THE EXPERT'S VOICE® IN SQL SERVER

# Practical Maintenance Plans in SQL Server

Automation for the DBA

—

Bradley Beard

Apress®

# Practical Maintenance Plans in SQL Server

## Automation for the DBA



Bradley Beard

Apress®

***Practical Maintenance Plans in SQL Server***

Bradley Beard
Palm Bay
Florida, USA

ISBN-13 (pbk): 978-1-4842-1894-5 ISBN-13 (electronic): 978-1-4842-1895-2
DOI 10.1007/978-1-4842-1895-2

Library of Congress Control Number: 2016938532

Copyright © 2016 by Bradley Beard

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Lead Editor: Jonathan Gennick
Development Editor: Douglas Pundick
Technical Reviewer: Mike McQuillan
Editorial Board: Steve Anglin, Pramila Balen, Louise Corrigan, Jim DeWolf, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing
Coordinating Editor: Jill Balzano
Copy Editor: Kim Burton-Weisman
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springer.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Printed on acid-free paper

*For you, my dearest Jessica*

# Contents at a Glance


About the Author ..... xiii

About the Technical Reviewer ..... xv

Acknowledgments ..... xvii

Introduction ..... xix

■ Chapter 1: Introduction to Maintenance Plans ..... 1

■ Chapter 2: Backing Up a Database ..... 7

■ Chapter 3: Checking Database Integrity ..... 33

■ Chapter 4: Executing SQL Server Agent Jobs ..... 45

■ Chapter 5: Cleaning Up SQL Server Agent History ..... 93

■ Chapter 6: Cleaning Up Maintenance Files ..... 107

■ Chapter 7: Rebuilding Indexes ..... 131

■ Chapter 8: Reorganizing Indexes ..... 147

■ Chapter 9: Shrinking the Database and Files ..... 155

■ Chapter 10: Updating Object Statistics ..... 165

■ Chapter 11: Executing T-SQL Statements ..... 175

■ Chapter 12: Notifying Database Operators ..... 189

■ Chapter 13: Tying It All Together ..... 205

Index ..... 269

# Contents

**About the Author** ..... xiii

**About the Technical Reviewer** ..... xv

**Acknowledgments** ..... xvii

**Introduction** ..... xix

**Chapter 1: Introduction to Maintenance Plans** ..... 1

Before We Start ..... 1

What Is a Maintenance Plan? ..... 2

Maintenance Plan Wizard Task Options ..... 3

Maintenance Plan Design Surface Options ..... 4

Summary ..... 6

**Chapter 2: Backing Up a Database** ..... 7

Recovery Models ..... 7

Full ..... 9

Bulk Logged ..... 9

Simple ..... 9

Backup Types ..... 9

Full Backups ..... 10

Differential Backups ..... 10

Transaction Log Backups ..... 10

Setting Up the Maintenance Plan ..... 12

Full Backup Configuration ..... 12

Differential Backup Configuration ..... 18

Transaction Log Backup Configuration ..... 21
Configuring the Jobs ..... 25
Summary ..... 31
**Chapter 3: Checking Database Integrity ..... 33**
What Is Database Integrity? ..... 33
Practical Application of Database Integrity Principles ..... 34
Setting Up the Maintenance Plan ..... 36
Summary ..... 43
**Chapter 4: Executing SQL Server Agent Jobs ..... 45**
E-mail from the Database ..... 45
Configuring E-mail ..... 46
Sending a Test E-mail ..... 56
Enabling the Mail Profile ..... 57
SQL Agent Job Creation ..... 59
Creating an Example Table ..... 60
Writing a Query for the Job ..... 60
Creating a SQL Agent Job ..... 61
SQL Server Agent Options ..... 62
Steps Tab ..... 63
Schedules Tab ..... 68
Alerts Tab ..... 70
Notifications ..... 76
Targets ..... 79
Gmail's SMTP ..... 80
Setting Up a Profile ..... 80
Testing E-mail Configuration ..... 88
Allowing Access to Google's SMTP Server ..... 89
Enabling POP E-mail ..... 90
Summary ..... 91

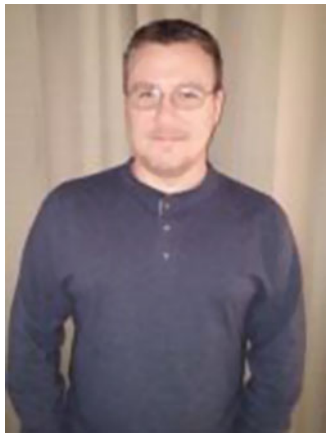
**Chapter 5: Cleaning Up SQL Server Agent History ..... 93**
Setting Up the Maintenance Plan ..... 93
Choosing Tasks ..... 96
Define What to Clean ..... 100
Review ..... 102
Summary ..... 105
**Chapter 6: Cleaning Up Maintenance Files ..... 107**
Separation of History from Maintenance ..... 107
Setting Up the Maintenance Plan ..... 108
Backups Cleanup ..... 108
Deleting Backup Files ..... 113
Deleting a Specific File ..... 113
Searching and Deleting Based on Extension ..... 113
Logs Cleanup ..... 117
Text Files Cleanup ..... 124
Summary ..... 130
**Chapter 7: Rebuilding Indexes ..... 131**
Indexes Explained ..... 131
Beginning Indexes ..... 132
B-Tree Structures ..... 136
Rebuilding vs. Reorganizing ..... 137
Setting Up the Maintenance Plan ..... 137
Summary ..... 146
**Chapter 8: Reorganizing Indexes ..... 147**
Reorganizing vs. Rebuilding ..... 147
Setting Up the Maintenance Plan ..... 148
Summary ..... 154

**Chapter 9: Shrinking the Database and Files ..... 155**
Disk Usage Reporting ..... 155
Disk Space Considerations ..... 156
The Transaction Log ..... 157
Setting Up the Maintenance Plan ..... 157
Summary ..... 164
**Chapter 10: Updating Object Statistics ..... 165**
Distribution Statistics Explained ..... 165
Setting Up the Maintenance Task ..... 166
Summary ..... 174
**Chapter 11: Executing T-SQL Statements ..... 175**
Setting Up the Maintenance Plan ..... 175
Implementing the Maintenance Plan ..... 180
Executing the Maintenance Plan ..... 183
Summary ..... 187
**Chapter 12: Notifying Database Operators ..... 189**
Setting Up the Maintenance Plan ..... 189
Creating an Operator Profile ..... 199
Summary ..... 203
**Chapter 13: Tying It All Together ..... 205**
Checking Your Environment ..... 205
Ordering of the Maintenance Tasks ..... 206
Determining Complexity of the Maintenance Plan ..... 207
Planning the Maintenance Plan ..... 209
Scenario 1 ..... 209
Scenario 2 ..... 210
Creating the Maintenance Plan ..... 213
Editing the Jobs ..... 213
Finalizing the Jobs ..... 214

Saving the Changes ..... 219
Reviewing Your Schedule Needs ..... 220
Adding the Tasks to the Plan ..... 220
Full Backup Maintenance Activities ..... 222
Adding the Check Integrity Task ..... 222
Adding the Rebuild Index Task ..... 224
Adding the Shrink Database Task ..... 225
Adding the Update Statistics Task ..... 226
Adding the Cleanup History Task ..... 228
Adding the bak Files Task ..... 229
Adding the txt Files Task ..... 231
Adding the trn Files Task ..... 233
Differential Backup Maintenance Activities ..... 235
Adding the Check Integrity Task ..... 236
Adding the Reorganize Index Task ..... 238
Adding the Shrink Database Task ..... 239
Adding the Update Statistics Task ..... 240
Transaction Log Backup Maintenance Activities ..... 242
Adding the Check Integrity Task ..... 242
Adding the Reorganize Index Task ..... 244
Adding the Shrink Database Task ..... 245
Adding the Update Statistics Task ..... 247
Precedence Constraints ..... 248
Constraint Options ..... 250
Multiple Constraints ..... 251
Testing the Maintenance Plan ..... 255
Starting a Job from a T-SQL Script ..... 256
Summary ..... 267

**Index ..... 269**

# About the Author



**Bradley Beard** is a software engineer with more than 15 years' experience writing dynamic, interactive web sites using ColdFusion and SQL Server. He graduated from Florida Institute of Technology in 2007 with a master of science in computer information systems, and studied for his undergraduate degrees in CIS and technology management at Herzing University. In 2013, he earned the MCSA: SQL Server 2012 certification from Microsoft, and in 2016, he earned the MCSE: Business Intelligence certification as well. His continual quest for learning has earned him shelves full of books at home and at work, most of which are about SQL Server, ColdFusion, or general web architectures or frameworks.

Bradley lives in Palm Bay, Florida, with his wife, Jessica, and children, Josh, Kaylee, Matthew, and Emma. He also apparently runs an animal shelter made up of his dogs, Lady and Bella, and cats, Spice, Simba, Mercury, and Dobby. In his free time, he enjoys fishing and spending time with his wife and kids.

Bradley is available for consultation and third-shift remote employment on ColdFusion and SQL Server by contacting `bradley.beard@gmail.com`.

# About the Technical Reviewer

**Mike McQuillan** is a software and database specialist who lives with his wife and daughter in the United Kingdom. Mike is a polyglot programmer who began messing around with computers in the 1980s, first with an Atari 800XL and then a Sinclair Spectrum. He took up databases in the 1990s, and quickly fell in love with SQL. He's been working with SQL Server since version 7 and he is an SQL Server MCSA.

When he's not tinkering with computers, Mike and his family enjoy lengthy walks around Cheshire with the family pups, Dolly and Bertie (who keep his feet warm when he's writing).

# Acknowledgments

First of all, a huge thanks to Jonathan Gennick for all the guidance and tips along the way.

A big thanks to my wife and kids for understanding.

Big, huge, gigantic thanks to my good friend and mentor John Wysocki, without whom I would not know half as much as I do about anything related to databases (or chemistry, or plumbing, or nuclear physics… seriously, he knows everything).

Thanks to my other database nerd friends, including Cam, Eric (phpfreak), Leisha, Kyle, Isabelle, and everyone else in the MCSA class, and the guru herself, Suzy Moore. You guys are awesome.

To my friends, brothers, and sisters. I told you I was writing a book and none of y'all believed me.

To my brother Brian and my niece Holly, thanks for being you… even when I was too dumb to appreciate it.

And finally, thank you, Lord, for second chances. I could never deserve your mercy and love.

# Introduction

Have you ever wondered about what keeps your database running smoothly? We all hope the database keeps running as well as it did as when it was first installed, but what really is happening under the covers? More importantly, can we control what is happening? As it turns out, we certainly can control these events, and they are extremely powerful tools that we can use to make sure that our databases stay in good shape.

This book will introduce you to the core concepts of creating a maintenance plan that will handle all of the tasks necessary to keep your database functioning at 100%. There are exercises that you can run in your own installation of SQL Server to ensure heightened data security and integrity by harnessing the power of SQL Server Agent. With the proper amount of foresight and planning, database administrators of any experience level will be confidently using the tools available in SQL Server to create extremely versatile maintenance plans in a short amount of time.

## What Is the Scope of this Book?

In the scope of this book, I am going to concentrate on using the SQL Server 2012 Management Studio interface as much as possible. I am not going to get into a lot of scripting by hand; it will be point-and-click as much as possible. There will be some areas where we will venture into the void, but I'll keep that to a minimum. To be clear, I will be working with the Maintenance Plan Wizard, and not from the Maintenance Plan Design Surface. There will be times when I will focus directly on the design surface, but for the most part, we will work with the wizard. Most of what I detail and outline here is transferable between the two interfaces anyway, as you will soon see.

We are going to make a single maintenance plan that will automatically execute every day on a set schedule. I am going to show you how we can manipulate the database engine into performing the maintenance activities we choose on the schedule we choose. I am also going to show you how to expand on the concept of reporting and logging, from the database point of view, to keep us aware of any possible issues. In the end, we are going to have a maintenance plan that does everything we need it to do, period. This will give us an assistant, so to speak, that runs these menial although important tasks without our supervision. When I think of maintenance plans, I immediately think of the old Ron Popeil commercials with "set it… aaaaaaand forget it!" That's exactly what we do here. We are going to take the time to set it up correctly, and examine the why and not just the how. We are then going to ensure that our reporting and logging is set up correctly, so that we are aware of any issues. After that, it will be smooth sailing and you can enjoy your newly augmented rock star DBA status.

## Who Is this Book Written For?

This book was written for you, of course! It should be a refresher course for the majority of readers. My hope is that this book will augment what you already know, and perhaps shed some light on some concepts that you didn't know. At a minimum, you should come away from this book with a clear understanding of what a maintenance plan is, how it benefits a database, how to structure it correctly, how it runs, and how to possibly make it run more efficiently. I would like to think that this isn't the first you have read about

maintenance plans, as there is a certain level of assumed knowledge about the interface and mechanics of SQL Server. If it is the first you've heard of it, that's fine; I keep everything as simple as possible so that anyone can read along and understand… even my 12-year-old daughter! I try to keep things "light", meaning that I tend to try and put some element of humor into my writing to keep things interesting, but I will cover some pretty heavy topics also, so sometimes it gets serious.

Chapter 1 provides introduction into the separate components of the maintenance plans, and concise examples of what you can do to make the maintenance plan work for you. Chapters 2 through 12 will detail each of the tasks available within the maintenance plan, and Chapter 13 will tie everything together so that you can create a complete maintenance plan based on your specific needs. Along the way, I will examine the multitude of rabbit holes that pop up, because there are actually quite a few little bits of interesting things that need to be discussed. In the end, you will have a complete set of tools and the knowledge to use those tools to achieve great things for your database.

Having said all that, let's dive into this adventure! You can use any installation of SQL Server you would like, provided that you have SSIS installed. That implies any version after and including SQL Server 2005, and it must be Standard, Enterprise, or Business Intelligence versions. Although this book was specifically written with SQL Server 2012 in mind, I also ran everything through SQL Server 2014 and had absolutely no issues beyond slight interface differences.

CHAPTER 1

# Introduction to Maintenance Plans

Ever heard of the faster/cheaper/better paradox? It says that anything can be broken down into three groups: get it faster, get it cheaper, or get it better… but you can only pick two. So something must always be sacrificed: faster and cheaper means it won't be better, and cheaper and better means it won't be faster. Does this strike anyone as a bit unrealistic? Why can't something be all three?

Proper planning can isolate almost any failure. Correctly aligning resources can diminish nearly any risk. At the heart of database administration is the ever-present challenge to provide these three principles: *faster* performance, *cheaper* overhead, and *better* quality data. The failure to provide these for the organization employing you will definitely result in your lack of employment. Most of the time, it's not really as serious as this, but the point can be made that you are the person responsible for the safe handling of the company's most important asset—its data. Whether proprietary or trade data, or government classifications, or even just a simple database holding names and phone numbers, the data that you are responsible for is important to the people that need to access it. For that reason, we as DBAs are the last line of defense to make sure that we are maintaining our databases correctly in order to provide a higher level of database integrity for our end users, whether they are our grandmothers, or stockbrokers, or any other level of user.

The most important part to database administration, beyond installation and actual development, is the maintenance of the database. This ongoing practice should be a part of the daily life of any seasoned database administrator (DBA). Since the DBA is the primary focal point for the database, if it goes poof in the dark, you had better hope it's not your fault. Luckily, SQL Server provides a wealth of tools specifically geared toward giving a lot of power to the DBA in as small a package as possible, with the hope that the DBA uses these tools to mitigate any possible risk to the data they are responsible for.

After all, the primary responsibility of the DBA is to ensure protection, integrity, consistency, and availability of their data. Following the instructions in this book and implementing a complete maintenance plan will get you started accomplishing those goals.

## Before We Start

I have a very specific way that I have my files structured. You might or might not, but I wanted to explain it because I frequently reference it throughout this book.

I have a logical E:\ drive that I use for all of my SQL Server files. Not the installation files, just the database files. The root of this drive is E:\ and there is exactly one folder named SQL Server. So, the main directory for all of my database files is E:\SQL Server. Inside of this folder, I have the following directories.

- **Backups**: The .bak files for each backup, stored in folders per database
- **Data**: The .mdf and .ldf files for each database, stored in folders per database
- **Logs**: The .trn files for each database, stored in folders per database

© Bradley Beard 2016
B. Beard, *Practical Maintenance Plans in SQL Server*, DOI 10.1007/978-1-4842-1895-2_1

Everything that I need for SQL Server development is found in these folders. This might not work for your setup, but I hope it does. If it doesn't, just adjust your own particular folder structure to the examples in the chapters. I am going to be adding folders to this structure in later chapters, so read on to find out what they will contain.

You will also need a Windows login with sysadmin rights to the database engine, and a SQL Server login with sysadmin rights as well (usually, your sa account works just fine). These accounts are very common in newer SQL Server installations, so it shouldn't be an issue. The large majority of the book will use the Windows login, as that login typically either owns the database, or has sysadmin permission to modify the database as needed. If you don't have an account that has these permissions, you may not be able to create and execute the plans against the database you need to maintain. In this case, you are really not a DBA and are more like a "data facilitator." As a DBA, you should have one account that gives you complete and total control of the database. Use that account to set up the things in this book.

# What Is a Maintenance Plan?

When you create a maintenance plan, SQL Server creates an Integration Services package that is executed by the SQL Server Agent. A maintenance plan exists for one reason: to make the life of a DBA easier by automating administrative tasks. That's it! With a well-thought-out maintenance plan, you can do all sorts of things automatically on a set schedule.

So, exactly what things can be done in a maintenance plan, you ask? First of all, fire up SQL Server Management Studio and expand the Management section. Expand the Maintenance Plans section next. On a clean install, there won't be anything in here, which is fine. That's why you're reading this, hopefully. You might see what is in Figure 1-1.

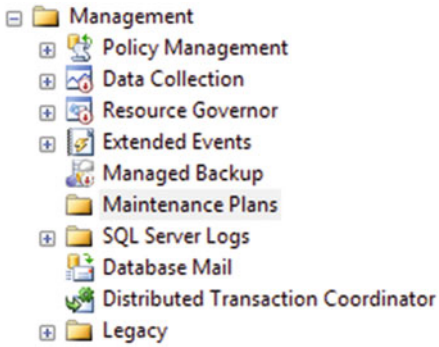



***Figure 1-1.*** *Nothing in the Maintenance Plans folder!*

There are two ways to set up a maintenance plan from here. You can either go with the wizard or create it from scratch yourself by using what is referred to as the Maintenance Plan Design Surface. The differences between the two choices are slight, but there is no real power difference. Let's examine the differences.

Right-click Maintenance Plans and choose Maintenance Plan Wizard. An introduction screen pops up with general information about the wizard, so just click Next to continue.

You are then presented with the first "real" screen of the wizard. Just click Next here, because we're going to cancel it after the next step. This is just to get us familiar with the maintenance plan task choices.

# Maintenance Plan Wizard Task Options

After you hit Next, you are presented with the following options, which can be set up from the wizard interface.

- **Check Database Integrity**: The Check Database Integrity task performs internal consistency checks of the data and index pages within the database.
- **Shrink Database**: The Shrink Database task reduces the disk space consumed by the database and log files by removing empty data and log pages.
- **Reorganize Index**: The Reorganize Index task defragments and compacts clustered and nonclustered indexes on tables and views. This improves index-scanning performance.
- **Rebuild Index**: The Rebuild Index task reorganizes data on the data and index pages by rebuilding indexes. This improves performance of index scans and seeks. This task also optimizes the distribution of data and free space on the index pages, allowing faster future growth.
- **Update Statistics**: The Update Statistics task ensures the query optimizer has up-to-date information about the distribution of data values in the tables. This allows the optimizer to make better judgments about data access strategies.
- **Clean Up History**: The History Cleanup task deletes historical data about Backup and Restore, SQL Server Agent, and Maintenance Plan operations. This wizard allows you to specify the type and age of the data to be deleted.
- **Execute SQL Server Agent Job**: The Execute SQL Server Agent Job task allows you to select SQL Server Agent jobs to run as part of the maintenance plan.
- **Back Up Database (Full)**: The Back Up Database (Full) task allows you to specify the source databases, destination files or tapes, and overwrite options for a Full backup of a database or transaction log.
- **Back Up Database (Differential)**: The Back Up Database task allows you to specify the source databases, destination files or tapes, and overwrite options for a Differential backup of a database or transaction log.
- **Back Up Database (Transaction Log)**: The Back Up Database task allows you to specify the source databases, destination files or tapes, and overwrite options for a Transaction Log backup of a database or transaction log.
- **Maintenance Cleanup Task**: The Maintenance Cleanup task removes files left over from executing a maintenance plan.

Go ahead and cancel that screen now. Go back and right-click Maintenance Plans again, but this time, choose New Maintenance Plan….

The first thing you need to do is give it a name, but you can just click OK here. Again, we're just looking at the options for now. A whole new interface opens with a toolbox on the left Subplan information across the top. This is the Maintenance Plan Design Surface. Figure 1-2 should be what you see now.

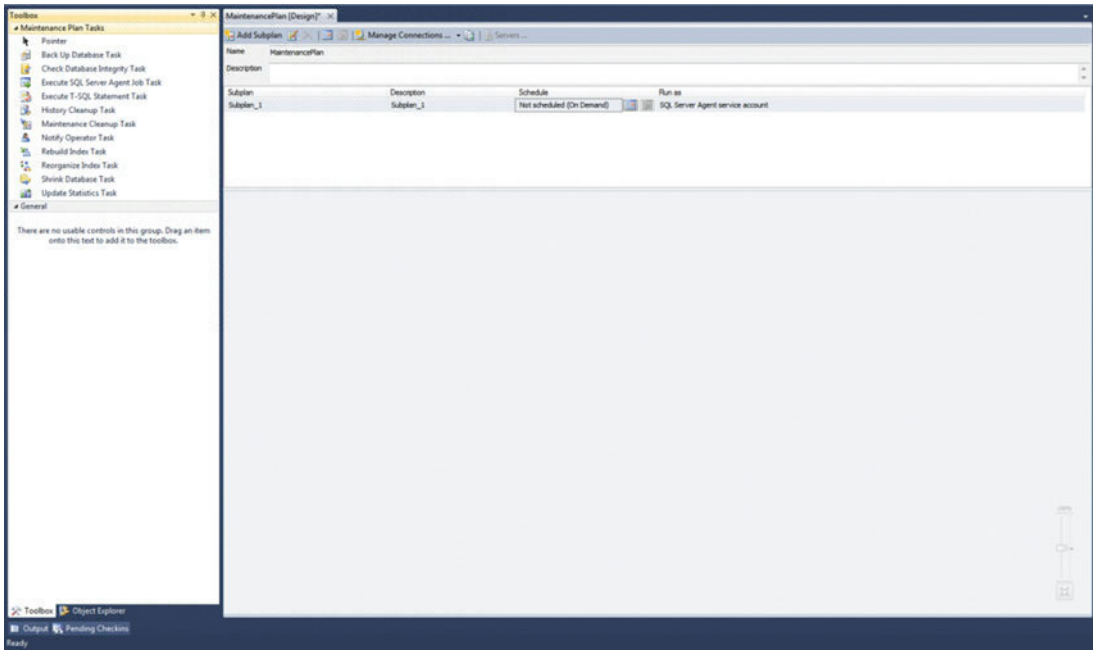



***Figure 1-2.*** *The Maintenance Plan Design Surface*

■ **Tip** If you don't see the Toolbox, just press Ctrl+Alt+X and it will show.

# Maintenance Plan Design Surface Options

If you look in the toolbox on the left, you have some different choices here. I wonder why that is? I will explain why in just a second. Let's look at the options first.

- **Back Up Database**: The Back Up Database task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup of a database or transaction log.
- **Check Database Integrity**: The Check Database Integrity task performs internal consistency checks of the data and index pages within the database.
- **Execute SQL Server Agent Job**: The Execute SQL Server Agent Job task allows you to select SQL Server Agent jobs to run as part of the maintenance plan.
- **Execute T-SQL Statement**: The Execute T-SQL task allows you to run SQL queries as part of the maintenance plan.
- **History Cleanup**: The History Cleanup task deletes historical data about Backup and Restore, SQL Server Agent, and Maintenance Plan operations. This wizard allows you to specify the type and age of the data to be deleted.
- **Maintenance Cleanup**: The Maintenance Cleanup task removes files left over from executing a maintenance plan.

- **Notify Operator**: The Notify Operator task allows for email to be sent from SQL Server after the execution of a maintenance plan. This is for both success and failure.
- **Rebuild Index**: The Rebuild Index task reorganizes data on the data and index pages by rebuilding indexes. This improves performance of index scans and seeks. This task also optimizes the distribution of data and free space on the index pages, allowing faster future growth.
- **Reorganize Index**: The Reorganize Index task defragments and compacts clustered and non-clustered indexes on tables and views. This improves index-scanning performance.
- **Shrink Database**: The Shrink Database task reduces the disk space consumed by the database and log files by removing empty data and log pages.
- **Update Statistics**: The Update Statistics task ensures the query optimizer has up-to-date information about the distribution of data values in the tables. This allows the optimizer to make better judgments about data access strategies.

Most of those look familiar, because they are mostly shared between the interfaces. What are the differences, you ask? Let's take a look and see. Figure 1-3 compares the two types of tasks.

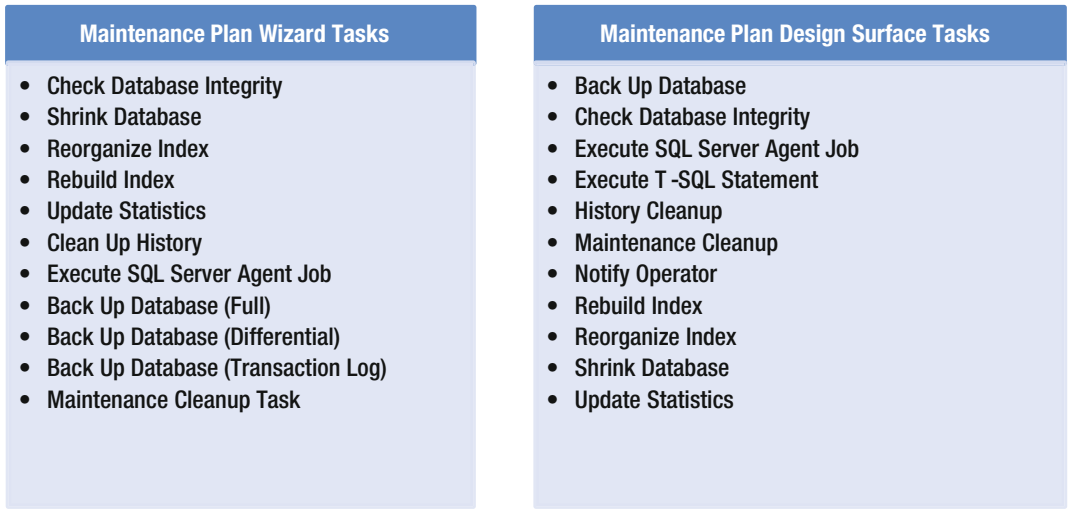



***Figure 1-3.*** *Differences between the wizard and Design Surface Task*

There are 11 tasks in each area… but they are not the same. The difference is, basically, how you want to define the requirements for the workflow that you are creating as part of the maintenance plan. There really isn't a major difference between the two sections, as you can see in Figure 1-3. Or is there?

The first difference is the lack of the Execute T-SQL Statement task in the wizard, but if there is SQL that needs to be executed, it can easily be worked into the Execute SQL Server Agent Job task. Other than that, they are pretty much the same.

The second difference is the lack of the Notify Operator task in the wizard. You can always set the notifications in the tasks themselves though, so this really isn't an issue either.

The 11 tasks shown in Figure 1-3 (in either iteration, really) are at the very core of database administration. Please note that all of these tasks individually can be done manually within the SQL Server Management Studio interface. It's not like this is the only place you can find this functionality. Quite the opposite; true to Microsoft standard operating procedure, there are always at least two ways to do any one task.

It is also worth pointing out that any number of these tasks can be joined in the same maintenance plan, so you really are free to completely customize the maintenance plan to your exact need and purpose. The process of joining the tasks together to make a cohesive maintenance plan is what I call the "workflow" of the plan. Without this, the first thing would just run and then stop. Nothing else would execute unless explicitly told. So, to get around this, what we do is define constraints on the success or failure of a task; that way, we can at least track the error if something goes wrong.

# Summary

Let's dive into this adventure! You can use any installation of SQL Server you would like, provided that you have SSIS installed. That implies any version after and including SQL Server 2005, and it must be Standard, Enterprise, or Business Intelligence versions. Although this book was specifically written with SQL Server 2012 in mind, I also ran everything through SQL Server 2014 and had absolutely no issues beyond slight interface differences.

CHAPTER 2

# Backing Up a Database

Backing up a database is arguably the most important aspect to database administration. Without backups, you cannot recover in cases of data loss or corruption. You cannot rebuild quickly without the data that you, ultimately, are responsible for. Have you ever met a DBA that is sort of *mehhhh* about backups? I guarantee you that is because they are either lazy or have never had to recover from a catastrophic failure. The worst part about either of those reasons is that one is by choice and one is by circumstance, but both can be remediated with a little foresight and planning.

Don't get me wrong; sometimes, a failure is going to happen. It just will; Murphy's law says so. But that doesn't mean that we can't recover from it successfully, does it? The purpose of this chapter is to explain not only the importance of regular backups, but also the proper way to structure these into a cohesive and complete maintenance plan.

The one thing I want you to take from this chapter is one simple sentence: it makes absolutely no sense at all to NOT have database backups.

As you can guess, there are multiple different facets to database backups though. Let's look at those before I explain the actual task itself.

**Recovery models**: This is how your database is set up to recover lost data. This is typically set during the installation of the database engine, but can be changed at any time.

**Backup types**: Certain types of recovery models allow for certain types of backups. This can also be changed at any time, depending on the settings of the recovery model of the database needing to be backed up.

## Recovery Models

An important note before starting this is that the recovery model for a specific database comes into play here. What is a recovery model? Simply put, it tells the database how to recover and in what way. The recovery model is set in the initial setup and configuration of the database, but can also be accessed by right-clicking the database name and choosing Properties. Choose Options from the menu choices, shown on the left in Figure 2-1, and the second option down will show the Recovery Mode.

© Bradley Beard 2016
B. Beard, *Practical Maintenance Plans in SQL Server*, DOI 10.1007/978-1-4842-1895-2_2

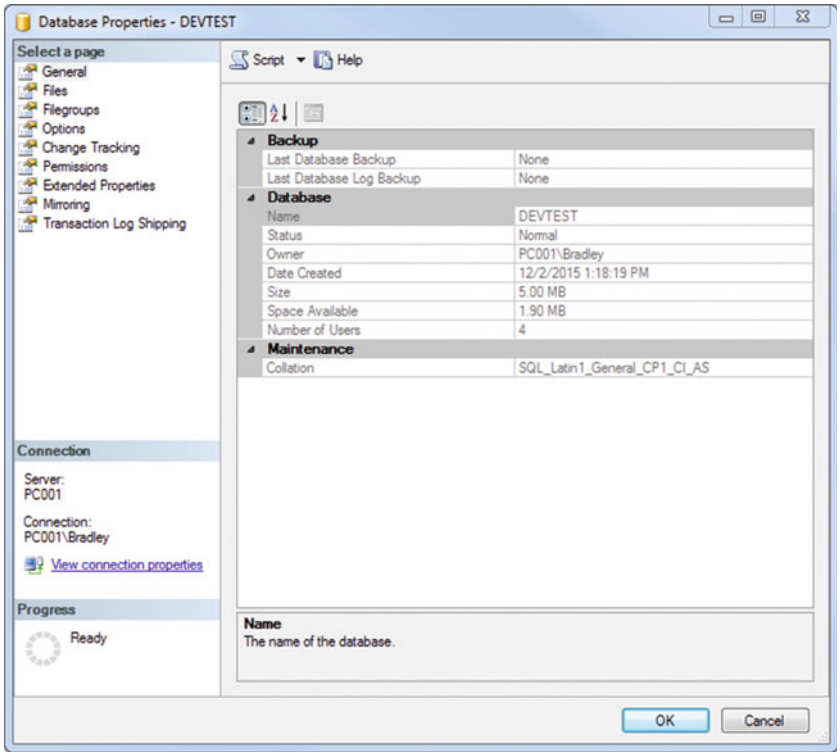



***Figure 2-1.*** *Database Properties window*

Let's look at the differences in recovery models. Figure 2-2 shows the basic differences between the models available, so it's really up to the database administrator to determine the actual need or requirement to properly meet the need of the customer or application.

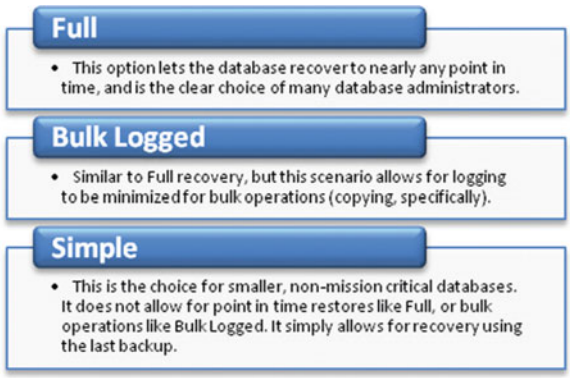



***Figure 2-2.*** *Recovery models*