

Part Contents

■ Preamble	211
Goals	212
■ Static Provision of Data	215
Static versus Dynamic	215
A Simple, Sample Type Provider	217
Design Phases & Implementation Boundaries	221
■ Assembly Configuration Settings	227
Overview	228
Xml Schemas	232
<i>Schema Components</i>	233
Attributes and Simple Types	234
Complex Types	239
Primitive Types in F# and Xml	241
Handling Dates and Times	242
Element Types	246
Testing the Schema	250
Schema Location	251
■ XSettings Underlying Data Access	253
Planning for Data Provision	254
Coding Core Functionality	257
<i>Preliminaries</i>	257
<i>Supporting Structure Types</i>	260
<i>Error and Result Reporting</i>	263

<i>Composite Function Evaluations</i>	266
<i>System.Uri - a Common Currency for File Referencing</i>	267
<i>Xml Manipulation Functions</i>	270
<i>Core Data Provision</i>	276
■ The Type Provider Common Library	281
Project Layout & Structure Overview	282
XPath for Attribute Values	286
Enhancing Uri File Referencing	287
Functions supporting the XSettings Type Provider	305
■ Generating an API Reference	313
An Overview of Xml Comments	314
Implementing SHFB in your Solution	317
<i>Installing Sandcastle Help File Builder</i>	318
<i>Installing NuGet SHFB Packages</i>	319
Configuring the SHFB Project	319
<i>The Content and Version Files</i>	320
<i>An Icon for your Help Output</i>	322
<i>SHFB Project Properties</i>	323
Building an SHFB Project	329
<i>Website based Help Output</i>	329
<i>MS Help File Format Output</i>	332
<i>Open Xml Format Help Output</i>	334
■ Debugging the Common Library	337
Friendly Assemblies	337
Testing Referenced Internal Types	338
■ Coding the XSettings Type Provider	343
Implementing XSettings	345
<i>Type Provider Parameters</i>	349
Instantiation of a Type Provider Implementation	351
<i>Parsing Type Provider Instantiation Parameters</i>	353
<i>Setting up the Modified ProvidedTypeDefinition</i>	355
<i>Informational Provided Properties</i>	356

<i>Quotation Expressions in a Small Nutshell!</i>	358
<i>Mapping Underlying Data to Provided Properties</i>	362
■ Debugging a Type Provider	371
Setting Project Debug Properties	372
An Overview of the Microsoft Process Explorer	373
A Script for Type Initialisation Debugging	374
Debugging an Instantiated Type Provider	380
■ A Manifest Resources Type Provider	383
Core Functionality for Manifest Resources	384
<i>Accessing Assemblies for Manifest Resources</i>	385
<i>Qualifying the Data Source</i>	387
Mime type Evaluation	390
RESX Localisation	391
<i>Modelling Type Provider Data</i>	392
Modelling Output Types	392
Modelling Provided Properties	393
<i>Mapping the Underlying Data Source</i>	396
Preamble	397
Manifest Resource Typing	398
Uri Assignment and Assembly Loading	399
Mime type evaluation	400
Mapping the Manifest Resources	405
Enumerating an Assembly's Manifest Resources	407
<i>Embedding Core Functionality in the Common Library</i>	411
Characterising a WPF Resource	413
Characterising an Embedded Resource	415
Characterising a RESX Resource	416
<i>Wrapping a Target Assembly Manifest</i>	421
Mapping Wpf and Embedded Repositories	425
Mapping Resx Repositories	425
Extended Functionality for Manifest Resources	428
<i>Outputting Graphics</i>	429

<i>Outputting Binary Resources</i>	432
<i>Persisting a Resource to a Disk File</i>	432
Persisting Icon and Binary Resource Types	433
Persisting Image Resource Types	435
<i>Output Types for BAML and Xaml Resource Types</i>	438
<i>Implementing WPF Graph as Extended Functionality</i>	440
Implementing the Manifest Type Provider	445
<i>Evaluating Provided, Nested Properties</i>	446
The Top Level of Nesting - The Manifest Repository	447
Dealing with Resx Repositories	449
Dealing with Resx Satellite Assemblies	451
Evaluating Provided Properties	451
Triggering Provided Property Evaluation	453
Synchronous or Asynchronous?	454
<i>Testing the Manifest Type Provider</i>	454
■ An Extended Resources Provider	459
Schema Simple Type Enumeration Facets	460
Schemes and Schemas	464
<i>Media Types</i>	464
<i>Language Codes</i>	466
Downloading and Parsing BCP47 Tags	468
BCP47 Schema Generation	481
<i>The XResources Schema</i>	486
<i>Sprite Support</i>	496
Modelling the Type Provider Nesting Levels	499
XResources Extended Functionality	501
Core Functionality	503
<i>Supporting Types</i>	503
<i>Mapping Underlying Resources</i>	508
Implementing the XResources Type Provider	519
<i>The Trigger for Evaluating Provided Properties</i>	523
<i>Nesting Level Functions</i>	524

<i>Evaluating Provided Properties</i>	525
For a Resource Type of String	527
For a Resource Type of Binary	527
For a Resource Type of Image or Icon	527
For a Resource Type of File	533
For a Resource Type of Media	534
For a Resource Type of Xaml	535
Using the XResources Type Provider	535

Accompanying Videos

231. Demonstrating Static Type Provision - Parts I through III.	221
232. Assembly Configuration Settings.	231
233. An overview of XSD types used by the XSettings Schema.	233
234. Schema Attribute Groups & Simple Types – Parts I through III.	239
235. Schema Complex Types	241
236. Defining the Root Element - Parts I & II.	250
237. Planning for data provision - Parts I & II.	256
238. Core functionality preliminaries - Parts I & II.	260
239. Core Functionality Supporting Structures.	262
240. Error Management and Standardisation of function output.	266
241. Using Uris for file referencing.	269
242. Xml functions - Parts I & II.	276
243. Building the source data map.	278
244. Common Class Library Overview - Parts I & II.	283
245. Using XPath to evaluate attribute values.	287
246. Host Design Time = Type Provider Runtime.	289
247. The Host Assembly's Project Path as a relative Uri search base.	290
248. Mode-specific Uri type extensions.	293
249. Dealing with web based file verification - Parts I & II.	296
250. Determining and quantifying Uri usage - Parts I & II.	299

251. Searching for a file.	300
252. File validation for a parsed potential file name and path.	301
253. The Uri.Make extension - Parts I & II.	305
254. XSettings related function for Type Provider Support - Parts I through III.	311
255. An Overview of Sandcastle Help File Builder - Parts I & II.	314
256. SHFB Project Components.	321
257. SHFB Project Properties - Parts I through V.	329
258. Declaring friendly assemblies to enable internal type access.	337
259. Debugging Common internal types.	340
260. Layout of the implementation class for a Type Provider - Parts I through III.	350
261. Parameter validation and type definition.	355
262. Defining informational provided properties - Parts I & II.	358
264. Evaluating Provided Properties for the XSettings Type Provider - Parts I through III.	370
265. Setting up a Type Provider Project for Debugging.	372
266. A Type Provider Debugging Session - Parts I through III.	382
267. An Overview of The Manifest Type Provider source data repositories.	383
268. Qualifying resource repositories and types.	389
269. Reusing our mime type guesstimation.	390
270. Characterising the Manifest data for provided property output - Parts I & II.	393
271. Mapping manifest resources prerequisites - Parts I & II.	404
272. Mapping the manifest resources - Parts I & II.	405
273. Testing manifest resource mapping - Parts I & II.	410
274. Common Library Core Functionality for Embedded and WPF Repositories - Parts I through III.	418
275. Common Library Core Functionality for RESX Repositories - Parts I through IV.	422
276. Manifest Extended Functionality - Output Types for Image, Icon and Binary Resource Types.	432
277. Persisting Images, Icons and Byte Arrays to Disk - Parts I & II.	437

278. Manifest Extended Functionality - Output Types for Baml and Xaml Resource Types.	438
279. Implementing WPF Graph as Extended Functionality - Parts I & II.	445
280. Evaluating Wpf and Embedded Provided Properties - Parts I through III.	449
281. Evaluating Resx Provided Properties - Parts I & II.	453
282. Synchronous or Asynchronous Parallelisation.	454
283. Running the Manifest Type Provider - Parts I & II.	457
284. An Overview of Media Type and Language Tag sourcing	459
285. Extracting Simple Type Enumeration Facets from a Schema Set - Parts I & II.	463
286. Constructing the Media Types Schema - or not as the case is.	466
287. An example of how not to use a Union for Language Codes.	467
288. Setting up the parsing of BCP47 Tags - Parts I through III.	478
289. Parsing each BCP47 Tag row - Parts I & II	479
290. Generating the BCP47Tags Schema - Parts I & II.	485
291. Testing BCP47 Schema Generation.	486
292. An Overview of the XResources File and Schema - Parts I through IV.	496
293. Dealing with Sprites.	498
294. Nesting Levels for the XResources Type Prvoder.	500
295. Extended Functionality for the XResources Type Provider.	502
296. Core Functionality Supporting Types.	503
297. Encapsulating Resource Node Information for Provided Property Generation.	511
298. Evaluating the set of underlying sources for subsequent category totalling - Parts I & II.	515
299. Category Totalling the Underlying Resources by Segment and Language.	519
300. XResources Type Provider Overview and Informational Provided Properties - Parts I & II.	522
301. Evaluating the XResources Provided Properties - Parts I through III.	535
302. Using the XResources Type Provider - Parts I & II.	537