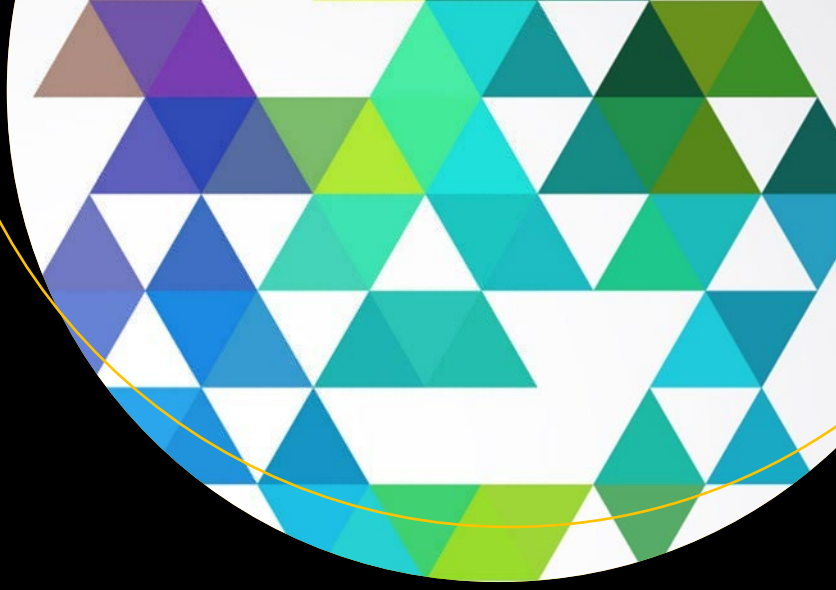


Beginning SQL Server R Services

Analytics for Data Scientists

—

Bringing data science to the database

—

Bradley Beard

Apress®

Beginning SQL Server R Services

Analytics for Data Scientists

Bradley Beard

Apress®

Beginning SQL Server R Services: Analytics for Data Scientists

Bradley Beard
Palm Bay, Florida
USA

ISBN-13 (pbk): 978-1-4842-2297-3 ISBN-13 (electronic): 978-1-4842-2298-0
DOI 10.1007/978-1-4842-2298-0

Library of Congress Control Number: 2016958725

Managing Director: Welmoed Spahr
Lead Editor: Jonathan Gennick
Technical Reviewer: Kathi Kellenberger
Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan, Jonathan Gennick, Todd Green, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing
Coordinating Editor: Jill Balzano
Copy Editor: Kim Burton-Weisman
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

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 materials referenced by the author in this text are 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/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

This book is dedicated to the memory of my late grandmother,
Bessie Dejaynes, who passed away during the writing of this book.
I love you, I miss you, and I will see you again.

Contents at a Glance

About the Author xiii

About the Technical Reviewer xv

Acknowledgments xvii

Introduction xix

■ Part I: Setup and Installation 1

■ Chapter 1: Setup and Installation of SQL Server 2016 3

■ Chapter 2: Setup and Installation of R Tools for Visual Studio 33

■ Chapter 3: Project Scenario Definition 73

■ Part II: Learning the Basics 87

■ Chapter 4: Building R Models with RTVS 89

■ Chapter 5: Plotting in RTVS 119

■ Part III: Creating and Viewing Reports 131

■ Chapter 6: Configuring SQL Server Reporting Services 133

■ Chapter 7: Report Builder Installation and Data Preparation 149

■ Chapter 8: Building Reports Using Report Builder 181

■ Chapter 9: Viewing the Reports in Report Server 209

Part IV: Appendices 223

Appendix A: Installing a SQL Server 2016 Instance in a SQL Server 2014 Installation 225

Appendix B: Software Requirements Document..... 253

Appendix C: R Plot and Tabular Code for R Tools for Visual Studio 255

Index..... 259

Contents

About the Author xiii

About the Technical Reviewer xv

Acknowledgments xvii

Introduction xix

Part I: Setup and Installation 1

Chapter 1: Setup and Installation of SQL Server 2016 3

Planning 4

Beginning the Installation 4

Product Key 6

License Terms 7

Install Rules 7

Feature Selection 8

Instance Configuration 10

Server Configuration 11

Database Engine Configuration 13

Reporting Services Configuration 17

Consent to install Microsoft R Open 18

Ready to Install 19

Installation Progress 20

Install Complete 20

Services Verification 21

SQL Server Management Tools 22

Summary 32

Chapter 2: Setup and Installation of R Tools for Visual Studio 33
SQL Server Data Tools 33
Visual Studio 34
Download R Tools for VS 39
Download Microsoft R Open 41
Visual Studio Environment 46
Session 47
Plots 48
Data 48
Working Directory 48
Windows 49
Install Microsoft R Client… 49
Change R to Microsoft R Client 49
Microsoft R Products… 49
RTVS Documentation and Samples 49
R Documentation 50
Feedback 50
Check for Updates 50
Survey/News 50
Editor Options 50
Options 51
Data Science Settings 51
Exploring Samples 53
A First Look at R 53
Summary 71
Chapter 3: Project Scenario Definition 73
Scope Creep 73
Project Definition Phases 74
Phase I: Requirements Gathering 74
Spiral Development Process 75

Software Change Request Process 76
Phase II: Initial Interface Design 78
Loading the R Solution 79
Summary 86
Part II: Learning the Basics 87
Chapter 4: Building R Models with RTVS 89
Exploring Samples 89
R Package Manager 96
Plotting in R 100
Linear Regression in R 107
Regression Diagnostics 110
The Model Object 111
Summary 118
Chapter 5: Plotting in RTVS 119
Report 1: Average Wind Speed by Airport ID 119
Importing the Dataset 120
Opening a Script Pane 122
Preparing the Dataset 123
Average Wind Speed by Airport ID (Tabular) 124
Average Wind Speed by Airport ID (Plot) 126
Report 2: Average Temperature by Airport ID (°F) 128
Average Temperature by Airport ID (Tabular) 128
Average Temperature by Airport ID (Plot) 129
Summary 130
Part III: Creating and Viewing Reports 131
Chapter 6: Configuring SQL Server Reporting Services 133
Connecting to an Instance 133
Service Account 135
Web Service URL 136

Database 137
Web Portal URL 138
E-mail Settings 139
Execution Account 141
Encryption Keys 142
Subscription Settings 143
Scale-out Deployment 144
Power BI Integration 145
Summary 148

Chapter 7: Report Builder Installation and Data Preparation 149
Download Report Builder 149
Setup New Database and Tables 158
Importing Weather Data 162
Generating the Binary Data 175
Summary 180

Chapter 8: Building Reports Using Report Builder 181
Report 1: Average Wind Speed by Airport ID 181
Setting up the Report Layout 182
Data Configuration 185
Adding the Dynamic Image 195
Report Body Properties 202
Running the Report 204
Report 2: Average Temperature by Airport ID 206
Summary 208

Chapter 9: Viewing the Reports in Report Server 209

Viewing Reports 209

Managing Reports 211

Properties 212

Data Sources 212

Subscriptions 215

Dependent Items 219

Caching 219

History Snapshots 220

Security 221

Saving Reports 221

Summary 222

Part IV: Appendices 223

Appendix A: Installing a SQL Server 2016 Instance in a SQL Server 2014 Installation 225

Getting Started 225

Product Key 227

License Terms 228

Install Rules 229

Feature Selection 230

Instance Configuration 232

Server Configuration 234

Database Engine Configuration 237

Server Configuration 238

Data Directories 239

TempDB 240

FILESTREAM 242

Reporting Services Configuration 242

Consent to Install Microsoft R Open 243

Ready to Install 244
Complete 247
Summary 251

Appendix B: Software Requirements Document 253

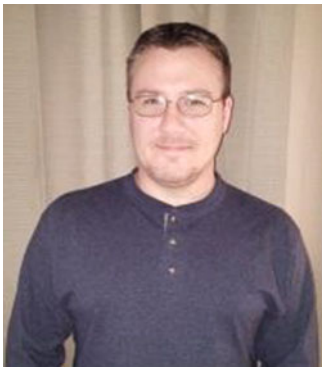
Project 253
Author 253
Prepared for 253
Problem 253
Solution 253
Language/Platform 254
Medium 254

Appendix C: R Plot and Tabular Code for R Tools for Visual Studio 255

Average Temperature by Airport ID (Plot) 255
Average Temperature by Airport ID (Tabular) 256
Average Wind Speed by Airport ID (Plot) 256
Average Wind Speed by Airport ID (Tabular) 257

Index 259

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 the 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. 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.

He 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 him at `bradley.beard@gmail.com`.

About the Technical Reviewer

Kathi Kellenberger, known to the SQL Server community as Aunt Kathi, is an independent SQL Server consultant associated with Linchpin People and a Data Platform MVP. She loves writing about SQL Server and has contributed to more than a dozen books as an author, co-author, or technical reviewer. Kathi enjoys spending free time with family and friends, especially her five grandchildren. When she is not working or involved in a game of hide-and-seek with the kids, you may find her at the local karaoke bar. Kathi's blog is at `www.auntkathisql.com`.

Acknowledgments

Another big thanks to both Jonathan Gennick and Jill Balzano for guiding me through the process of getting this book published. You guys are awesome.

To my first mentor once again, John Wysocki, who is currently enjoying semi-retirement in some lavish resort somewhere: I can't thank you enough for stoking the fire of creativity in me.

To my newest mentor, Suzy Moore, who is hands down the smartest person I know: I can't wait to learn more from you.

To the one and only Chester Flake: Thank you for your guidance on certifications. Anyone needing any sort of certification training needs to go to `www.certificationcamps.com` and sign up today. You won't be sorry.

To my brother Brian, niece Holly, sister-in-law Andie, and Drew, Austin, and Sam: Mom said no fireworks past 10:00 PM!

To my parents Richard and Carolyn, my in-laws Steve and Carey, my out-laws Al and Val, other brothers Joe, Rick, Zimmer, and Dave, and other sisters Morgan, Erika, Jennifer, Kim, and Michelle, and everyone else I forgot.... Well, you know me . . . .

And finally, to my wife, Jessica, and kids—Josh, Kaylee, Matthew, and Emma, who had to deal with me leaving for hours on end while writing this book: Thank you so much.

Oh, yeah.... I can't forget my best fraaaaaaaand Courtney. She's such a little cutie patootie!!

Introduction

In an effort to not sound like a complete Microsoft fan boy, SQL Server 2016 has some seriously cool additions. Not the least of these is the inclusion of a massive data analysis tool widely used throughout the industry. This tool is called, simply, R. Some of you might be asking why Microsoft would possibly include this tool, since it isn't really a database thing as it is an analysis or graphing tool.

The reason, I think, is fairly simple to deduce: Microsoft is expanding their reach. It seems to me that R is a great way to do that. The inclusion of R Tools as a part of the Visual Studio toolset and the SQL Server database instance will most definitely be a game changer for SQL Server development. It used to be where the database developer would have to prepare the data to be consumed by some service for analysis; not anymore. R Tools for Visual Studio (RTVS) allows the user to either prepare their scripts in Visual Studio, or directly in SQL Server Management Studio. Although this isn't recommended by Microsoft, it is still possible.

What We Will Cover

What this book covers is pretty simple and straightforward. We will…

- Set up a new instance on SQL Server 2016
- Set up the necessary R resources to properly create, consume, and execute R
- Briefly review the history, syntax, and functions within R
- Create a custom R solution using R Tools for Visual Studio
- Configure SQL Server Reporting Services
- Install and configure Report Builder
- Create reports in Report Builder based on R code developed in R Tools for Visual Studio
- Consume those reports through Reporting Services

It is important to note a few things at this point. Specifically, …

- R Tools for Visual Studio is a brand-new release, so the chances of it being buggy are pretty good.
- We are fully installing SQL Server 2016 as a completely new instance because I wanted to be able to show the advantages that a user stands to gain by incorporating R into their workflow, even though they may not be completely sold on the benefits of R.

All the components needed to install and configure what is necessary to run R within SQL Server 2016 are covered on MSDN. As of now, the link for the resources is at `https://msdn.microsoft.com/en-us/library/mt604883.aspx`. Please note that Microsoft may not keep that same link forever, so remember that Google is your friend. I will explain what is needed to get it up and running. It may be up to you to find the necessary tools on Microsoft.com.

Why R?

Good question. Why R? The answer is actually quite simple… I think because Microsoft wants to expand their reach into the data science field with SQL Server, so it makes sense that they would want the best product out there for data analysis, which is arguably R. Microsoft's Business Intelligence offerings are already extremely sharp, so if they could find a way to …

- Acquire a data analysis product already in heavy usage
- Incorporate that product into their existing database platform
- Offer a GUI which integrates seamlessly into Visual Studio
- Make it all available *for free*

… then the (data analysis) world would be their oyster! To my knowledge, there isn't any other database system that will allow for such complete interaction with R. There are plenty of instances where data is prepared in a database and then imported to R, but this is a totally different level of interaction.

SQL Server 2016 actually runs an instance of the R engine as native to the database engine. Fascinating! For those of you that have worked with SQL Server when they radically changed to SQL Server 2005, you remember the shock of Integration Services and "what happened to my DTS scripts??" This is on that same sort of level; for those intrigued by data analysis or business intelligence, the implications of the addition of R are guaranteed to be long-reaching and will most certainly result in some amazing advancements in data science. I, for one, can't wait to be a part of that.

Also, just to get this out of the way…

In other words, *thank you, Microsoft, for letting me use a whole lot of your stuff! I really appreciate it.*

If you read my first book, *Practical Maintenance Plans in SQL Server* (Apress, 2016), then you are already familiar with my writing style. I tend to try and keep the mood a bit light and sometimes quirky, without sacrificing technical content. At times, I might divert down a rabbit hole, but I always have a point at the end. Most of the time, it might take me a while to get there, but I do eventually get there in the end.

Right now—before we get into installing and setting up our environment—is a great time for you to take a few minutes to get familiar with R and what it can do.

Once you've done that, let's begin our journey as data scientists!

PART I

Setup and Installation

CHAPTER 1

Setup and Installation of SQL Server 2016

One of the major updates to SQL Server 2016 is the addition of R as an integral part of the database engine. R began in 1993 as a data analysis language developed by Robert Gentleman and Ross Ihaka at the University of Auckland. It started as a language that could rival the S language in statistical analysis and evolved into arguably the most popular language in the world for statistical computing, data analysis, and machine learning.

With the business world making a major shift toward business intelligence and data analysis, the addition of R as an integral part of SQL Server is a smart business move for Microsoft. Not only is Microsoft introducing new functionality into an already widely accepted platform, but they are also leaving the core of the language intact so that current R statisticians can easily move onto the SQL Server platform to enhance their statistical computing methodology. In the end, this enhances visibility for Microsoft in the business intelligence field, and hopefully, leads to even greater acceptance for SQL Server in everyday data analysis operations.

In 2016, Microsoft bought Revolution Analytics, which is built around R and provides both an open source (Revolution R Open) and commercial (Revolution R Enterprise) development platform for R. Heavy integration of R into existing products is now Microsoft's focus, with the obvious choice of SQL Server and, eventually, Azure. This is an exciting release, because it gives Azure hosted services the opportunity to deliver content based on R computations done in an Azure site or database.

Since R has been added as an installation portion of SQL Server 2016, all we need to do is select the option during installation to add it and then run through some minor configuration tasks.

There are certain things that we need to check and install to make sure that R runs, but I will show all that when we get there. For now, download SQL Server 2016, and then follow along with me on how to install it. It is worth noting that your installation screens may vary slightly from mine, depending on the service packs or if Microsoft decides to change the install screens, but I think that the gist of the content will be the same.

B. Beard, *Beginning SQL Server R Services*, DOI 10.1007/978-1-4842-2298-0_1

Planning

First things first though. Once you download SQL Server 2016, you want to plan out the basics, such as the account that you're going to use and where your default file locations are going to be. If you read my last book, then you know that I have a very particular way in which I organize my file system for SQL Server. For this book, I make a separate logical disk (E:\) with the following folder structure:

- E:\SQL Server
  - Backups
  - Data
  - Logs

So, one main folder, SQL Server, and then three folders inside of that folder to hold the different bits as needed. There can be other folders, such as DTSX or Output, which you can use for other things, but for the most part, those three subfolders inside of the main folder work nicely.

■ **Note** There are other locations that SQL Server wants to place files in during installation; this is fine, since this is how SQL Server wants to categorize the system files to keep everything copacetic. We will have control over our data, logs, and backups in the folders specified earlier.

As far as which account you should use to run the functionality of SQL Server 2016, this should be a no-brainer. It needs to run as if it were a regular database installation, so it needs to have the account assigned that it would normally have. To be clear, assign the same account that you are currently using for whatever version of SQL Server you are running. Most times, this needs to be an administrator in order to install programs.

A quick side-note here: if you haven't read the hardware and software requirements for SQL Server 2016, you probably want to do that. Also, Appendix A covers installation of SQL Server 2016 onto an existing SQL Server 2014 server. If you are running SQL Server 2014 and want to try SQL Server 2016 on the same server, then look to Appendix A for guidance. (But in no case should you ever use a production server to follow along with this book).

Beginning the Installation

Here we go! Double-click the **setup.exe** file in the download folder. You should see what's shown in Figure 1-1.

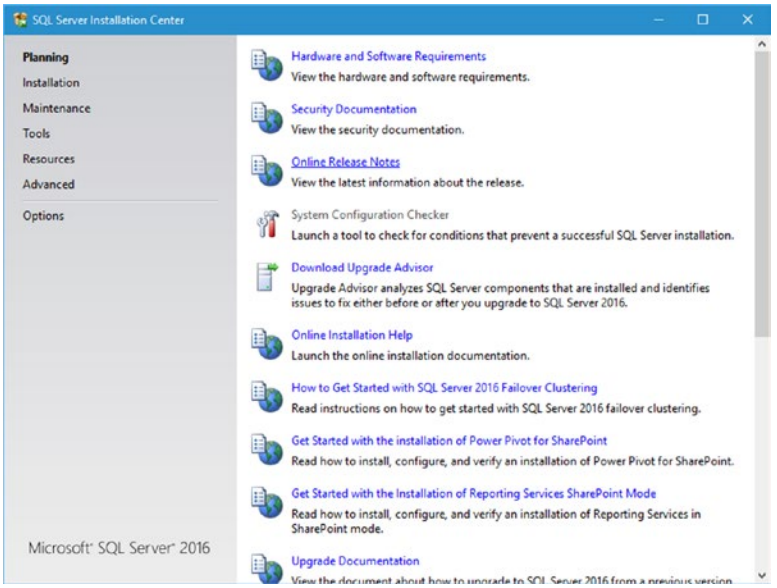


Figure 1-1. *Initial SQL Server 2016 installation screen*

If you see the screen asking to make changes to your system, go ahead and say **Yes**.

Figure 1-1 shows the first screen that you should see when you start installation. This screen should look pretty familiar to you, if you have ever installed SQL Server before. Click the **Installation** link on the left. You should see what is shown in Figure 1-2.

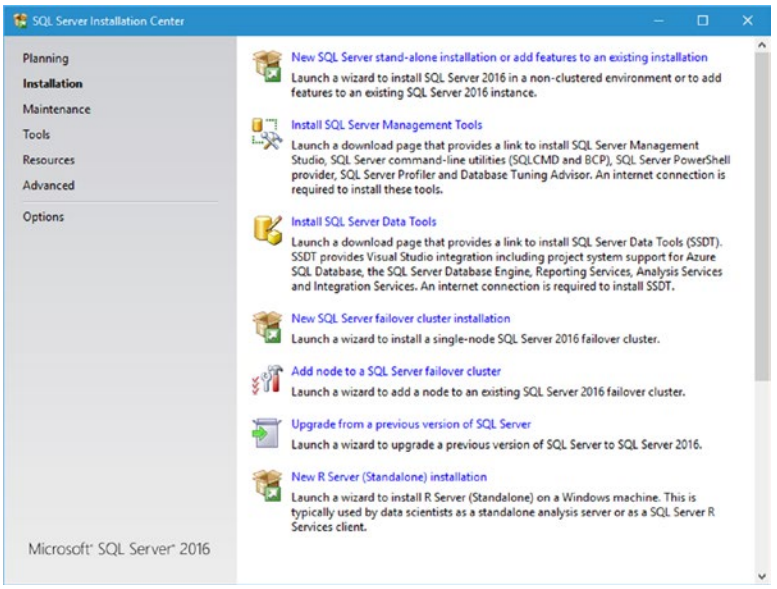


Figure 1-2. *SQL Server 2016 installation options*

Once here, click the top link, titled **New SQL Server stand-alone installation or add features to an existing installation**.

Note that the very bottom option is something new. It says **New R Server (Standalone) installation**. You would select this option if you only wanted to install R Server as either a server (standalone, self-contained data analysis server, in other words) or a client (manipulating data from a remote SQL Server R Services installation). Note that you need the SQL Server 2016 services running as well, so this would be to add R services to an existing SQL Server 2016 installation.

Product Key

Next is to enter your product key. Figure 1-3 shows the screen you see after continuing from Figure 1-2 in the prior section. Here you can specify that you wish to run the free edition, or you may enter a product key in order to run a licensed edition.

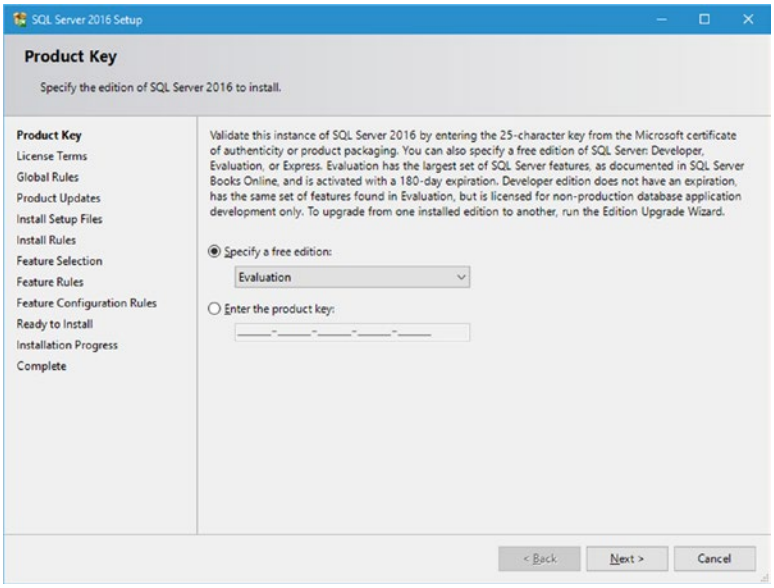


Figure 1-3. *Product Key screen*

SQL Server 2016 can be installed in one of three free editions:

- **Evaluation**: A full set of features; basically, the Enterprise version of SQL Server 2016, but only good for 180-day spans.
- **Developer**: A full set of features, but cannot be used for production database work.
- **Express**: The smallest, bare-bones installation of SQL Server 2016; does not expire and can be used for production use.

If you would like to choose an option other than Evaluation, go right ahead. Just understand the implications of choosing that option; for example, the Express option doesn't support R so I wouldn't choose this option. For what you need here, the Evaluation version is perfect, because you certainly decide within 180 days if this new functionality is something you want to permanently include in your SQL Server installation.

When you have chosen the version you are most comfortable with, click **Next** to continue.

License Terms

The next screen, shown in Figure 1-4, simply asks you to accept the license terms.

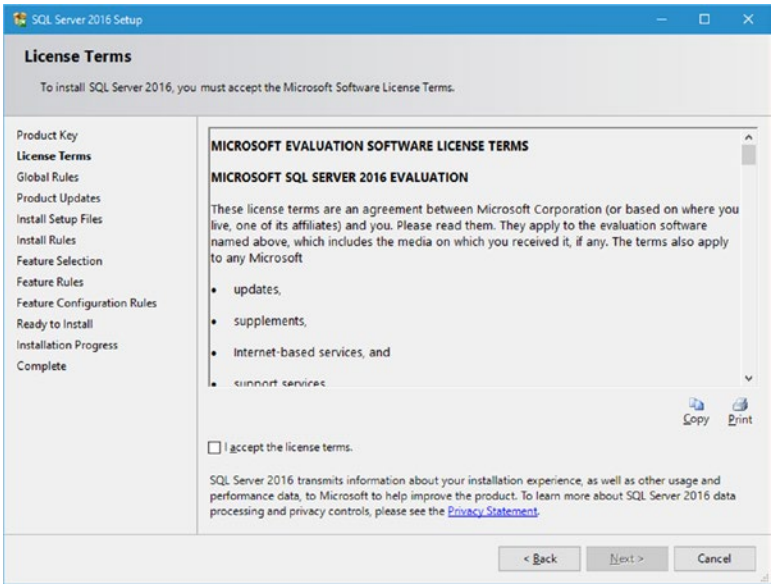


Figure 1-4. *License terms*

I honestly have never read this license all the way through. I can't say that I know anyone who has. Obviously, just click the **I accept the license terms** check box and then click **Next** to move on.

Install Rules

This screen shows you what happens as SQL Server goes through the preliminary steps to check for a clean installation. If you get any errors or warnings, you should look at correcting them so that you can install as cleanly as possible.

My screen flashed a few times and I eventually ended up at the screen shown in Figure 1-5.

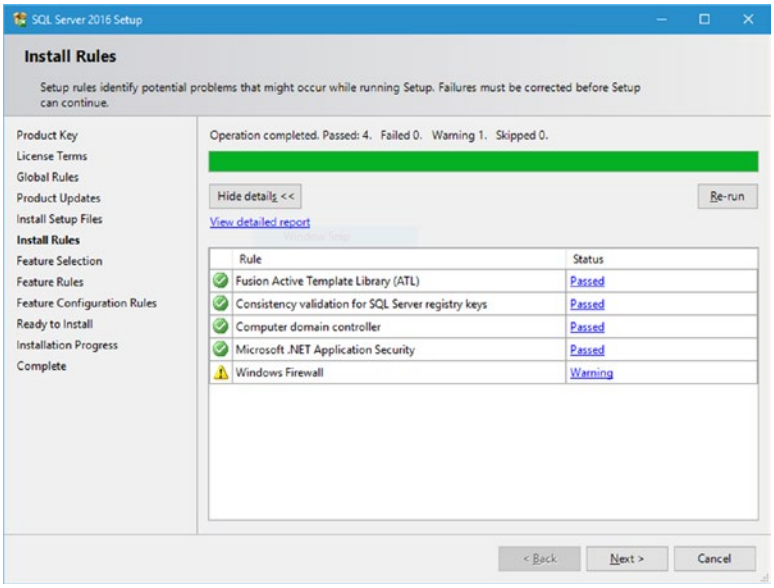


Figure 1-5. *Install Rules*

It's worth noting that there could be an update to SQL Server 2016 that gets downloaded and installed during this step, so if a message comes up with that information, go ahead and install it.

So everything looks good except for my firewall rule. Since no connections come from a network on my laptop, this should be fine, so I'm going to click **Next** to continue.

Feature Selection

Now we get to choosing what we want as part of the actual installation. Figure 1-6 shows the screen we have been waiting for.

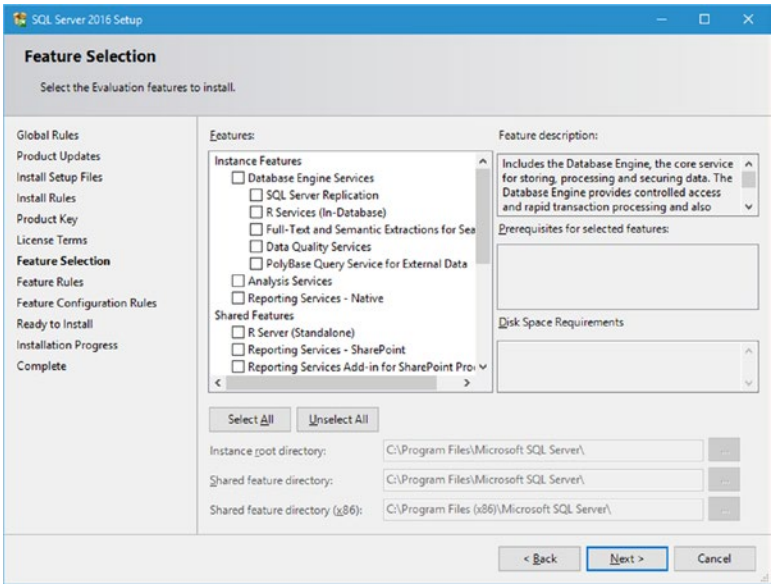


Figure 1-6. *Feature Selection*

At this point, we need to choose just the bare minimum of what we need to test the functionality of R within our database instance. In this book, we get familiar with R and create charts using R Tools for Visual Studio, and then duplicate those results in SSMS, ultimately serving those results in reports through Reporting Services. Because of this, we only install R Services (In-Database) and Reporting Services – Native. This gives us everything we need to really get a feel for R and what it can do for us. It also means that we don't need to install the entirety of SQL Server 2016. Figure 1-7 shows the selected options that you should have at this time.

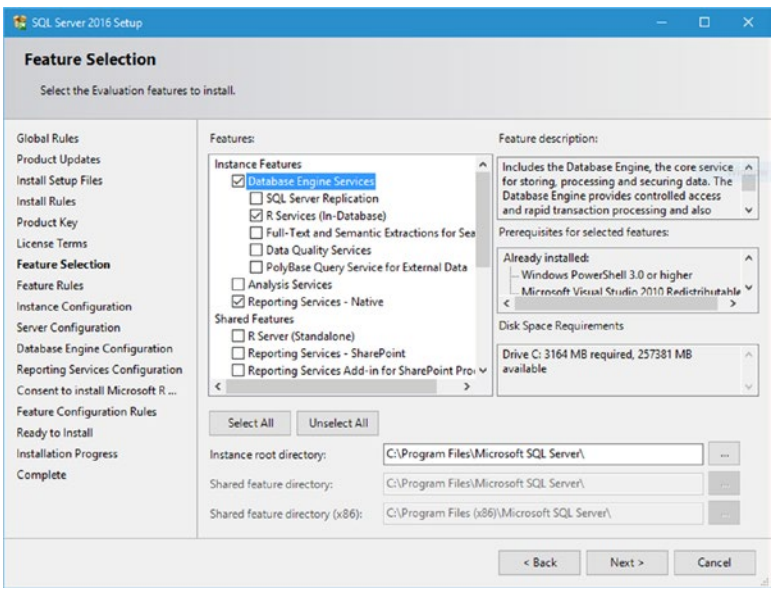


Figure 1-7. *Selected options*

Click **Next** here to move on. It takes a second to think about what it wants to do, but eventually, you see the **Instance Configuration** screen shown in Figure 1-8.

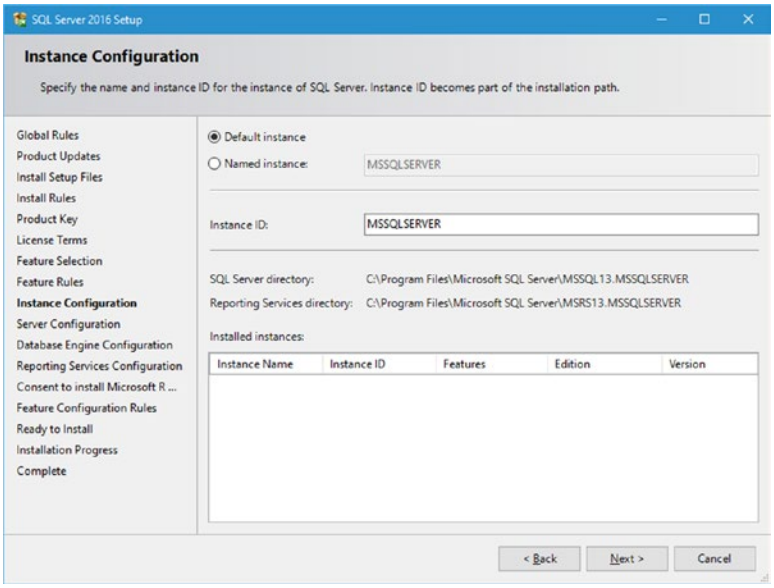


Figure 1-8. *Instance Configuration*

Instance Configuration

Since this is a new installation and there isn't a previous version of SQL Server installed, the option is available for Default Instance. We can certainly do this with no issue, but I usually prefer to name my instances. I leave this up to you, but understand that I will use a Named instance and not a Default instance for the remainder of this book.

At this point, we need to define our new instance. If you look on the **Installed Instances** section, you see that there is nothing there. We choose the **Named Instance** option and call it SQL2016RS for SQL Server 2016 R Services. The Instance ID field should be updated to SQL2016RS as well. Once you do that, you see what is shown in Figure 1-9.

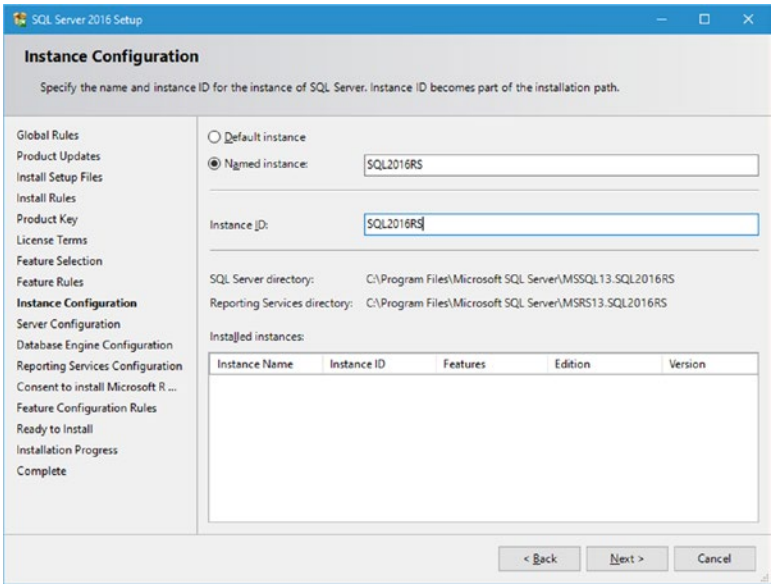


Figure 1-9. *Updated Instance Configuration screen*

Pay attention to the **Named Instance** field, the **Instance ID** field, the **SQL Server directory** location, and the **Reporting Services directory** location listed on this screen. Those need to all have SQL2016RS referenced in them. Once you are satisfied that everything is as it should be, click **Next** to continue.

When you are ready, click **Next** to move on.

Server Configuration

The next screen is where we define the service accounts and startup types for the services. This screen is shown in Figure 1-10.

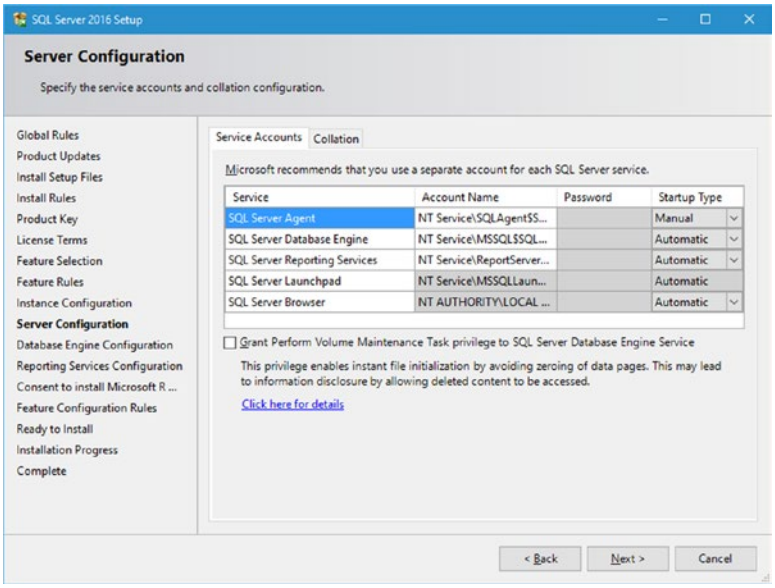


Figure 1-10. *Server Configuration*

The only service to really pay attention to here is the SQL Server Launchpad service. This service handles the execution of R within the database engine, so if R isn't behaving as expected, check this service first.

The SQL Browser service is running under the context of local services, so there isn't a new service account being created. We won't worry about that one, in other words.

These service accounts are the default, but can always be changed to your own service accounts, if you have them. If you don't have your own service accounts, you can keep these suggested service accounts. I know a lot of server administrators that insist on employing the principal of least privilege for services, so if that is the case for your particular environment, then you need to get the service name and login information from the server administrator in order to proceed. Another way you can go about this is to copy these service names and include them in a summary to your system administrator regarding the accounts that were created during installation, so that the system administrator can audit the permissions for this user as needed. It is important to note here that I am referring to a separate individual or entity for "system administrator" that is not a database administrator, but rather the Windows-level administrator. The person in charge of the operating system level, one step up from the Application layer, in other words.

We only want to change a little bit here; specifically, set the SQL Server Agent service **Startup Type** to **Automatic**. That is the only change we need to make. Figure 1-11 shows what you should see at this point.