Pro Mapping in BizTalk Server 2009

Jim Dawson and John Wainwright
Anyone who has spent much time in the IT world knows that success is often due in very large measure to those who came before us who took the time to stop and explain a technique or a method, patiently guided us past our mistakes, or had the grace to let us learn from our failures. While the debt incurred cannot be paid back, it can certainly be acknowledged. So we raise our glasses to the many individuals in our checkered pasts who viewed expertise as something to be shared and not hoarded.
Contents at a Glance

About the Authors ................................................................. xvii
About the Technical Reviewer .................................................. xix
Acknowledgments .................................................................. xxi
Introduction .......................................................................... xxiii

PART 1 ■ ■ ■ Basic Concepts of BizTalk Mapping
■ CHAPTER 1 Creating a Simple Map ........................................ 3
■ CHAPTER 2 How BizTalk Maps Work ................................... 31
■ CHAPTER 3 Using Scripting in Maps ...................................... 59
■ CHAPTER 4 Testing BizTalk Maps ........................................ 93

PART 2 ■ ■ ■ Step-by-Step Mapping
■ CHAPTER 5 Mapping Conditionals ....................................... 115
■ CHAPTER 6 Dealing with Numbers ......................................... 131
■ CHAPTER 7 Manipulating Strings .......................................... 153
■ CHAPTER 8 Manipulating Dates and Times ............................ 169
■ CHAPTER 9 Collecting Data .................................................. 183
■ CHAPTER 10 Accessing External Data ................................. 203

PART 3 ■ ■ ■ Looping
■ CHAPTER 11 Using Basic Looping Controls ........................... 229
■ CHAPTER 12 Handling Advanced Looping ............................ 259
PART 4  ■  ■  ■ EDI Mapping with BizTalk

■ CHAPTER 13  Introducing Electronic Data Interchange (EDI)  ........................................... 277
■ CHAPTER 14  Exploring EDI Looping .................................................................................. 301
■ CHAPTER 15  Processing EDI Code Pairs .......................................................................... 317
■ CHAPTER 16  Unraveling the SDQ Segment ....................................................................... 335
■ CHAPTER 17  Taming the Dreaded 856 ASN HL Loop ......................................................... 359

PART 5  ■  ■  ■ Advanced Techniques

■ CHAPTER 18  Building Custom Assemblies and Functoids .................................................. 393
■ CHAPTER 19  Examples from the Fringe ............................................................................ 413

■ INDEX .................................................................................................................................. 441
Contents

About the Authors ................................................................. xvii
About the Technical Reviewer .................................................. xix
Acknowledgments .................................................................. xxi
Introduction ............................................................................ xxiii

PART 1  Basic Concepts of BizTalk Mapping

CHAPTER 1  Creating a Simple Map ........................................ 3

Using the Development Studio ............................................ 3
  The Mapper Grid ............................................................... 3
  The Solution Explorer Window ....................................... 6
  The Toolbox Window ...................................................... 6
  The Properties Window ................................................... 8
  The Working Environment .............................................. 8
Creating the Map .............................................................. 9
  Creating a BizTalk Project .............................................. 9
Adding a Project File Structure ......................................... 10
Creating the Input and Output Schemas ......................... 11
Creating a New BizTalk Map File ................................... 14
A Test Input file .............................................................. 16
Building the Map ............................................................ 17
Summary ........................................................................... 30

CHAPTER 2  How BizTalk Maps Work ................................... 31

Understanding How BizTalk Maps Work ....................... 31
  Constructing Map Rules ................................................. 32
  Generating the Underlying Map ................................... 33
Examining the Hidden Constructs in Maps ..................... 36
  Compiler-Generated Links .......................................... 36
  Compiler-Generated Variables ................................... 38
  Compiler-Generated Scripts ...................................... 40
Digging into the HelloWorld Map Rules .............................................. 40
Using the Logical String Functoid Rule to Qualify a
Data Extract ........................................................................ 42
Looking Deeper into Understanding Maps.......................... 46
Summary................................................................................. 57

CHAPTER 3 Using Scripting in Maps ........................................ 59
Choosing Between Scripts and Functoids ................................. 59
Our Guidelines for Scripts vs. Functoids ................................. 61
Using the Rule of Six .......................................................... 61
Selecting from Available Scripting Languages ......................... 65
C# .................................................................................. 66
VB.NET ............................................................................. 71
JScript .NET ....................................................................... 72
XSLT Call Template ............................................................ 74
Using Inline XSLT ................................................................ 74
Using External Assemblies ..................................................... 76
Choosing Your Scripting Language ......................................... 82
Combining Types of Scripting ............................................... 84
Examples of When You Should Use a Script ............................ 88
Summary................................................................................. 91

CHAPTER 4 Testing BizTalk Maps ............................................. 93
Incremental Testing .............................................................. 93
Addressing Architecture Considerations ................................. 96
Organizing Your BizTalk Applications ..................................... 97
Organizing Your BizTalk Solutions ......................................... 98
Organizing Your BizTalk Projects .......................................... 98
Separating Schemas and Maps ............................................... 99
Test Data ............................................................................. 100
Generating Test Data from a Schema ..................................... 100
Generating Test Data from a Flat File .................................... 101
Generating Test Data from an EDI File .................................. 103
Testing in the Map Editor ...................................................... 103
Advantages to Testing in the Map Editor ............................... 104
Testing Techniques for the Map Editor ................................. 104
Testing Output Structures .................................................... 109
Summary................................................................................. 110
PART 2  ■  ■  ■  Step-by-Step Mapping

CHAPTER 5  ■  Mapping Conditionals

- Checking If Data Exists .................................................. 115
- Blocking Output When Data Doesn’t Exist .......................... 117
- Checking If a Specific Condition Exists ............................. 119
- Checking If a Specific Condition Does Not Exist ..................... 121
- An If/Else Condition ....................................................... 123
- Using the Logical OR for a Single Return Value ..................... 125
- Using a case Statement .................................................... 126
- Summary ............................................................... 128

CHAPTER 6  ■  Dealing with Numbers

- Is This a Number? .......................................................... 131
  - Determining if a String Is a Number ................................ 133
  - Using a Regular Expression to Check for a Number ............. 134
  - Testing for a Numeric Value with the XSLT Number Function . 134
- Is This Number an Integer? ............................................. 135
- Rounding Numbers ......................................................... 137
- Summing Numbers ......................................................... 139
  - Using the Cumulative Sum Functoid to Sum Data ................ 139
  - Using an Inline XSLT Script to Sum Data ......................... 140
  - Summing Complex Data ................................................ 141
- Counting Records .......................................................... 141
  - Counting Records Simply ............................................. 142
  - Counting Qualified Records Simply ................................ 142
  - Counting Complex Qualified Records ............................... 143
- Converting Real Numbers to Integers (and Back Again) ............ 145
- Moving the Sign to a Trailing Position ................................ 146
- Converting a Number to or from Sign-Trailing Overpunch Format ... 149
- Summary ............................................................... 152

CHAPTER 7  ■  Manipulating Strings

- Trimming Strings .......................................................... 153
- Trimming Nonblank Characters from a String ....................... 156
- Padding a String .......................................................... 157
- Selecting Substrings from String Data .................................. 158
- Concatenating Strings ..................................................... 161
PART 3  ■  Looping

CHAPTER 11  Using Basic Looping Controls .................................................. 229

Understanding Loops ................................................................. 229
Understanding BizTalk Mapping Engine’s Basic Looping Concepts ................. 231
 Adding Context Nodes ............................................................ 234
 Adding Compiler-Generated Links ............................................. 236
 Limiting the Output of a Loop .................................................... 239
 Forcing Looping in the Output .................................................... 242
 Simultaneously Forcing and Limiting a Loop .................................... 244
 Many-to-One Looping .............................................................. 247
 One-to-Many Looping .............................................................. 249
 One-to-Nested Looping ............................................................ 250
 Nested-to-One Looping ............................................................ 254
 Summary .................................................................................. 257

CHAPTER 12  Handling Advanced Looping ..................................................... 259

Many-to-One Looping with a Source Child Loop ..................................... 259
One-to-Many Looping with a Child Target Loop ..................................... 266
Summary .................................................................................. 273

PART 4  ■  EDI Mapping with BizTalk

CHAPTER 13  Introducing Electronic Data Interchange (EDI) ....................... 277

EDI Standards-Based Transactions .................................................... 278
Versions and Releases .................................................................... 279
X12 ......................................................................................... 279
EDIFACT .................................................................................... 280
CONTENTS

Using the Basic Building Blocks of EDI Messages ........................................ 280
  Examining a Simple X12 Data Element ........................................ 281
  Using Simple EDIFACT Data Elements ........................................ 282
  Using Composite X12 Elements ............................................... 283
  Using Composite EDIFACT Elements ....................................... 284
  Creating X12 Segments ................................................ 284
  Creating EDIFACT Segments .............................................. 286
  Exploring X12 Segment Syntax Rules and Mapping .................... 286
  Understanding X12 Transactions ...................................... 287
  Understanding EDIFACT Transactions ................................ 290
  Exploring X12 Interchanges ............................................ 291
    ST and SE ........................................................ 291
    GS and GE ...................................................... 292
    ISA and IEA .................................................. 293
  Exploring EDIFACT Interchanges ...................................... 295
    UNH and UNT .................................................. 296
    UNG and UNE .................................................. 297
    UNA, UNB, and UNZ ........................................... 298
  Illustrating the Differences Between X12 and EDIFACT ............. 299
  Summary ..................................................................... 300

CHAPTER 14  Exploring EDI Looping ......................................................... 301

  Looping from EDI to an Application .................................. 301
    Exploring the X12-to-Application Solution ......................... 305
    Exploring the EDIFACT-to-Application Solution .................... 309
  Looping from an Application to EDI .................................. 310
  Summary ..................................................................... 316

CHAPTER 15  Processing EDI Code Pairs ............................................... 317

  Introducing EDI Code Pairs .............................................. 317
  Handling ID Codes in X12 REF and EDIFACT RFF Segments ........ 318
    Looking for Specific Code Pairs in an
    X12-to-Application Map .......................................... 318
    Looking for Specific Code Pairs in an
    EDIFACT-to-Application Map .................................... 320
  Using a Script to Select Specific Code Pairs from
    One REF or RFF Loop ............................................ 321
CONTENTS

Using a Recursive XSLT Call Template to Process Code Pairs ........322
Using Nested XSLT Call Templates to Process Code Pairs ............324
Multiple Code Pair Positions in a Single Segment ...................327
  Handling Multiple Code Pairs in an Inbound Segment with C# ....328
  Handling Multiple Code Pairs in an Inbound Segment
    with XSLT ..................................................................330
Creating Code Pairs in EDI Output ......................................330
Summary .............................................................................333

CHAPTER 16 Unraveling the SDQ Segment ..................................335

Exploring the SDQ Segment ................................................335
Mapping Inbound SDQ Data ...............................................336
  Gathering the Data .......................................................337
  Sorting and Storing the Data into Hash Tables ....................338
  Retrieving the Locations for Output ................................341
  Outputting the Locations ..............................................343
  Adding the Item Numbers and Quantity Information
    to the Output ..............................................................345
  Considering the Decisions in This Map ............................347
Mapping Outbound SDQ Data .............................................348
  Collecting the Item and Quantity Information ....................348
  Retrieving Item Numbers from the Hash Table ..................350
  Building the LIN and ZA Loops .....................................351
  Adding the SDQ Node to the ZALoop1 Nodes ......................355
Summary .............................................................................358

CHAPTER 17 Taming the Dreaded 856 ASN HL Loop .......................359

Exploring the HL Loop ........................................................359
Planning to Map the HL Loop ..............................................363
Mapping an Inbound 856 .....................................................363
Mapping an Outbound 856 ASN ..........................................370
  Building the ASN with an XSLT Script .............................378
Creating the Simple Outbound 856 ASN ..............................381
Summary .............................................................................390
PART 5 ■ Advanced Techniques

CHAPTER 18  Building Custom Assemblies and Functoids .............................................. 393
  Deciding to Use a Custom Assembly or Custom Functoid ............................................. 393
  Creating an External Assembly for Date Conversion Functions ............................... 394
  Building a Custom Functoid ......................................................................................... 396
    Establishing a Common Terminology ........................................................................ 397
    Laying Out the Approach ..................................................................................... 397
    Building the Functoid Worksheet ........................................................................... 398
    Creating the Class Library Project ....................................................................... 399
    Creating a Strong-Named Key File ..................................................................... 399
    Setting Up the Application Values ................................................................... 400
    Setting Up the Functoid Resource File .............................................................. 401
    Adding the Bitmap for the Functoid Icon ......................................................... 402
    Adding Resources to the Project ..................................................................... 403
    Adding Code to the Class Library Object ......................................................... 403
    Building and Deploying the Class Library Object ............................................ 406
    Adding the Functoid to the Toolbar ................................................................ 406
    Testing the Functoid ............................................................................................ 407
  Adding More Functoids ................................................................. 408
  Doing a Custom Database Lookup ...................................................................... 408
  Summary .................................................................................................................. 412

CHAPTER 19  Examples from the Fringe ................................................................. 413
  Modifying Outbound EDI Envelope Data ................................................................. 413
    Creating the Custom Pipeline Component ............................................................ 414
  Building BizTalk Maps Outside the Map Editor ....................................................... 417
    Deciding to Use an XSLT File As Your Map ...................................................... 417
    Creating an External XSLT Map ........................................................................... 418
    Implementing the External XSLT File As a Map .............................................. 420
    Accessing C# Scripts in a Stand-Alone XSLT Map ........................................... 421
    Calling Other XSLT Templates From a Stand-Alone XSLT Map ...................... 422
    Building Dynamic Tag Names in Your XML Output ........................................ 422
  Breaking Up Large Strings Using Table Looping .................................................. 423
  Reformatting Source Files for Easier Mapping ...................................................... 428
  Testing C# and VB Scripts .................................................................................... 433
  Setting Up a Test Project ....................................................................................... 433
  Creating Test Code ................................................................................................. 435
About the Authors

JAMES (JIM) LEE DAWSON spent his first life in the U.S. Marine Corps, gaining a number of skills that had no direct applicability to civilian life. He spent a tour in Vietnam as an infantry platoon and company commander and left the Marines in 1978 as a captain.

Since that time, his new life includes more than 25 years’ experience in technical environments, including working as a programmer analyst, systems analyst, EDI analyst, BizTalk integrator, team leader, project manager, and departmental director. He has programmed in C, C++, C#, VB, XSLT, PHP, SQL, HTML, FORTRAN, PL1, and Assembly. He has worked with the BizTalk mapping engine for the last six years, concentrating on applying BizTalk to EDI uses as well as creating BizTalk maps.

Jim is currently a managing partner of Second Star Professional Services (SSPS), LLC, a Microsoft Registered Partner that provides expert electronic commerce implementation and integration services. SSPS focuses on BizTalk Server and EDI mapping using Mercator, GXS Application Integrator, and the BizTalk mapper. Jim has extensive experience with the Covast EDI accelerators for BizTalk 2002, 2004, and 2006.

Jim’s BA from the University of North Carolina in English enabled him to produce a number of obscure written works, including nonfiction with esoteric titles such as *Mechanized Forces in the Link-up Operation* and *The Soviet Meeting Engagement* and fiction titles like *Black Dugal’s Music Shoppe* and *The Sapient Sorcerer*.

Jim currently resides in Siler City, North Carolina.

JOHN D. WAINWRIGHT has over 30 years of experience in various areas of information management, with extensive programming and project management experience in software development. He worked in corporate, manufacturing, and consulting environments on a wide variety of systems, including inventory control, shipping, distribution, cost accounting, shop floor control, and computer-integrated manufacturing control. After 15 years doing software development, John began working as a consultant in EDI, focusing mainly on implementing EDI in supply chain management, transportation, and distribution. John has spent much of the last five years doing BizTalk mapping and claims that he’s never seen a purchase order map he doesn’t like.

He is currently a managing partner of Second Star Professional Services (SSPS), LLC. John and coauthor Jim Dawson, tired of IT management jobs that weren’t any fun, started the company nine years ago so they could get back to doing interesting stuff. They now provide consulting in the EDI integration arena. During that time, John worked primarily with BizTalk mapping, starting out with the Covast EDI accelerator for BizTalk and moving on to work with the R2 product. He’s also had extensive mapping experience with Mercator and Application Integrator.

John has an interesting, but not useful, BA in history from Allegheny College. He did graduate work (more interesting and less useful) at Temple University. Fortunately, while learning that an academic career didn’t seem too promising, he learned some programming skills as well.

He currently resides in Raleigh, North Carolina.
JEFF SANDERS is a 16-year IT industry veteran with extensive experience in solutions architecture, BizTalk, SharePoint, and .NET. Jeff’s interests lie in design patterns of message-based architectures and connected systems design, Dublin, WCF, WF, and reducing complexity.

Jeff is a group manager and solutions architect with Avanade Inc., a global IT consultancy specializing in solutions based on the Microsoft enterprise platform that help to achieve profitable growth. He works out of the east region with some of the most talented and customer-obsessed professionals he has ever met.

Jeff also independently consults with DynamicShift and speaks at regional and local user groups on Microsoft technologies and industry-related topics.

With a deep interest in providing better information to business decision makers, event-driven architecture, complex event processing, and business intelligence, Jeff is coauthoring Pro BAM in BizTalk Server 2009 (Apress, 2009), to be released later this year. You should buy a copy, or five.

Jeff would like to thank Jim Dawson and John Wainwright, Mark Beckner, and Kylie Johnston. Most of all, he would like to thank Lisa, who deserves so much more. Yeah, Beesley, we’re in the same time zone, but it does feel far.
We’d like to thank the editors and reviewers at Apress and all the clients who allowed us to learn at their expense.
When people begin learning Visual Basic, C#, BizTalk, and other technical subjects, they buy how-to books. That is why most technical folk have a reference library close at hand. If they are working with BizTalk, that library contains cookbooks and references on XML, XSLT, and XPATH as well as BizTalk.

The one subject that references fail to cover well is BizTalk mapping. No tips and techniques. No teach yourself books. No mapper’s bible. The few BizTalk reference manuals available include limited information on mapping, but these short treatments provide no help in understanding how to overcome minor mapping difficulties, much less help solving complex mapping problems.

As we learned how to adapt BizTalk to X12 and EDIFACT mapping, we wondered if perhaps BizTalk and EDI were not meant to work together in any meaningful way. But as we used brute force to plow our way through EDI maps, we discovered that we could solve any mapping problem without resorting to pre- or post-processing. In fact, we learned that there are often several ways to solve any mapping problem. We’ve put the most useful of those techniques into this book.

We began with the intent of writing a book with a bunch of sample maps but soon came to the conclusion that such a book is neither possible nor desirable. Why? Because BizTalk mapping is never about solving one large mapping problem. The questions we face every day as mappers are more like “How do I solve this little piece here?” rather than “How do I create this map?” Any mapping project consists of a myriad of small mapping problems, a series of one-percent solutions. Solve the first, test it, and then move on the next. Eventually, all the pieces are solved, and the map is completed.

In that vein, we cover many little mapping solutions in the hope that as you encounter a new type of problem, you can turn to this reference and find a solution or at the least some options that will light the way toward a solution. Anyone who creates maps using the BizTalk Map Editor will benefit from owning this manual.

Who This Book Is For

We include a little something for everyone in this book, and we present a broad range of subjects in the hope that mappers with all levels of skill and experience will find valuable information here. We include basic material to introduce the novice BizTalk mapper to basic mapping techniques. For those BizTalk experts who are not experienced with EDI, we provide a section of the book dedicated to EDI mapping techniques. For advanced BizTalk mappers, we provide examples of how to deal with of more complex mapping topics such as accessing external data or building custom functoids.
How This Book Is Structured

The book is divided into five parts.

Part 1, “Basic Concepts of BizTalk Mapping,” consists of four chapters that provide an introductory overview of BizTalk mapping. Since the BizTalk mapper works a bit differently from other mapping engines, Chapter 1, “Creating a Simple Map,” introduces those who are new to BizTalk mapping to basic mapping concepts. Chapter 2, “How BizTalk Maps Work,” takes a look under the hood to explore how BizTalk maps actually work and focuses on the underlying map code that is generated when you build a map. Chapter 3, “Using Scripting in Maps,” looks at how custom scripting can be employed in BizTalk maps. Chapter 4, “Testing BizTalk Maps,” concludes the section with an overview of how to test and debug BizTalk maps in the map editor and the development studio using basic testing techniques.

Part 2, “Step-by-Step Mapping,” contains six chapters, each of which deals with a specific type of mapping problem: conditionals, numbers, strings, dates and times, collecting and counting, and handling external data.

Part 3, “Looping,” contains two chapters that deal exclusively with how to handle the complexities of looping in BizTalk maps.

Part 4, “EDI Mapping with BizTalk,” begins with an introduction to EDI and includes subsequent chapters on solving EDI looping problems, using EDI code pairs, and unraveling SDQ segments. This section ends with a chapter devoted exclusively to untangling some of the mysteries of how to map the Advanced Ship Notice.

Part 5, “Advanced Techniques,” includes one chapter on constructing external assemblies and custom functoids and a final topic on miscellaneous topics, such as specialized pipeline construction for EDI envelope modification, construction of external XSLT maps, and a section on how to test your custom scripts in the development studio.

Conventions

We use several standard conventions for various types of text in the book. They are presented in a special typeface to bring them to your attention.

*Note, Tip, and Caution elements will look like these:

---

❑ **Note** Notes will usually contain information that we think might be worth keeping in mind.

❑ **Tip** These are special techniques or methods that might be of help.

❑ **Caution** These usually offer a warning to help you avoid a mistake we have made.
Code fragments look like this:

```csharp
public string LoadPO1Table(string inQty, string inPairs)
{
    string pairs = inPairs;
    string pair = "\n; // Empty the hash table each time to avoid having residual values
    PO1Table.Clear();
    // pairs contains a copy of the input string. If there is still data in the string
    // continue processing through the while loop
    while (pairs != "")
    {
        // ...
    }
}
```

We've also included code comments: We tried to annotate the code liberally. Some comments are prefixed with a double slash /**; others use <!-- to begin and --> to end a comment. We did not distinguish between script types but generally used the method that made the code easier to read. The // is actually the standard character string for commenting a single line of C# code, while <!-- and --> are the standard comment strings for enclosing XSLT comments. Please forgive us for this transgression. If you download the code, any comments will be correctly formatted so that the code will compile.

Prerequisites

The book focuses on mapping with the BizTalk Map Editor, a topic that requires a basic knowledge of the support structure for that tool, in particular Microsoft Visual Studio. Basic knowledge of the C# or VB programming languages, BizTalk schemas, XML, and XSLT—all pieces of the BizTalk mapping environment—is also important. A basic understanding of BizTalk port and party configuration in the BizTalk Administration Console helps with testing.

EDI knowledge is not a prerequisite (although it is certainly beneficial), because we realize that most users of the BizTalk mapping engine are not familiar with EDI.

Downloading the Code

All examples in the book—maps, scripts, and test data—are available to readers at www.apress.com, in this book’s Downloads section. In most cases in this book, we build a map incrementally and show graphics and code for each step. Generally speaking, the downloadable maps include only the completed map.

The downloadable scripts and maps are arranged as BizTalk projects named for the chapter in which they appear. This method of organization means that some schemas, scripts, maps, and data files may appear in more than one chapter.

The examples in this book may be duplicated on your machine by downloading and unzipping the chapters. With the exception of the Encoding module used in the custom pipeline discussed in Chapter 19, we’ve included complete code fragments in the text. Aside from that exception, downloading the code is not necessary to understanding the information in this book.
Contacting the Authors

We welcome comments, criticism, corrections, and alternate methods. We also welcome suggestions for examples or areas we’ve overlooked. Jim and John can be contacted at the following e-mail addresses:

*Jim Dawson:* jim@sspsi.com  
*John Wainwright:* john@sspsi.com

You can also reach us by snail mail at

Second Star Professional Services, LLC  
7008 Robbie Drive  
Raleigh, NC 27607

You can visit [www.sbsp.com](http://www.sbsp.com) for more information about Second Star Professional Services.
PART 1

Basic Concepts of BizTalk Mapping
This chapter introduces the basics of mapping with BizTalk, including a high-level overview of the Visual Studio tools with which you will work when creating maps. You will create a simple mapping project in Visual Studio and add to that project a source schema and a target schema. You will then create a map that moves the data from a set of address records in the source schema to the more compact structure of the target document. Finally, you will test the map.

If you are an experienced BizTalk developer, you will already be familiar with the information discussed in this chapter. However, if you are transitioning from other mapping tools to BizTalk and have neither BizTalk nor general development experience, this chapter is for you.

Using the Development Studio

The Visual Studio will be your home for your mapping activities, so as a preface, we’ll do a quick tour of the various Visual Studio windows that you will use most frequently. Remember that, in Visual Studio, you can set most windows to float. Alternatively, you can dock them in several optional locations. The positions of the windows in our figures reflect how we happen to like them set up. You should arrange them in the positions that work best for you.

The Mapper Grid

The mapper grid shown in Figure 1-1, which has the Source Schema on the left and the Destination Schema on the right, is the main working window of the map editor. Each map may have up to 20 mapper grids, and each grid has one tab at the bottom. In the figure, there are two tabs, Page 1 and Page 2, thus this map will be spread over two grids.
CHAPTER 1  CREATING A SIMPLE MAP

Figure 1-1. The Visual Studio map editor

Partitioning the map into grids is especially useful in reducing the clutter. You don’t want the map to become so densely populated with functoids and links that you can’t distinguish one item from another. By using multiple grids, you split the visual representation of the map onto more than one page. This technique allows you to focus on part of the map without having other parts of the map interfere with your view.

Note Mapper grids are self-contained when creating a map. For example, a link cannot begin on one grid page and end on another grid page. Thus you can work on only one grid at a time. The grids are invisible to the compiler, so organizing the map into many grids has no impact on how the map functions.

Figure 1-2 shows how to create a new grid. Right-click the existing tab, and click Add Page.
You can rename a new or existing grid tab by right-clicking the tab you wish to rename and replacing the existing text with your own text. Figure 1-3 shows the Rename Page selection.

Keep in mind that even when you have only one grid page, you still see only a very small percentage of the map. You can right-click the grid and select Grid Preview to see the entire grid for the current tab. Figure 1-4 shows one entire grid in the Grid Preview window.
The cross-hatched rectangle shows the area that will be displayed in the grid window when you close the grid preview. Links are not show in the preview window, but functoids are shown. To change the grid window view to show a particular functoid, drag the green box to the location of the functoid, and close the Grid Preview window. The functoid will now appear in the window.

**Tip** When you are looking at a mapper grid and the grid appears to be empty, you should check the Grid Preview window to see if there are functoids outside the current viewing area.

---

**The Solution Explorer Window**

The Solution Explorer shown in Figure 1-5 contains the current solution on which you are working along with all the projects and project files included in that solution. When working with maps you are mostly concerned with map files and schema files.

![Solution Explorer](image)

**Figure 1-5. The Solution Explorer window**

The Solution Explorer is a good window to keep handy. You use this window frequently during map development when you use incremental testing procedures. This is where you set parameters for map testing and start a map test. Testing is covered in detail in Chapter 3.

**The Toolbox Window**

The Toolbox window provides access to all of the functoids that you use in your map. The screen shot in Figure 1-6 shows you a standard display of the toolbar.
In BizTalk, you can customize the toolbox. By default, the General tab is empty. We use a subset of all the functoids more often than most and like to have the ones we use frequently in one location. We customize our toolbox by dragging and dropping selected icons into the General tab; our General tab is shown in Figure 1-7.

![Figure 1-6. The standard Toolbox window](image)

![Figure 1-7. Our Toolbox window's custom General tab](image)
You move an icon to a different tab using simple drag and drop. As you see, we selected functoids from the Advanced, Logical, String, and Cumulative Functoids tabs and moved them to the General tab. We also moved the Advanced Functoids menu from the top to the bottom, another drag-and-drop effort.

**Tip** The bulk of mapping is done with a small selection of the functoids. Moving the ones that you use most into one location, such as the General tab, will lessen the work you have to do to place functoids onto the mapper grid.

The Properties Window

The last menu you use extensively in mapping is the Properties window. Here, you can see the properties of whichever object you have currently highlighted in the mapper grid or in the source or destination schema windows. Figure 1-8 shows the Properties window as it appears when we highlight the root node of our target schema.

![Properties Window](Image)

**Figure 1-8. A sample Properties window**

The Working Environment

The working environment for mapping has lots of windows and takes up quite a bit of space on your monitor screen. Figure 1-9 is a screen shot of the way John arranges his windows for optimum use while he maps. This arrangement provides immediate access to all of the principal windows you use when creating maps.
Figure 1-9. One way to arrange your working screen

Where John likes to keep his windows visible all the time, Jim uses a different arrangement and prefers to hide windows when the cursor is not over them. You should arrange your screen in the manner that fits your working methods.

Creating the Map

Next, we will build a map from scratch, beginning with the creation of the project and solution. We’ll start with a blank slate and go step by step—creating a new project in the development studio, importing schemas into the project, building a map, and testing the map.

Creating a BizTalk Project

Your first step is to open the development studio so that you can create a new BizTalk project. If you want your project to be placed under any particular directory structure, you should create that structure first, but do not create the project directory. The development studio will do that for you when you create your project.

In the development studio, go to the File drop-down, and select New ➤ Project. Fill in the window as shown in Figure 1-10.
CHAPTER 1  CREATING A SIMPLE MAP

10

Figure 1-10. Creating a new project and solution

When you click OK, the development studio creates the directory tree C:\ProMapping\SampleMaps\Chapter_01. SampleMaps is the solution directory; Chapter_01 is the project directory. Over the course of the rest of this book, we will build a project for each chapter, and those projects will all fall under this solution.

Caution  Don’t consider the solution and project structure we use in this book a recommended method. Our standard practices for this are much different. For example, normally, we create a separate project for each map. We’ve cheerfully violated our normal practices in this book to make loading projects for a specific chapter easier for you.

Once you have created the solution, click the Save All icon to save your work.

Adding a Project File Structure

Your basic mapping project and the default directories are built during the saving process. Inside your Chapter_01 directory, you will find some other default directories created by the development studio. There are two, bin and obj, in the Chapter_01 directory. Each of those has several subdirectories as well, all used by the development studio for building and deploying your project.

We want you to add some folders to the Chapter_01 directory. We use this setup because it works for us in this book. You will develop your own preferences as to how best to organize your folders. Let’s add some subdirectories to Chapter_01. You can see them in Figure 1-11.