Market for Wireless Browsers and Other Applications 64
WAP Browsers 64
Java2 Micro Edition 65
i-Mode and cHTML 66
Palm Query Application 66
Web Browser 66

Using the Short Message Service

The Short Message Service (SMS) allows you to send and receive messages of about 160 characters via your mobile phone using a GSM network. This is a relatively old technology but is still quite popular.

Short Message Service 67
Summary 68
Solutions Fast Track 68
Frequently Asked Questions 70

Chapter 3 A New Markup: WML 73
Introduction 74
A Brief History of Wireless Content 74
Developing the Intelligent Terminal Transfer Protocol 74
Developing the Handheld Device Markup Language 75
Developing the Tagged Text Markup Language 75
Forming the WAP Forum 75
Combining Languages into the Wireless Markup Language 76
Projecting Future Growth 77
WML Overview 77
Creating Well-Formed Documents 78
Nesting 78
Creating Valid Documents 79
Using WML Syntax 83
Following Syntax Rules 84
Replacing Entities 84
Closing Elements 85
Characterizing the Element with Attributes 86
Case Sensitivity 86
Handling White Space 86
Commenting 87
Using Variables 87
Formatting Text 87
Displaying Fonts 88
Reserved Characters 89
Displaying Tables 90
Meta Information 91
Controlling Caching 91

Exploring the <postfield> Element

The <postfield> element specifies a name and value pair that will be sent to the server as part of a URL request. The following are attributes for the <postfield> element:

- **name** The name of the field.
- **value** The value of the field.

Bookmarking 92
Understanding the Deck of Cards Paradigm 92
WML Elements 93
Adding Attributes 93
The id and class Attributes 94
The <a> Element 94
The <access> Element 95
The <anchor> Element 95
The <b> Element 96
The <big> Element 97
The <br/> Element 97
The <card> Element 97
The <do> Element 98
The <em> Element 100
The <fieldset> Element 100
The <go> Element 101
The <head> Element 101
The <i> Element 102
The <img> Element 102
The <input> Element 103
The <meta> Element 104
The <noop> Element 105
The <onevent> Element 105
The <optgroup> Element 106
The <option> Element 106
The <p> Element 107
The <postfield> Element 108
The <prev> Element 108
The <refresh> Element 108
The <select> Element 108
The <setvar> Element 109
The <small> Element 110
The <strong> Element 110
The <table> Element 110
The <tr> Element 111
The <td> Element 111

The <template> Element 112
The <timer> Element 112
The <u> Element 113
The <wml> Element 113
Creating WML Content 114
Navigating within the Deck 114
Getting Information from the User 115
Using Server-Side Programs to Create Dynamic WML 117
Using Openwave Extensions Introduce Context 120
Navigating Parent/Child Relationships Using Extensions 121
Using the <spawn> and <catch> Extensions 123
Using the <exit> and <throw> Extensions 124
Using the <catch> Extension 124
Using the <send> and <receive> Extensions 124
WML Editors, WAP SDKs, and Emulators 126
WML Editors 126
Other Editors 129
WAP SDKs 130
Ericsson WapIDE SDK 130
Motorola Mobile ADK 130
Nokia WAP Toolkit 130
Openwave UP.SDK 130
WAPObjects 131
WML Emulators 131
Summary 132
Solutions Fast Track 132
Frequently Asked Questions 135

Chapter 4 Enhancing Client-Side Functionality with WMLScript 137
Introduction 138
What Is WMLScript? 138
Not All Phones Support WMLScript 138
WMLScript Compilation 140
How WMLScript Interacts with WML 140
Understanding the Basic Elements of WMLScript 141
Examining WMLScript Syntax 141
Examining WMLScript Data Types 142
Examining WMLScript Operators 143
Examining WMLScript Control Structures 146
Using WMLScript Libraries 147
Functions in the Class Libraries 148
Learning to Interpret WMLScript 148
Dissecting the Code 150
Performing Mathematical Operations Using WMLScript 151
Dissecting the Code 152
Using WMLScript for Input Validation 153
Dissecting the Code 155
Credit Card Validation 157
The Credit Card Validating Algorithm 157
Dissecting the Code 160
Using WMLScript and Microsoft Active Server Pages (ASP): Case Study 162
Designing the Application 162
Creating the Database 163
The WML Deck 163
Generating the WMLScript Program from ASP 165
Debugging the WMLScript 170
Lessons Learned 173
Caching Problems 174

Examining WMLScript Data Types

WMLScript supports five built-in data types:

- Integer
- Floating Point
- String
- Boolean
- Invalid

Debugging the Emulators 174

Emulators Are Relatively Unstable! 174

Summary 175

Solutions Fast Track 175

Frequently Asked Questions 177

Chapter 5 Wireless Development Kits 179

Introduction 180

The Openwave UP.SDK 4.1 180

Installing Openwave UP.SDK 181

System Requirements for the Openwave UP.SDK 181

Obtaining the Openwave UP.SDK 182

Installing the Openwave UP.SDK 182

Working with the Openwave UP.SDK 183

Accessing and Editing Local Files 184

Accessing Files through a Gateway 186

Debugging Techniques 187

The Nokia WAP Toolkit 2.1 188

Installing Nokia's WAP Toolkit 189

System Requirements for the Nokia WAP Toolkit 189

Obtaining the Nokia WAP Toolkit 190

Installing the Nokia WAP Toolkit 190

Working with the Nokia WAP Toolkit 191

Accessing and Editing Local Files 192

Accessing Files through a Gateway 195

Debugging Techniques 196

The Motorola Mobile Application Development Kit 2.0 199

Installing the Motorola Mobile ADK 199

System Requirements for the Motorola Mobile ADK 199

Obtaining the Motorola Mobile ADK 201

Installing the Motorola Mobile ADK 201

Using the Mobile ADK 204

Accessing and Editing Local Files 206

Using the Nokia WAP Toolkit 2.1

The Nokia WAP Toolkit is an environment for developing, viewing, and testing WAP applications. It includes:

- Editing, validating, and viewing WML decks
- Editing and debugging WMLScript files
- Viewing and changing WML variables inside the WAP browser
- Examining debug messages from the WAP browser
- Creating and editing WBMP images

Accessing Files through a Gateway 207
Debugging Techniques 208
The Ericsson Mobile Internet WAP-IDE 3.1 209
Installing the Ericsson Mobile WAP-IDE 209
System Requirements for the Ericsson Mobile WAP-IDE 209
Obtaining the Ericsson Mobile WAP-IDE 210
Installing the Ericsson Mobile WAP-IDE 210
Working with the Ericsson Mobile WAP-IDE 211
Accessing and Editing Local Files 212
Accessing Files through a Gateway 214
Debugging Techniques 215
The Yospace SmartPhone Emulator 2.0 216
Installing the Yospace SmartPhone Emulator 216
System Requirements for the Yospace SmartPhone Emulator 217
Obtaining the Yospace SmartPhone Emulator 217
Installing the Yospace SmartPhone Emulator 217
Developing with the Yospace SmartPhone Emulator 218
Accessing and Editing Local Files 220
Accessing Files through a Gateway 220
Debugging Techniques 221
Summary 222
Solutions Fast Track 224
Frequently Asked Questions 227

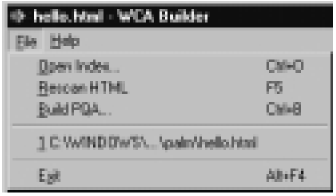
Chapter 6 Web Clipping 229

Introduction 230
What Is Web Clipping? 231
The Components of Web Clipping 233
Client-Side Components 233

Server-Side Components 234
A Typical Web Clipping Transaction 235
What Types of Hardware Support Web Clipping 236
Palm VII/VIIx Connected via Mobitex 237
Other Handheld Devices Connected via CDPD 237
Palm-Compatible Handhelds Connected via the Mobile Internet Kit 238
Working with the Palm OS Emulator 239
Downloading and Installing the Emulator 239
Transferring a ROM Image 240
Obtaining ROM Images from Palm 242
Starting the Emulator 242
Connecting the POSE to the Internet 242
Creating a Web Clipping Project with the WCA Builder 243
Hello, World! 246
Scanning the HTML 247
Creating the .pqa File 247
Installing and Uninstalling the Web Clipping Application on the POSE 248
Viewing the Web Clipping Application 249
Adding Images and Additional Pages to Your WCA 250
Web Clipping Basics 252
Unsupported Tags and Elements 252
Supported Tags and Elements 254
Using the <title> Tag 255
Using the <meta> Tag 256
Using the <body> Tag 257
Using the <table>, <tr>, and <td> Tags 257
Using the <p> Tag 259
Using the <b>, <i>, and <u> Tags 260
Using the <strong> and <em> Tags 260
Using the <ol>, <ul>, and <li> Tags 260
Using the <h1> – <h6> Tags 261

Creating a Web Clipping Project with the WCA Builder

The WCA Builder has three main options from the File menu: Open Index, Rescan HTML, and Build PQA.



Using the <img> Tag 261
Using the <a> Tag 262
Using the <form> Tag 263
Using the <select> Tag 263
Using the <input> Tag 264
Web Clipping Extensions 266
Palm-Specific <meta> Tags 268
Identifying Users with a Device ID 268
Estimating User Location by ZIP Code 270
Selecting a Date with the Datepicker Object 270
Choosing a Date with the Timepicker Object 272
Web Clipping in Action: Examples 274
Using a mailto: Link with Parameters 274
Sending E-mail via a Web Server 276
Guidelines for Authoring your Web Clipping Application 281
Summary 283
Solutions Fast Track 283
Frequently Asked Questions 286

Avoiding the Common Mistakes Made by Webmasters

Some of the more common mistakes made by Webmasters include:

- Wasting bandwidth
- Forgetting task-based design
- Providing too many options or too much information
- Using branded, Web-like terminology instead of plain language

Chapter 7 Deck of Cards: Designing Small Viewpoint Content 289
Introduction 290
Thinking In the Hand, not On the Web 291
Common Mistakes Made by Webmasters 293
Wasting Bandwidth 293
Forgetting Task-Based Design 294
Providing Too Many Options or Too Much Information 295
Using Branded Terminology Instead of Plain Language 296
Thinking Like a Mobile User 297
Segregating Tasks 298
Optimizing Bandwidth 299
Designing Coherent Navigation 303

Stacking a Deck of Cards 304
Parceling Navigation and Content 305
Utilizing WML Variables 314
Examining Display Differences Between Browsers 320
UP.Browser Interpretation 323
Nokia Interpretation 325
4thPass Kbrowser Interpretation 327
Directory.wml Example 328
Directory2.wml Example 329
Summary 332
Solutions Fast Track 333
Frequently Asked Questions 334

Choosing Mobile Content

One of the first steps to building any site is choosing what content to display on the site. The primary questions that arise in adapting a large existing site to the wireless Internet are:

1. What content/services might our users want to access while they are mobile?
2. What limitations are there to the existing mobile interfaces that we must consider?

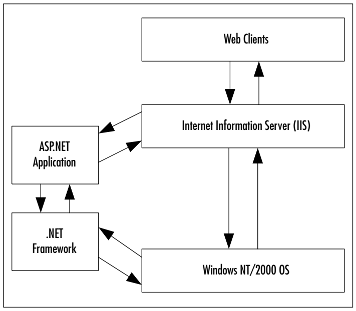
Chapter 8 Wireless Enabling Your Big Bandwidth Site 337
Introduction 338
Defining WAP MIME Types 338
Selecting which MIME Types to Add 339
Adding MIME Types to Your Server Configuration 340
Configuring the Apache Web Server 341
Adding MIME Types to Microsoft IIS 343
Detecting WAP Devices 344
Parsing Header Information 344
HTTP_USER_AGENT 345
HTTP_ACCEPT 349
Reading Other Environmental Variables 350
Redirecting Your Users to Static Content 352
Redirecting Users in PHP 353
Redirecting Users in Perl 353
Optimizing Content Distribution 356
Choosing Mobile Content 357
Convert or Redevelop? 357
Delivering Wireless Data 359
Making Your Applications Accessible 360
Implementing Wireless Graphics 362

File Formats 362
Maintaining Accessibility 363
Converting Your Images 363
Summary 364
Solutions Fast Track 364
Frequently Asked Questions 366

Chapter 9 Microsoft Mobile Internet Toolkit 367

Introduction 368
Overview of the .NET Mobile Architecture 368
Devices Supported by the Microsoft Mobile Internet Toolkit 369
System Requirements 369
Obtaining and Installing the Microsoft Mobile Internet Toolkit 370
Introduction to ASP.NET 371
The Content Components 376
HTML Server Controls 377
ASP.NET Server Controls 377
The Code Components 378
ASP.NET Architecture 380
Developing Mobile Web Forms 381
Using Multiple Forms in a Single Page 385
Linking to Forms on Other Pages 386
Dissecting Code 388
User Inputs 389
Text and Password Input 389
List Selection 393
Selecting from a List 394
Data Binding List Items 396
Dissecting the Codes 399
Events 400
Displaying Images 401
Validation Controls 405
Paginations 407
Calendar Control 409

The Architecture of ASP.NET



Accessing Data with ADO.NET 411
A Brief Look at ADO.NET 411
Data Providers 412
ADO.NET DataReader 414
Dissecting the Codes 415
ADO.NET Dataset 417
Summary 423
Solutions Fast Track 423
Frequently Asked Questions 426
Chapter 10 Securing Your Wireless Web 429
Introduction 430
Comparing Internet and Wireless Security 431
Security Challenges of the Wireless Web 433
Lack of Standards 434
Horsepower, Bandwidth, and Weak Encryption 434
User Awareness and Unsecure Devices 435
Mistrust of Wireless ASPs 436
Potential for New Viruses 436
Understanding Your Security Objectives 437
Security Models of the Wireless Web 438
Public and Private Key Cryptography 439
WTLS and Point-to-Point Security Models 442
How WTLS Works 443
WTLS Classes 444
The WAP Gap 444
The Seven Layers of Point-to-Point Security 446
Embedded Security Technology 447
Mobile Operator Network Security 448
Secure Mobile Operator Gateways 448
Authentication 448
Data Center and Network Security 449
Secure Application Interfaces 452
Problems of a Point-to-Point Security Model 452
Sniffing and Spoofing 452
Session Management and URL Rewriting 453

Understanding the Seven Layers of Point-to-Point Security

Point-to-point security can be broken down into seven layers, corresponding to the steps in the communication path between mobile devices and Web servers or applications:

1. Embedded Security Technology
2. Secure Air-Connect Technologies
3. Mobile Operator Network Security
4. Secure Mobile operator Gateways
5. Authentication
6. Data Center and Network Security
7. Secure Application Interfaces

Man-in-the-Middle Attack 453
No Complete Solution 454
PKI Technology and End-to-End Security Models 454
How to Deploy a PKI 456
Server Side PKI Integration 456
Client Side Devices 456
Choosing a Certificate Authority 456
Certificate Management Framework 457
Certificate Deployment 457
Practical Limits of PKI Technology 457
Using PDAs with PKI Security 458
The Future of Security on the Wireless Web 458
Summary 460
Solutions Fast Track 461
Frequently Asked Questions 464

Webmaster's Guide to the Wireless Internet Fast Track 467

Index 489