



Quick answers to common problems

IBM Cognos TM1 Cookbook

Build real world planning, budgeting, and forecasting solutions with a collection of over 60 simple but incredibly effective recipes

Ankit Garg

[PACKT] enterprise 
PUBLISHING professional expertise distilled

IBM Cognos TM1 Cookbook

Build real world planning, budgeting, and forecasting solutions with a collection of over 60 simple but incredibly effective recipes

Ankit Garg

[PACKT] enterprise 
PUBLISHING professional expertise distilled

BIRMINGHAM - MUMBAI

IBM Cognos TM1 Cookbook

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: December 2011

Production Reference: 1081211

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84968-210-7

www.packtpub.com

Cover Image by Sandeep Babu (sandyjb@gmail.com)

Credits

Author

Ankit Garg

Project Coordinator

Jovita Pinto

Reviewers

Wim Gielis

Abhishek Sanghani

Proofreaders

Mario Cecere

Kelly Hutchinson

Acquisition Editor

Amey Kanse

Indexer

Hemangini Bari

Development Editor

Hyacintha D'Souza

Graphics

Manu Joseph

Technical Editors

Kedar Bhat

Kavita Iyer

Production Coordinator

Alwin Roy

Cover Work

Alwin Roy

About the Author

Ankit Garg is a software professional having over seven and a half years of experience in implementing, designing, and managing data warehousing and BI solutions for industry leaders. He has catered to multiple clients, spanning various geographies and industry domains. His core skills include a full set of BI and DW skills, which includes ETL, BI, Data Warehousing, and Predictive Analysis.

Currently, he is involved in product R&D and is employed by a software major.

The book is the result of his technical experience acquired, while implementing planning and budgeting solutions for the industry.

I would like to thank the PACKT team and Editors for their full faith on my authoring and technical skills. They played a pivotal role in bringing the book to the current shape.

Dedicated to my mother and special thanks to my friend Abhishek for the initiative he has taken, for me to get started.

All the best to dear readers, really hope they find the stuff useful and worth reading.

About the Reviewers

Wim Gielis, is currently a Business Intelligence consultant at Axis Belgium. He is responsible for implementing custom IBM Cognos TM1 applications as well as providing (custom) TM1 training sessions.

He regularly posts articles written in English on IBM Cognos TM1 on his blog: <http://www.wimgielis.be>. He is also very active on several online message boards regarding TM1 (such as <http://www.tm1forum.com>), and Excel VBA (such as www.mrexcel.com, www.ozgrid.com, and various others).

Wim Gielis was awarded the Microsoft Excel MVP award in 2011 for exceptional contributions to the Excel community.

He can be contacted at wim.gielis@gmail.com.

Abhishek Sanghani was born in India and attended Mumbai University where he majored in Computer Engineering. He began his career in 2004 as a Business Intelligence and Cognos Consultant, and has worked with leading IT and Finance Services companies since then.

He pursued Finance Management degree along with his work in the field of Cognos and BI, successfully progressing and winning awards and certifications year after year. Presently, he is working in the United Kingdom, utilizing his skills in Cognos, SQL, BI and Data Warehousing. In his free time, he writes technical blogs and also provides training/seminars on demand.

He wrote his first book *IBM Cognos 8 Report Studio Cookbook* with Packt Publishing that covers many basic to advanced features of Report Authoring. It begins by bringing readers on the same platform and introducing the fundamental features useful across any level of reporting. Then it ascends to advanced techniques and tricks to overcome Report Studio 8 limitations.

He thinks that this book is "a wonderful compilation of knowledge and techniques to work on TM1 which will prove very useful to the readers."

It was a great pleasure to review this book. Thanks to Ankit for producing such useful literature and to Packt for giving me this opportunity to review. I hope my feedback proved useful.

I would also like to thank my lovely wife Dolly for all her support.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- ▶ Fully searchable across every book published by Packt
- ▶ Copy and paste, print and bookmark content
- ▶ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Instant Updates on New Packt Books

Get notified! Find out when new books are published by following [@PacktEnterprise](https://twitter.com/PacktEnterprise) on Twitter, or the [Packt Enterprise Facebook page](#).

Table of Contents

Preface	1
Chapter 1: Getting Started with TM1	7
Online Analytic Processing (OLAP)	9
OLAP data structures terminology	13
OLAP (Online Analytical Processing) versus OLTP (Online Transactional)	13
Introduction to TM1	15
Examine TM1 user roles	16
TM1 architecture	17
Installing TM1 server and clients on a machine	23
Creating a new TM1 Data Server	28
Connecting to an existing TM1 Server	32
Setting up TM1 Data Server to run as a Windows service	35
Summary	37
Chapter 2: Creating dimensions in TM1	39
Introduction	39
Creating dimensions manually	43
Creating a Months dimension	50
Adding additional levels in Months dimension	54
Adding Next and Previous attributes to the Months dimension	58
Creating dimensions using the Turbo Integrator	62
Execution of scripts for a TM1 process	69
Chapter 3: Building Cubes and Views	73
Introduction	73
Creating a static pick list	83
Creating the Sales_Plan cube	92
Creating a subset pick list	98

Chapter 4: Loading and Maintaining Data	107
Introduction	107
Loading data using Turbo Integrator processes	109
Using formulae in variables	115
Clearing data from a cube using a process	128
Accumulating data	132
Saving data	134
Chapter 5: Adding Business Rules	137
Introduction	137
Creating simple rules	138
Using a rule to override aggregation	148
Creating a rule to share data between cubes	157
Creating a pick list using a rule	161
Reviewing a Rules worksheet	167
Doing custom calculations in dimensional hierarchies	170
Chapter 6: Automating Data Load Processes using Scripts	173
Introduction	173
Loading data from multiple columns into a single column	175
Creating a dimension and a subset in Turbo Integrator	184
Creating a dimension with uneven/unbalanced hierarchy	188
Exporting data to ASCII text using Turbo Integrator	196
Moving data between versions	199
Moving data using scripts and parameters	205
Creating and scheduling a chore	212
Chapter 7: Customizing Drill Paths	215
Introduction	215
Creating a drill-through path to a view in another cube	216
Creating a drill path to an ODBC source	223
Chapter 8: Using Rules for Advanced Modelling	229
Introduction	229
Creating the Expenses cube	229
Examining a Spread Profile cube, Lookup cube, and Virtual cube	249
Moving balances from last month to next month	266
Chapter 9: Converting Currencies	279
Introduction	279
Creating currency dimension and attributes	280
Creating a currency cube	285

Populating currency attributes using control cubes	289
Creating rules to apply exchange rates and calculate an equivalent amount in target currency	293
Chapter 10: Modelling for Different Fiscal Requirements	299
Introduction	299
Adding a year dimension	300
Creating a continuous time model	307
Chapter 11: Optimizing Rules Performance	325
Introduction	325
Adding SKIPCHECKS and FEEDERS	329
Troubleshooting rules and feeders	333
Chapter 12: Working with Managed Planning Applications	339
Introduction	340
Installing and configuring IBM Cognos TM1 Contributor components	342
Installing Apache Tomcat 6.0.18 for IBM Cognos TM1 Contributor	343
Installing IBM Cognos TM1 Contributor	344
Deploying IBM Cognos TM1 Contributor on Tomcat Server	346
Configuring IBM Cognos TM1 Contributor with Microsoft IIS	348
Running IBM Cognos TM1 Contributor for the first time	348
Installing IBM CognosTM1 Contributor Administration tool	352
Contributing data to managed planning application	354
Introducing IBM Cognos TM1 Contributor workflow screen	357
Navigating the TM1 Contributor workspace	365
Opening a node and modifying a view	368
Entering, editing, and distributing data	378
Reviewing data	387
Creating and deploying managed planning applications to the Web	392
Setting security and access rights	400
Assigning security to managed planning applications	401
Chapter 13: Defining Workflow	409
Introduction	409
Adding users and groups to the TM1	410
Security	410
Installing IBM Cognos TM1 Workflow	416
Adding a new process and task using IBM Cognos TM1 Workflow	418
Reviewing submitted tasks	441

Chapter 14: Integration with IBM Cognos BI	451
Introduction	451
Integrating IBM Cognos TM1 with IBM Cognos 8 BI	455
Index	469

Preface

IBM Cognos TM1 is one of the most popular multidimensional analysis tools used to build collaborative planning, budgeting, and forecasting solutions, as well as analytical and reporting applications. This cube-based technology does fast number crunching at the server end and helps end customers move away from manual and tedious planning, budgeting, and analytics processes.

IBM Cognos TM1 Cookbook is a complete manual to building and managing applications with IBM Cognos TM1.

Every important aspect of TM1 is covered with the help of practical recipes, which will make you well acquainted with the tool and ready to take on TM1 projects in the real world. Existing TM1 users and developers will also benefit from the practical recipes covered in the book.

The book will start with a chapter focusing on fundamentals of MOLAP and dimensional modelling, which forms the foundation on which IBM Cognos TM1 is based. Thereafter, chapters will be more TM1-specific, starting with an introduction on dimensions and cubes and how to create and manage them. Further chapters will dig deeper into TM1 objects and rules.

Then we will learn the more advanced features of TM1 such as automating common processes using scripts, customizing drill paths, using rules for advanced modelling, converting currencies, modelling for different fiscal requirements, and more.

At the end of the book, we will cover how to present data and reports, workflows, and TM1 application security.

What this book covers

Chapter 1, Getting Started with TM1, introduces multidimensional concept, IBM Cognos Suite of tools, and concepts such as Enterprise Planning, BI, and OLAP. It introduces the idea of IBM Cognos TM1 being a strong OLAP tool. It also familiarizes you with the architecture and installation of IBM Cognos TM1 and setting up example planning application for the first time.

Chapter 2, Creating dimensions in TM1, introduces dimensions in IBM Cognos TM1 and working with them. It extends sample planning application introduced in *Chapter 1* to demonstrate use cases around TM1 dimensions.

Chapter 3, Building Cubes and Views, introduces cubes in IBM Cognos TM1 and working with them. It extends the example planning application to demonstrate use cases around TM1 cubes and views.

Chapter 4, Loading and Maintaining Data, familiarizes the readers with loading and maintaining data (ETL) in TM1. It introduces you to Turbo Integrator module. It extends the example planning application to demonstrate ETL processes using Turbo Integrator module.

Chapter 5, Adding Business Rules, introduces the readers to IBM Cognos TM1 rules editor to write business rules. It presents you the syntax and semantics of major business rules and demonstrates how business logic can be implemented using TM1 Business Rules.

Chapter 6, Automating Data Load Processes using Scripts, guides readers in writing scripts for some of the most useful and common use cases which a TM1 developer may need to write in a real-life planning implementation.

Chapter 7, Customizing Drill Paths, guides readers in implementing drill paths. Recipes are included for drill implementations across cubes and to an ODBC data source.

Chapter 8, Using Rules for Advanced Modelling, extends the idea of writing business rules to implement some of the complex business use cases for instance, virtual cubes, lookup cubes, and spread profile cubes.

Chapter 9, Converting Currencies, introduces the challenges of business being carried out in multiple currencies. Creating a currency dimension and cube in TM1 helps in keeping a single reporting currency and hence consistency while reporting the business results. The chapter has various recipes to demonstrate the concept.

Chapter 10, Modelling for Different Fiscal Requirements, introduces the readers to the time-related aspect and includes recipes to implement a time dimension and continuous time model.

Chapter 11, Optimizing Rules Performance, talks about tuning rules performance. Recipes are included to demonstrate how FEEDERS and SKIPCHECK are written to improve rules performance in a typical TM1 planning application.

Chapter 12, Working with Managed Planning Applications, covers IBM Cognos TM1 Contributor components in a comprehensive manner. It includes the recipes on installation, configuration, usage, and integration of IBM Cognos TM1 Contributor components with IBM Cognos TM1.

Chapter 13, Defining Workflow, introduces IBM Cognos TM1 Workflow component, its installation, configuration, usage, and integration with other IBM Cognos TM1 Components.

Chapter 14, Integration with IBM Cognos BI, deals with integration of the planning application and reporting layer. Recipes are included to demonstrate how IBM Cognos BI components can be interfaced with IBM Cognos TM1 planning application.

What you need for this book

IBM Cognos TM1 (9.5.2) Server, Web Client, and Perspectives for Microsoft Excel

IBM Cognos TM1 (9.5.2) Workflow

IBM Cognos TM1 (9.5.2) Contributor and Contributor Web Client

Microsoft Excel

Microsoft Internet Information Server

Apache Tomcat

Java and .NET framework installed and working

IBM Cognos 8.X BI Server and Framework Manager

Who this book is for

The book is intended for prospective TM1 developers or analysts who want to successfully build and manage a complete planning, budgeting, and forecasting solution with IBM Cognos TM1. No previous knowledge of TM1 is expected.

Existing TM1 users and developers will also benefit from the practical recipes covered in the book.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Each TM1 data server as shown must have `tm1s.cfg` (configuration file) and `tm1s.lic` file in the TM1 data server directory."



A line of code is set as follows:



```
static:value1:value2:value3:value4.
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
Profit = Unit Sales Price - Unit Cost
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "First dimension in column is **Time (Years)** and second dimension in column is **Product (Product Line)**".

 Warnings or important notes appear in a box like this. 

 Tips and tricks appear like this. 

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Getting Started with TM1

IBM Cognos TM1 integrates business planning, performance measurement, and operational data to enable companies to optimize business effectiveness and customer interaction, regardless of geography or structure.

TM1 provides immediate visibility into data, accountability within a collaborative process, and a consistent view of information, allowing managers to quickly stabilize operational fluctuations and take advantage of new opportunities.

TM1 has a distinctive advantage over other analytical tools in the market in the following major points:

- ▶ TM1 is based on multi-dimensional database in the form of cubes with write-back functionality to the source databases
- ▶ The TM1 engine performs complex calculations in memory on the client side, resulting in fast processing and scalability
- ▶ Only non-null data cells are transferred to the client machine which generally forms no more than 20 percent of the overall data set; hence, network bandwidth related issues are nonexistent

The forthcoming chapter will introduce TM1 as a solution along with general performance management concepts to start with.

IBM Cognos Performance Management Solution consists of the following major components:

- ▶ Enterprise Planning
- ▶ Enterprise Business Intelligence

TM1 would be covered under Enterprise Planning and will remain the focal point in this book. Please note that TM1 cubes can serve as a data source for Business Intelligence (BI) applications. The basics of this integration will be dealt with in last chapter of this book. However, BI will not be covered in full detail with respect to authoring reports, charts, ad hoc query generation, and other specific authoring practices adopted in Cognos BI suite. These are best explained on other offerings by Packt Publishing such as *Cognos Cookbook*.

The book will focus on creating and managing planning applications using TM1 offerings. It will also discuss the contributor component and its applications in detail. Lastly, it will touch upon integration points between the planning application and BI infrastructure which may or may not be useful to all readers.

Enterprise Planning

Tools under this category are meant for planning, budgeting, forecasting, modelling, consolidation, and financial reporting. A primary use case is to create managed planning applications that can be used by business users to contribute planning data to a central repository. Business data once submitted can be reviewed by stakeholders and can be approved or rejected. The whole workflow is controlled by applying different levels of security. Data, once finalized, can be viewed in different templates and users can perform a slice and dice as applicable.

An example would be a central budgeting application in which business users are spread across different geographical locations and contribute data according to the subsidiaries they are responsible for. Once submitted, data are reviewed by associated managers and if approved final numbers are written back to the source databases. Top-level management can be provided with reports and templates to view and manipulate data as they want.

Enterprise Business Intelligence

Tools under this category serve to model and analyze business data and discover hidden business trends. It enables users to define key performance indicators and track the critical business metrics against business-specific parameters and criteria. This would typically include reporting, dashboarding, scorecarding, and so on, and would each be covered by different tools under Cognos BI Suite such as Report Studio, Metric Studio, Event Studio, Analysis Studio, and so on.

Major Cognos tools for Enterprise Planning and Business Intelligence are shown below:

Category	Tool	Description
Enterprise Planning	Cognos Planning Suite	Planning and budgeting solution.
Enterprise Planning	Cognos TM1	Planning and budgeting solution.
Enterprise Business Intelligence	Cognos BI	Scorecarding, event management, analysis, and reporting.
Enterprise Business Intelligence	Cognos GO	Extended BI (Office integration, mobile portability, and flash-based dashboarding).
Enterprise Business Intelligence	Cognos NOW	Transactional and runtime BI solution.

In the following chapters we will focus on Cognos TM1 as a planning and budgeting tool, based on OLAP technology as explained in the following section.

Online Analytic Processing (OLAP)

In computing, OLAP is an approach to swiftly answer multi-dimensional analytical queries.

Typically, OLAP uses a multi-dimensional data model to build a relational database (ROLAP) or is directly based on a multi-dimensional cube (MOLAP).

The OLAP tool refers to the category of software tools that provides analysis of data and can include either a ROLAP or MOLAP. In this book we refer to TM1, which performs MOLAP as it is directly based on multi-dimensional cubes. A ROLAP tool would have a multi-dimensional Model/View on top of a relational database, allowing multi-dimensional queries. However, a relational database is still referred to in the backend.

When we say that data is stored in a multi-dimensional database we mean that data is stored in a cube data structure, a single point of data to be measured can be viewed and analyzed from different perspectives at the same time.

This single point of data (mostly numerical in nature) is analogous to a measure (which needs to be measured) and a perspective (descriptive in nature) is analogous to a dimension, which again can be hierarchical in nature.

Imagine an N-coordinate system with an axis referring to dimensions and a data tuple referring to a measure. The dimensions can again be flat or hierarchical in nature.

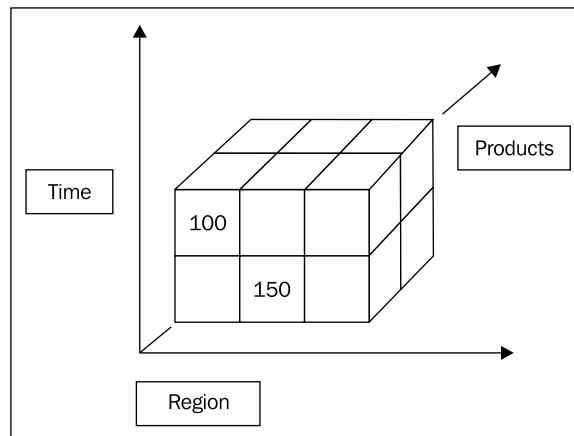
A hierarchical dimension refers to functionally similar data, arranged in different levels of detail.

For example, year, month, and day data refers to similar time-related data, but can be arranged in different levels as Years | Month | Day. This structure enables the user to navigate from a higher level of detail (Year) to a lower level of detail (Month | Day). Navigating from higher level details to lower level details is called drilling down, .Navigating from lower level details to higher level details is called drilling up. Hence, in the preceding example, the hierarchical nature of the time dimension enables the user to drill up (for example, from month to year) and drill down (for example, from year to month).

A similar analogy would be a region dimension, with different levels such as World | Country | Region | City.

As has been just explained, there can be multiple hierarchical dimensions and measures which can together be arranged as cubes. An example of such a cube can be a Sales cube which has Sales Revenue and Sales Target as two measures. Time (Year | Month | Day) and Product (Product Line | Product Type | Product) can be two hierarchical dimensions in such a cube. Please note it is not absolutely necessary for a dimension to be hierarchical in nature. The exact nature may depend on the business use cases that an analyst is trying to achieve.

As a logical conclusion to the explanation above, once such Sales cube is ready, a user can view Sales Revenue and Sales Target from different dimensions (Time or Product) and at a different level of details (drill up, drill down) and all this can be done at the same time, even when user is in offline mode.



The preceding figure depicts the multi-dimensional Sales cube with three axes signifying the three dimensions. A cube cell represents data value or tuples that have a numerical value such as Sales Target.



Please note that there are NULLS in the cube for some combinations. These NULLS will be skipped while processing is done in client side memory; the processing is much faster and more efficient.

A few possible dimensional analysis scenarios based on the above Sales cube setup would be worth showing as simple crosstabs created manually in Excel:

Region = 'Asia Pacific'				
Sales ('Millions)		Product (Product Line)		
		FMCG	Vehicles	Electronics
Time (Year)	1996	100	200	
	1997	340	200	
	1998	1000	340	
	1999	1500	400	

The preceding figure illustrates a dimensional modelling scenario where the measure, for example, Sales volume which is in millions is being viewed from more than one perspective and is represented as dimension (level) in rows, columns, and filters. The first dimension in column is **Time (Years)** and second dimension in the row is **Product (Product Line)**.

The data shown is only for the Asia Pacific region as stated in the filter for Region.

So, we are using three dimensions in the preceding analysis: **Region, Time,** and **Product Line** while measure being analyzed is Sales Revenue.

Year and Product Line are levels of Time and Product dimensions, respectively. Asia Pacific is a member of the topmost level of the Region dimension.

Time = 2010, Region = 'Asia Pacific'				
Sales ('Millions)		Product (Product Line)		
		FMCG	Vehicles	Electronics
Time (Months)	Jan	50	100	150
	Feb	30	200	100
	Mar	40	200	120
	Apr	100	400	100
	May	200	300	250

Similarly, the preceding figure illustrates analysis with three dimensions again: **Time (Months)**, **Product (Product Line)**, and **Region**.

Months, displayed in rows, is second level of **Time** dimension. In this case we drill down on 2010, to view monthly level of details along the **Time** dimension.

Ideally, monthly level of details should add up to give yearly figures as a default behavior; the year, being the consolidated element, is the sum of the individual monthly figures.

Time = 2010, Product Line = 'Vehicles', Region = 'Asia Pacific'				
Sales ('Millions)	Time(Months)	Product (Product Type)		
		2 Wheel	4 Wheel	Trucks
	Jan	50	40	10
	Feb	150	25	25
	Mar	125	15	60
	Apr	100	250	50
	May	150	150	0

The preceding figure extends the same example, but the **Product** dimension is further drilled down to the **Product Type** level to find details for the **Vehicles Product Line** as shown. Please note that detailed figures for each month add up to the **Vehicles** figures for the respective months.

Time = 2010, Product Line = 'Vehicles', Region = 'Asia Pacific'				
Profit Booked ('000 USD)	Time(Months)	Product (Product Type)		
		2 Wheel	4 Wheel	Trucks
	Jan	100	40	10
	Feb	150	25	25
	Mar	125	120	45
	Apr	20	140	50
	May	150	150	100

As shown in the above figure, measures can also be changed (Profit Booked instead of Sales Volume) according to the business use cases. The case shown in the above figure is similar, where figures now represent Profit Booked in '000 USD instead of Sales Volume in Millions.

OLAP data structures terminology

This topic intends to introduce the data structure terminology generally used when it comes to OLAP. These terms are quite liberally used throughout this book and in the data warehousing/BI world in general.

Please refer to the following table for the terminology:

Terminology	Description	Example
Dimensions	Perspectives through which users want to slice and dice data.	Product Time
Hierarchies	Group of functionally similar descriptive data arranged in levels, One or more hierarchies form a dimension.	Product Line Product Type Year Month Day
Levels	Individual building blocks of a hierarchy.	Product Line Day
Measures	Values which can be measured for various dimensional members.	Sales Revenue Sales Target
Cubes	Collection of dimensions and measures. Functionally similar data which can be analyzed in an OLAP tool.	Sales Cube
Members	Individual data values.	FMCG, 4 Wheelers 2010, JAN
Attributes	Provide additional details for member values.	Color Number of Hours

OLAP (Online Analytical Processing) versus OLTP (Online Transactional)

As we move further, it's worth understanding the difference between an OLTP -based and an OLAP-based system. This is just for the sake of knowing, as TM1 is based on OLAP technology which is the primary focus of this book.



In a nut shell, all the day-to-day processing of databases support daily transactional processing, do not focus on historical data analysis, and are based on OLTP technology.

OLAP-based systems are more historical databases, intended to support MIS or DSS with fast read operations and large volumes of aggregated data. **Denormalization** is the key to such databases as opposed to OLTP systems which follow a **Normalization** approach.

Denormalization refers to the process of attempting to optimize the read performance of a database by adding redundant data or by grouping data. Hence, different but related pieces of information are still duplicated across tables, avoiding joins while read operations are performed.

Normalization would try to avoid this duplication by storing different pieces of related information in different tables, making data manipulation statements (inserts/updates) fast, but read-only queries now have to use more joins and the performance is slower.

Refer to the following table to compare the OLAP and OLTP approaches:

	OLAP	OLTP
Definition	Multi-dimensional analysis of data arranged in cube format and modeled in a dimensional structure.	Analysis of data arranged in a relational database. Dimensional structuring of data is not necessary.
Schema	Star Schema, Snow Flake Schema.	Flat Schema.
Normalization	Renormalized.	Normalized.
Limitations	Inserts and updates are slow.	Read operations are slow.
Advantages	Fast read operations.	Fast updates and inserts.
Portability	Cubes can be created and accessed in an offline mode.	OLTP processing typically requires users to be online if they need to do any data analysis.
Use	Typically used where quick and efficient analysis is the primary use case.	Typically used where insertions and updates are to be performed frequently as part of day-to-day operations.
Data volumes	Typically involves large data volumes and historical data.	Typically data volumes are less and historical data is generally not maintained.
Pre-aggregated values	Computed at cube creation time itself for fast analysis, for example, total sales revenue for each quarter.	Computed at runtime, saving a lot of disk space but slower at the time of retrieval.

Introduction to TM1

TM1 is an OLAP tool with one or more powerful TM1 data servers running remotely or on a single server. TM1 data servers provide access to shared data in an organization. These data servers are controlled by one or more admin servers. Users can work on shared data and objects across the organization through intuitive client interfaces, which interact and send requests to admin servers. Admin servers then pass on the request to one of the TM1 data servers. The data server then provides access to the shared objects to the clients.

Client interfaces provided with TM1 are:

- ▶ Excel add-on (TM1 Perspectives)
- ▶ Web browser (TM1 Web)
- ▶ TM1 Architect

The interface users want to use depends on the role they play in the organization as well as the comfort level they have with the particular interface.

In addition, clients have to install TM1 Perspectives for Excel as a separate add-in. While installing TM1, though this gets installed automatically, clients who just need to access the data need not have a full blown TM1 setup, but only have the Excel plugin if they want to use TM1 Perspectives for Microsoft Excel.

TM1 Web does not require any installation on the client side and can be accessed through the Web.



In subsequent chapters we will be using TM1 Architect as the interface to create and view sample applications.

TM1 has a built-in ETL (Extraction Transform and Load) tool to perform data transfers from source databases to target multi-dimensional cubes, in the required format. In TM1 terminology the ETL module is called Turbo Integrator and ETL processes are called Turbo Integrator (TI) processes.

TI processes can also be written to write-back data in source databases, as and when the user modifies business data and decides to save that back to the backend databases.

In a nut shell, business-specific multi-dimensional models are created using one of the client interfaces stated above. Inbuilt TI processes enable users to pull data from the backend source databases and populate cubes on top of the models. TI processes can also be custom written and modified for specific ETL requirements. Business-specific reporting templates are then created, again using one of the client interfaces on top of the relevant cubes. Users are given access to the cubes and templates to view, analyze, and modify data. Data can also be saved to cubes to source databases using inbuilt TI processes.

Examine TM1 user roles

Users can be given access to client interfaces and tools in TM1 depending on the role they play in the organization. Some might be interested in defining business logic, while others may assume a role to model multi-dimensional cubes by defining dimensions and facts.

Admin roles also need to be clearly defined to perform housekeeping activities such as defining users, security policies, backup and recovery.

Contributors are end users who want to fill in business-specific data in already designed templates and play a role in final analysis and consolidation.

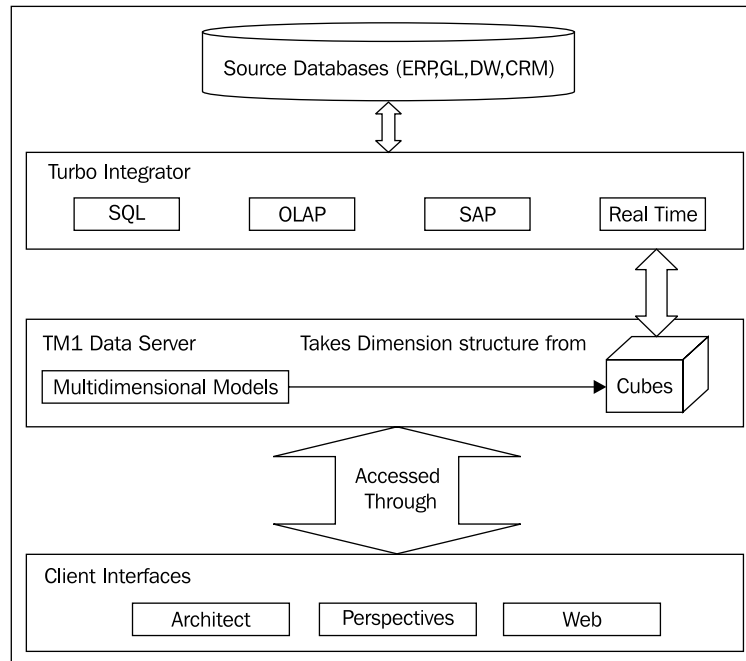
Please find the summary of the users and generic roles which are generally followed. The exact nature of user distribution and their respective responsibilities can differ from organization to organization.

User	Roles and responsibilities
Modeler/Analyst/Application Developer	Designs and builds multi-dimensional models and cubes, applies business logic, maps data flows, and creates custom templates and reports.
Administrator	Manages technical deployment of the application as a whole. Reviews hardware and software requirements. Keeps track of all the housekeeping activities such as security, replication, backup and recovery.
Contributor	Input data in templates. Views reports and answers business-specific queries.

Subsequent chapters will introduce different set of activities carried out by each of these in detail. Although the preceding table should be taken as a general guideline, there is no clear line of demarcation between duties performed by these users as roles often overlap among users.

TM1 architecture

TM1 is based on client server architecture and different components that interact with each other in a seamless manner to achieve the desired functionality. The following figure depicts the development process in a broad manner:



As depicted in the preceding figure, multi-dimensional cubes reside on TM1 data servers. These cubes in turn take their dimensional structure from multi-dimensional models, created by the modeler. Data is fed to these cubes from one of the backend data sources through inbuilt turbo integrator processes. Please note that these ETL processes can be modified as per the user requirements.

Turbo integrator processes can be written to fetch data from a variety of data sources including SAP, OLAP, relational (using SQL queries), and real time systems.

Import from text and Excel files is also possible.

Therefore, backend data sources can be of any type and turbo integrator handles all the database-specific complexities. Users don't have to worry about disparate multiple data sources.

Data can be modified through client interfaces and transferred to cubes. Therefore, changed data in cubes can then be saved back to backend databases, usually through inbuilt or customized TI processes as per the business-specific requirements.

The TM1 cubes are nothing but business data in a multi-dimensional format. Though dimensions and facts in the cubes are decided by the multi-dimensional model created by the analyst/modeler, the data in the cubes are usually viewed and analyzed using one of the client interfaces given by TM1.

TM1 cubes, along with its dimensions and other components, are referred to as TM1 objects and are hosted on one of the TM1 data servers. Different TM1 data servers can host different TM1 objects depending on the semantics of the organization. For example, there may be a separate data server to host financial-related cubes and processes, with only the finance department, CFO, and CEO having access to it. Similarly, a sales-related data server hosts sales plan, sales revenue, and expenses-related cubes shares only with the sales department and higher management.

TM1 Admin Server is a process that keeps track of all TM1 Data Servers running on a network and registered to it. An Admin Server runs on a computer known as an Admin Host.

The TM1 Data Server needs to have the associated Admin Host specified in the configurations file which is specific to each TM1 Data Server. In future sections we will discuss this in detail.

When a TM1 Data Server starts, the server registers itself with the Admin Server process that is running on the specified Admin Host. TM1 clients reference the Admin Server process to determine which TM1 Data Server are available on the network.

A user, depending on his role, can issue a request to access shared TM1 objects (cubes, dimensions, and subsets) through one of the client interfaces. These requests are routed to TM1 Data Server by one of the TM1 Admin Server processes.

The users, through client interfaces (Excel, web, or Architect), are given access to Server Explorer which is the interface provided by the TM1 Admin Server process. A Server Explorer lists all the TM1 Data Servers registered with the TM1 Admin Server process on the network, depending on the access rights a user may have.

TM1 Perspectives and TM1 Architect can connect to a local TM1 server, which acts as a repository for private TM1 data. With proper authority, we can copy data from a remote server to the local server by replicating that data, and then synchronize the updates back to the remote server.

The TM1 Data Server then exposes the shared TM1 objects it hosts to the user, depending on the security policies set by admin. This is how a user can create and access the TM1 objects using one of the client interfaces, again depending on the security policies already defined.

In the backend, turbo integrator processes come into play. ETL jobs related to these processes are then executed and can be customized.

Turbo integrator processes are executed from within Architect or Perspectives and hide from the users the complexity involved in querying multiple databases of different types.

Please refer to the following figure to get an idea of a typical deployment scenario of a TM1-based application, different components, and how they interact with each other.

All the TM1 Data Servers can reside on a single server or on multiple servers. Each TM1 Data Server as shown must have a `tm1s.cfg` (configuration file) and `tm1s.lic` file in the TM1 Data Server directory.

As we hinted previously, the `tm1s.cfg` file is the configuration file which is specific for every TM1 Data Server. This file is continuously polled at the interval of 60 seconds by the TM1 Data Server to detect any changes to the dynamic parameters. If a dynamic parameter is changed while the TM1 Data Server is running, the changed value is applied immediately and a restart is not needed.

Most parameters in the `tm1s.cfg` file are static. These parameter values are read from the `tm1s.cfg` file only when the TM1 Data Server starts. If we want to change a static parameter value, we must shut down the TM1 Data Server, edit the value in the `tm1s.cfg` file, and then restart the TM1 Data Server.

If a parameter value contains spaces, you must enclose the parameter values within double quotes.

Now, in the following section we will discuss another very important concept which is specific to each TM1 Data Server installed. We will return to the `tm1s.cfg` file after that.

Data directory

The data directory, which is again specific to a TM1 Data Server contains the cubes, dimensions, and system information that are loaded into memory when a TM1 Data Server is started. When we access a TM1 Data Server from any of the TM1 clients, TM1 reads not null data from that server's data directory and loads that in the memory.

Any changes we make to cube data values are immediately stored in the memory and in the transaction log (`Tm1s.log`). Calculations and consolidations are done on demand in the memory. Hence, all the TM1 data files, that is cubes, dimensions, and so on are maintained in the memory and all the manipulations on that are performed in the memory, making the processing really fast. This in-memory processing of data along with the feature that only not null data is loaded gives the tool a distinctive advantage in terms of processing speed.

TM1 saves the data back to the data directory on disk when any of the following occur:

- ▶ The TM1 Data Server is shut down.
- ▶ A user, depending on the privileges, right-clicks a server icon in **Server Explorer** and chooses **Save Data** from the pop-up menu. This directs TM1 to save the changes to the selected server.
- ▶ An administrator chooses **File | Save Data All** in **Server Explorer**. This directs the TM1 admin process to save the changes to all the connected TM1 Data Server, if he has the proper authority.
- ▶ A user saves the batch updates.

Therefore, until the data is saved either manually or through TI processes all operations are done in the memory, causing fast processing.

We need to choose the path for the data directory when TM1 is installed. Please note that we need to specify different locations of the data directory for local and remote servers.

- ▶ For a local server, we need to specify the location of the data directory by naming this directory in the `DataBaseDirectory` parameter of the `Tm1p.ini` file. We can change the `.ini` file by using the **TM1 Options** menu in **Server Explorer**. By default, the path of the data directory for the local server is `<install_dir>\custom\tm1data\pdata`.
- ▶ For a remote server, we need to specify the location of the data directory by using the `DataBaseDirectory` parameter in the `Tm1s.cfg`.



When TM1 is installed the planning sample TM1 Data Server is installed by default with the data directory as `<install_dir>\custom\tm1data\PlanSamp`. The password for an admin user is `apple` and the default user ID and password is `admin/apple`. Readers are encouraged to install TM1 and play around with the sample data server.

Please refer to the following table which describes each of the mandatory parameter values in the `Tm1s.cfg` file, described in the previous section. Both parameter values are static in nature.

Sr No	Parameter name	Description
1	AdminHost	Specifies the computer name or IP address of the Admin Host on which an Admin Server is running. We can specify multiple Admin Hosts by separating each host name with a semicolon on a Windows TM1 Server, or a colon on a UNIX TM1 Server. The string specifying the Admin Host(s) is limited to 1020 characters or bytes.
2	DataBaseDirectory	Specifies the data directory from which the server loads cubes, dimensions, and other objects. You can list multiple data directories by separating them with semicolons.

Now we will give a quick brief of another important file in TM1 called the `Tm1p.ini` file. The `Tm1p.ini` file specifies the environment information for a TM1 client (TM1 Perspectives, TM1 Architect) and is specific to each user logging in to the TM1 Data Server through one of the clients stated above. `Tm1p.ini` has the `AdminHost` parameter which has the TM1 Admin Host referenced by the client.

When we install TM1, the installation location for the system default version of the `Tm1p.ini` file is `%ALLUSERSPROFILES%\Application Data\Applix\TM1\Tm1p.ini`.

In most cases, the full path to the `Tm1p.ini` file is:

```
C:\Documents and Settings\All Users\Application Data\Applix\TM1\Tm1p.ini
```

The system default version of `Tm1p.ini` allows multiple users to use TM1 on a given computer. `Tm1p.ini` must be present the first time a user starts TM1 on the computer, the parameters in the system's default version govern the behavior of the initial startup of the TM1 client for each user.

After a user starts TM1 on the computer, a user-specific copy of `Tm1p.ini` is created in `%APPDATA%\Applix\TM1\Tm1p.ini`.

In most cases, the full path to the `Tm1p.ini` file is:

```
C:\Documents and Settings\\Application Data\Applix\TM1\Tm1p.ini
```

The user-specific copy of `Tm1p.ini` accepts all parameter settings and changes for the user and governs the behavior of the TM1 client for all subsequent user sessions of the TM1 client.

`DataBaseDirectory` is one of the important parameters specified in the file which is used to specify the full path to the local server data directory. We can specify multiple data directories by separating the directory names with semicolons. A detailed description of these setting can be referenced in TM1 Administer and Deploy guide.

Please take a note of the important parameters in the `Tm1p.ini` file.

Sr No	Parameter name	Description
1	<code>AdminHost</code>	Displays the Admin Host name on which an Admin Server is running. On TM1 Options, use Login Parameters Admin Host.
2	<code>DataBaseDirectory</code>	Uses the full path to the local server data directory. We can specify multiple data directories by separating the directory names with semicolons.

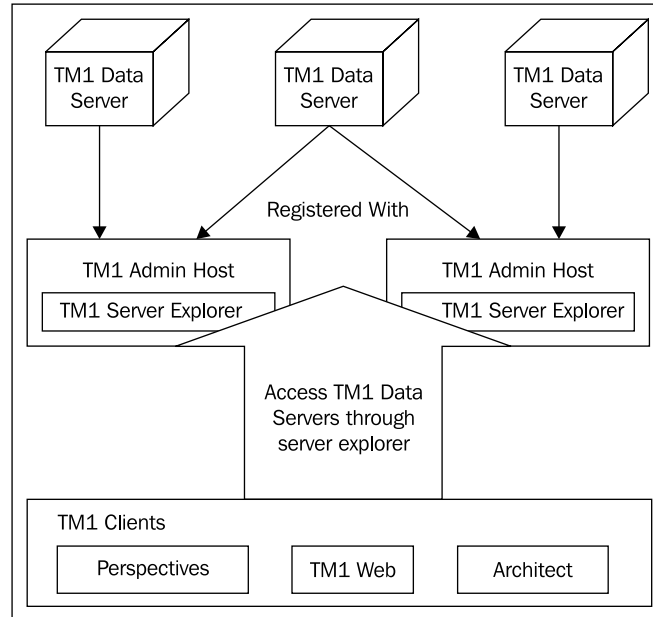
The **TM1 Options** dialog box also stores many of the settings available in `Tm1p.ini`. The **TM1 Options** menu is available in **Server Explorer**. We can change these settings using either the TM1 Options dialog box or by editing the `Tm1p.ini` file.

In a nut shell we can specify the location of the Admin Host differently for clients and remote servers.

- ▶ Specify the Admin Host referenced by clients in the `AdminHost` parameter of the `Tm1p.ini` file. You can change the `Tm1p.ini` file by using the **TM1 Options** menu in **Server Explorer**.
- ▶ Specify the Admin Host and the remote servers they are registered with by using the `AdminHost` parameter in the `Tm1s.cfg` file.

TM1 Perspectives and TM1 Architect can connect to a local TM1 server, which acts as a repository for private TM1 data. With proper authority, we can copy data from a remote server to the local server by replicating that data, and then synchronize the updates back to the remote server.

Now we will look at the whole workflow summary in the following diagram:



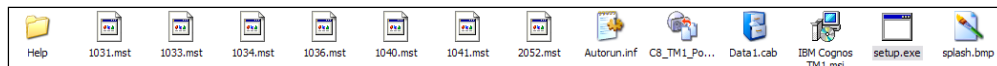
Installing TM1 server and clients on a machine

This section will demonstrate the installation of a typical TM1 single machine setup which we will use for subsequent examples to be followed in the book.

Please note that it is assumed that users have the necessary licenses in place to use the TM1V9.5.2 server and its components, which forms the infrastructure backbone of all the following practical applications.

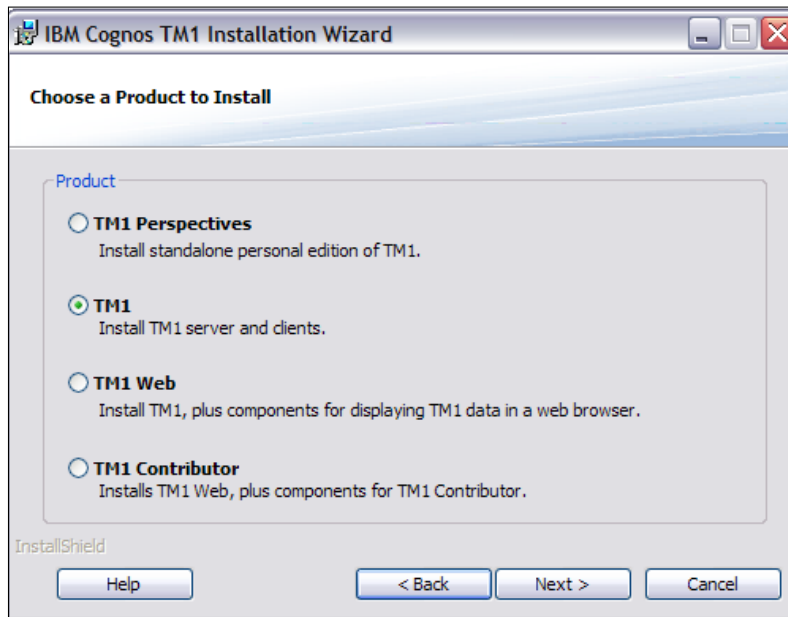
Let's begin with the installation:

1. Copy TM1 setup on the local drive and click on `setup.exe`.

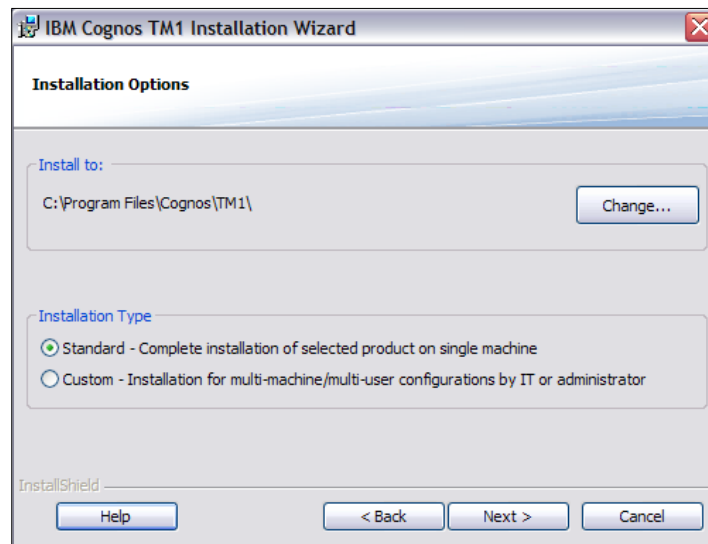


2. Click on **Next** and accept the default options in subsequent prompts and wizards. Only the important screens will be captured in subsequent steps, for the rest just accept the default values and click on the **OK** or **Next** button, whichever is applicable.

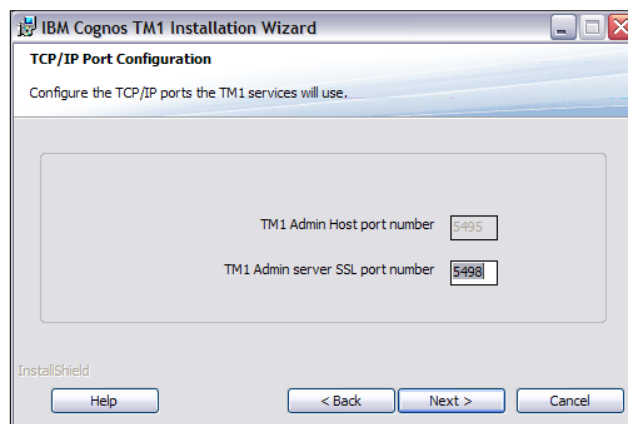
3. In the following screenshot you can select:
 - ❑ **TM1 Perspectives** to install the server and perspectives for Microsoft Excel for clients with an Excel interface.
 - ❑ **TM1** to install the TM1 Server, TM1 Architect, and TM1 Perspectives.
 - ❑ **TM1 Web** to install the TM1 Server, TM1 Architect, TM1 Web, and TM1 Perspectives.
 - ❑ **TM1 Contributor** to install TM1 Server, TM1 Architect, TM1 Web, TM1 Perspectives, and TM1 Contributor.
4. In the following screenshot **TM1** has been selected, but ideally readers should select **TM1 Contributor** if they are interested in TM1 Web and Contributor as well, which will be covered in detail later. The installation steps shown next will be the same for all of the preceding options.



5. Review and change the required installation directory, as shown in the first pane in the following screenshot. Select **Standard** as the installation type, it will install all the components of TM1 and TM1 Contributor on the single machine. The second option is useful if multi-server installation is required where components are distributed across multiple machines.



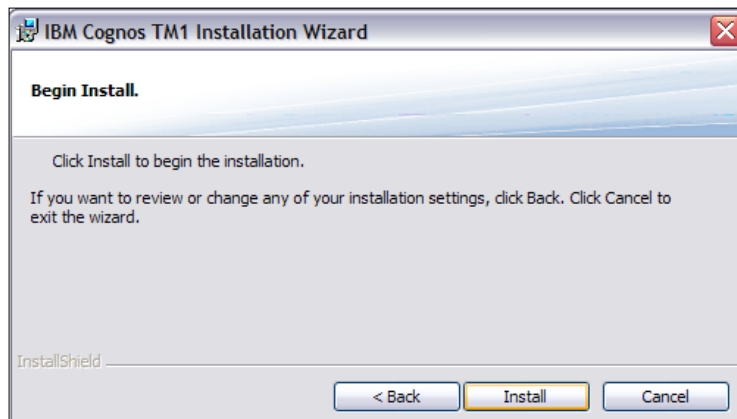
6. All the TM1 components communicate with TM1 Admin Server using SSL. The admin server also supports older TM1 clients which cannot use SSL by listening on two ports, one secured and the other unsecured. For TM1 clients that can use SSL, connect to the TM1 Admin Server using a secured port (otherwise they connect using unsecured port). If the secured port number is already in use in Admin Host, a different port can be specified for the secured port. In general this is not required and can be used as it is.
7. Please note down the port number that the TM1 Server will use during the course of action, as shown in the following screenshot:



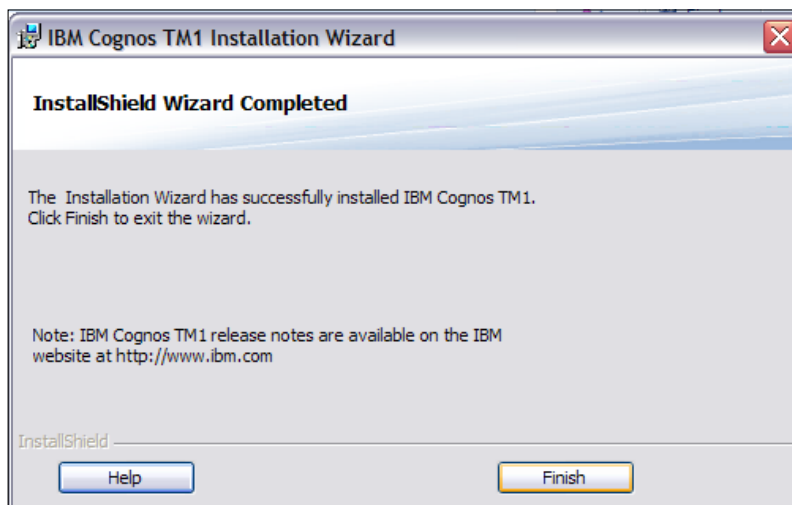
8. Type a valid Administrator username and password of the machine on which the TM1 components are being installed. Please ensure that the correct user credentials are given in the boxes, otherwise the installation will not proceed.



9. Click on **Install** button to start the installation.



10. Click on **Finish** to complete the installation and restart the machine.



The preceding steps will install the TM1 Server and components on the same server. Components installed depend on which option is selected at the beginning. As explained earlier, **TM1 Contributor** will install TM1 Server, its clients (including TM1 Web), and TM1 Contributor on the same server.

Also the **Planning Sample** TM1 Data Server is installed by default.

Please refer to the two new services, under Windows services explorer, which will become visible after the successful installation, as shown in the following screenshot.

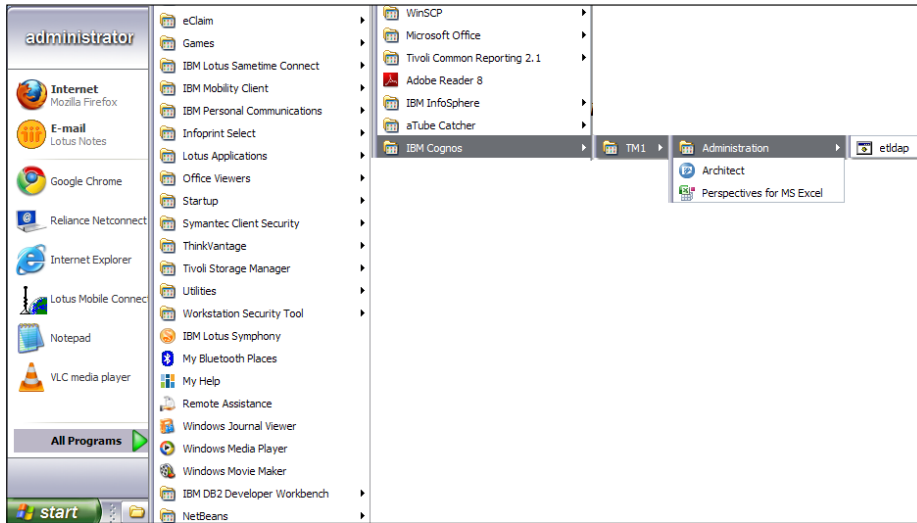
The first service is **TM1 Admin** Server and the second is the default **Planning Sample** TM1 Server, which is installed by default but is not necessarily required.

TM1 Admin Server	Service tha...	Started	Automatic	.\Administ...
TM1 Server - Planning Sample	Services re...	Started	Automatic	.\Administ...

Please note that old models, cubes, and TM1 objects can now be ported on the fresh installation and used as before.

All subsequent recipes will assume this local single server installation scenario on a Windows box.

After successful installation, **Architect** and **Perspectives for MS Excel** will be shown in the **Start Menu**, as shown in the following screenshot:

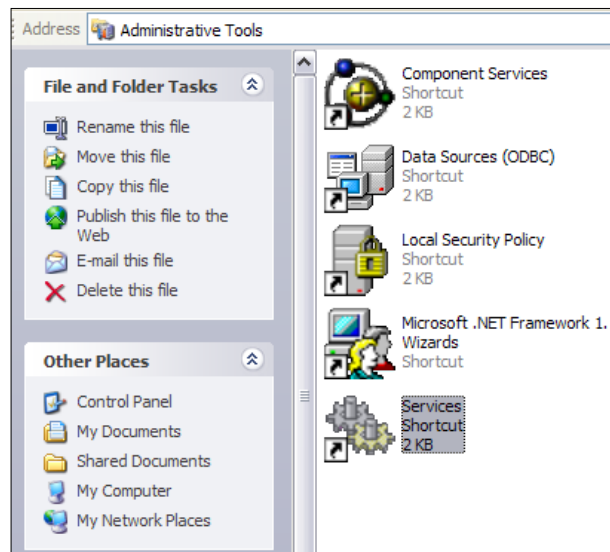


At the end of this section we have successfully installed TM1 V 9.5.2 on a Windows box as a single server installation. TM1 components are installed on the single server and we will be able to follow subsequent practical scenarios as required.

Creating a new TM1 Data Server

We will now create a new TM1 Data Server called DemoData and set up the connection.

1. From the **Start** menu, click on **Control Panel | Administrative Tools | Services** as shown in the following screenshot. Alternatively, press *Windows + R* and then type **Services.msc** to directly open Windows services.

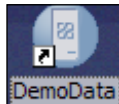


2. Ensure that the **TM1 Admin Server** service is started.

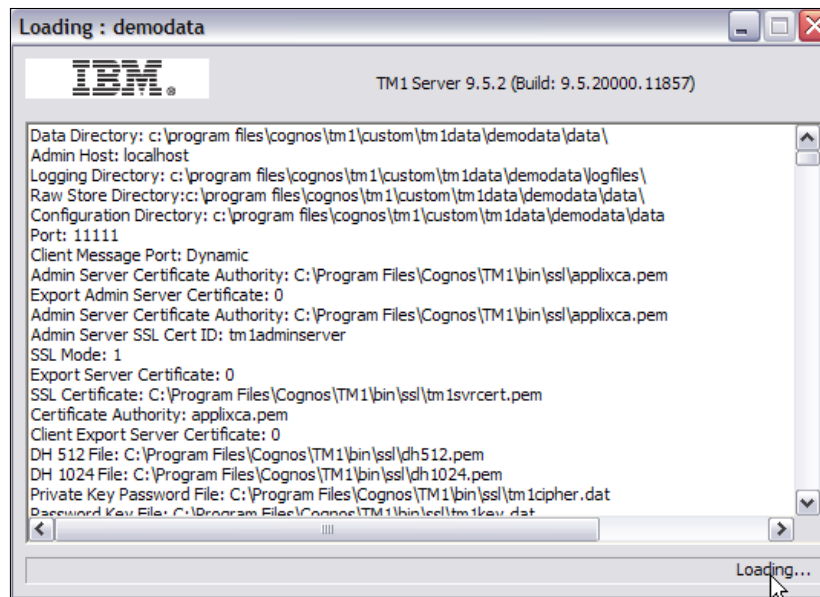
Services (Local)					
	Name	Description	Status	Startup Type	Log On As
TM1 Admin Server					
Stop the service Restart the service	Shell Hardware Detection	Provides n...	Started	Automatic	Local System
	Smart Card	Manages a...	Started	Automatic	Local Service
	SSDP Discovery Service	Enables dis...	Disabled	Automatic	Local Service
	Symantec AntiVirus	Provides re...	Started	Automatic	Local System
Description: Service that manages the TM1 Servers.	Symantec AntiVirus Definition Watcher	Monitors a...	Started	Automatic	Local System
	Symantec Event Manager	Event prop...	Started	Automatic	Local System
	Symantec Network Drivers Service	Symantec ...	Started	Automatic	Local System
	Symantec Network Proxy	Symantec ...	Started	Automatic	Local System
	Symantec SecurePort	Symantec ...	Started	Automatic	Local System
	Symantec Settings Manager	Settings st...	Started	Automatic	Local System
	Symantec SPBBCSvc	Symantec ...	Started	Automatic	Local System
	System Event Notification	Tracks syst...	Started	Automatic	Local System
	System Restore Service	Performs s...	Started	Automatic	Local System
	Task Scheduler	Enables a ...	Started	Automatic	Local System
	TCP/IP NetBIOS Helper	Enables su...	Started	Automatic	Local System
	Telephony	Provides T...	Started	Manual	Local System
	Telnet	Enables a r...	Disabled	Manual	Local System
	Terminal Services	Allows mult...	Started	Manual	Local System
	Themes	Provides u...	Started	Automatic	Local System
	ThinkPad HDD APS Logging Service		Started	Automatic	Local System
	ThinkPad PM Service		Started	Automatic	Local System
	Tivoli Integrated Portal - TIPProfile_Port_16310	Controls th...	Started	Automatic	Local System
	Tivoli Integrated Portal - TIPProfile_Port_16330	Controls th...	Disabled	Automatic	Local System
	TM1 Admin Server	Service tha...	Started	Automatic	.\administ...

3. Browse to C:\Program Files\Cognos\TM1\Custom\TM1Data\PlanSamp and copy the tm1s.cfg file.
4. Browse to C:\Program Files\Cognos\TM1\Custom\TM1Data and create a subfolder called DemoData.

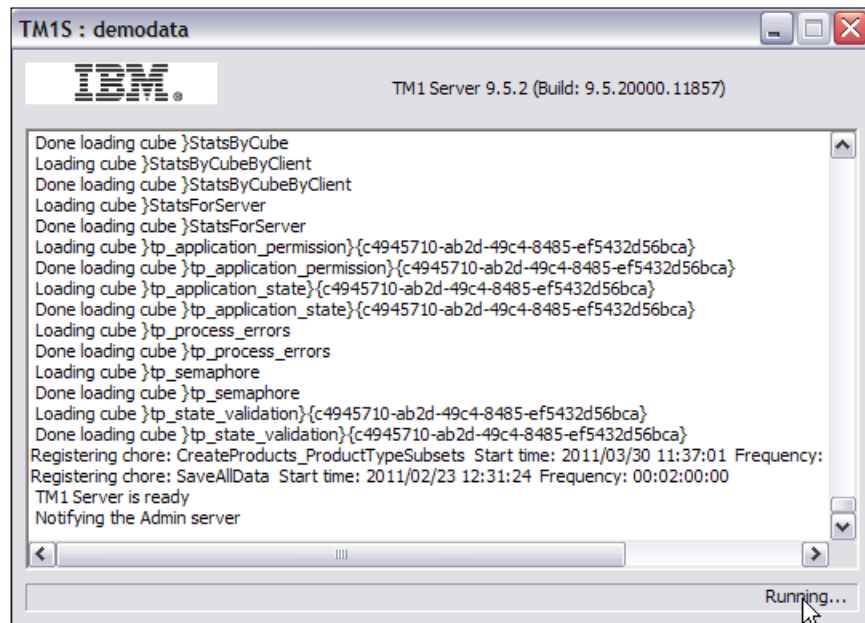
5. Create subfolders called Logfiles and Data in C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData.
6. Paste the tm1s.cfg file copied in step 3 and paste in to C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData\Data.
7. Open tm1s.cfg in Notepad.
8. Change ServerName to DemoData.
9. Change DataBaseDirectory to C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData\Data.
10. Change the PortNumber to 11111.
11. Change AdminHost to localhost.
12. Add a return after DataBaseDirectory = and then type the following
LoggingDirectory=C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData\Logfiles.
13. Save and close the file.
14. Browse to C:\Program Files\Cognos\TM1\bin and create a shortcut for tm1s.exe on the desktop.
15. Rename the shortcut to **DemoData**.
16. Right-click on the folder and then click on **Properties**.
17. In the **Target** box at the end of the given expression type -z "C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData\Data". The complete expression should now look like "C:\Program Files\Cognos\TM1\bin\tm1s.exe" -z "C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData\Data".
18. Click on **OK**.
19. Double-click the **DemoData** icon to start the data server just created.



20. The following window pops up to display the status of the server, which has started and the taskbar status will change from **Loading** to **Running**.



21. The server is now ready for usage.

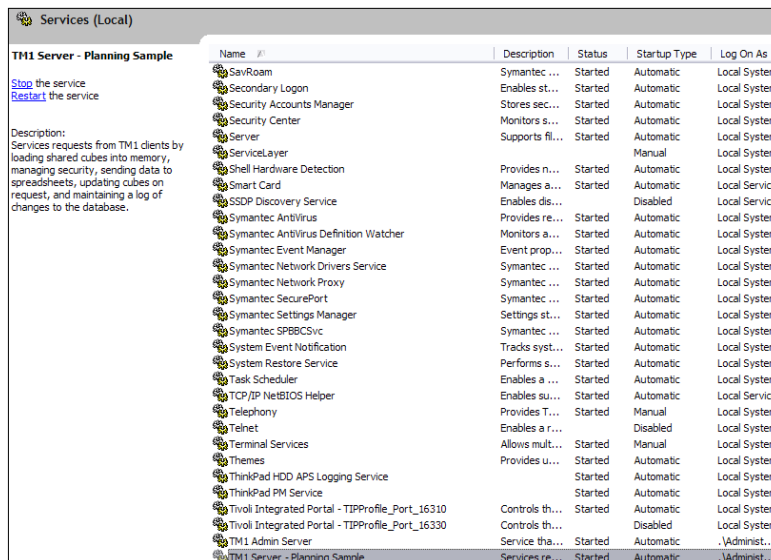


22. Close the preceding window to shutdown the server. Don't save anything as we haven't changed any configuration parameters for the server.
23. No two TM1 Data Servers running on the same machine can have same port number and data server name.
24. In our demo scenario we have just created a TM1 Data Server called **DemoData** on the machine. Another data server called **Planning Sample** is already created and shipped with the installation.
25. These data servers can be made visible as a Windows service, as in the case of **Planning Sample**, or they can be accessed as a desktop shortcut as in the case of **DemoData**.
26. Now these data servers can be accessed through Admin Server and any of the clients we installed in the preceding steps.

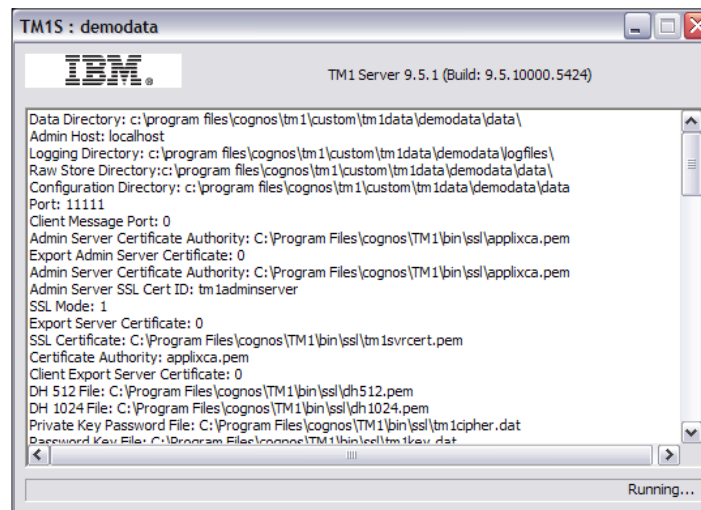
Connecting to an existing TM1 Server

We will now set up a connection to the existing TM1 Server called **Planning Sample**.

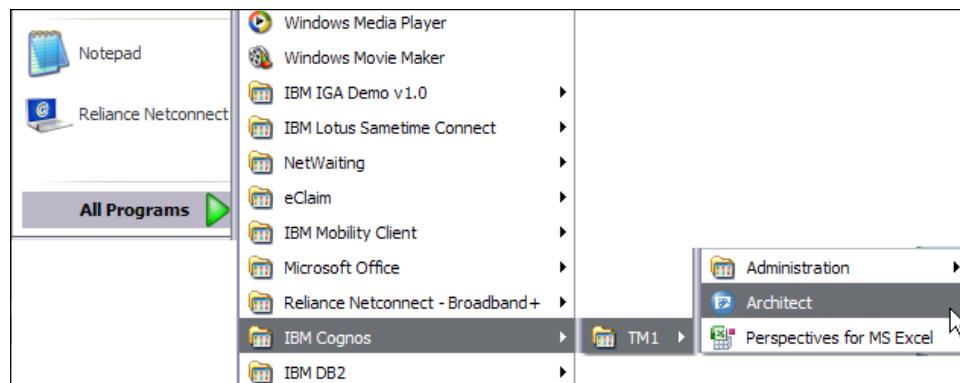
1. As explained in previous recipe, please open Windows services by pressing *Windows* + *R* and then typing **Services.msc**.
2. Ensure that the **TM1 Admin Server** service is started.
3. Ensure that **TM1 Server – Planning Sample** is started, as shown in the following screenshot:



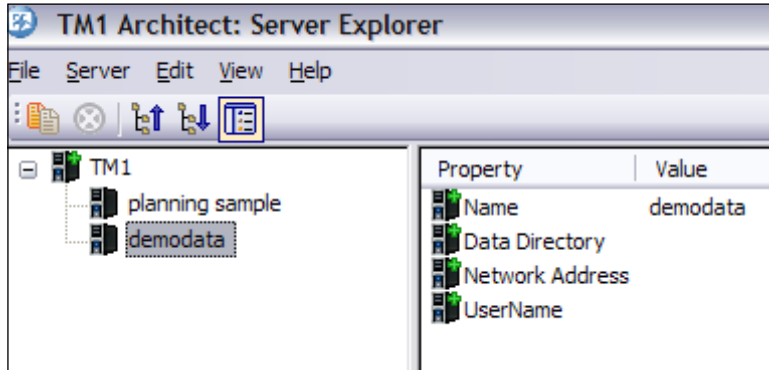
4. Double-click on the **DemoData** icon on the desktop to start the data server we created in the preceding steps.



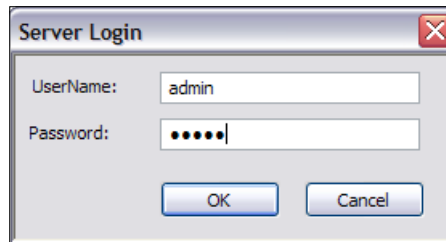
5. Open the **Architect** tool from the **Start** menu as shown in the following screenshot:



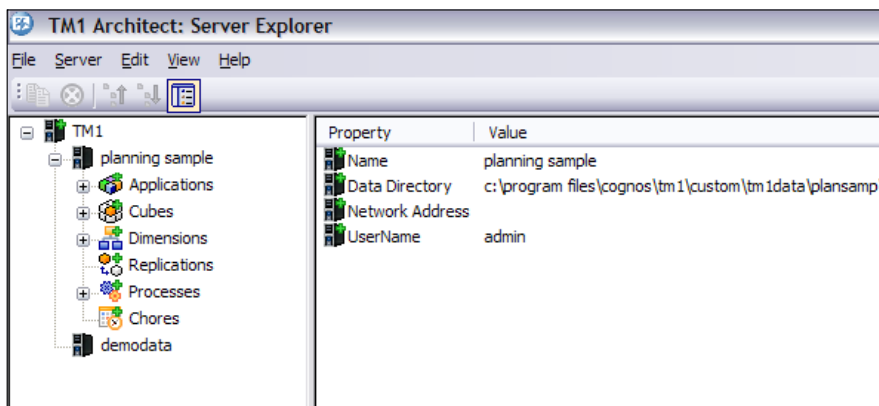
6. In the **Architect** our two data servers are visible.



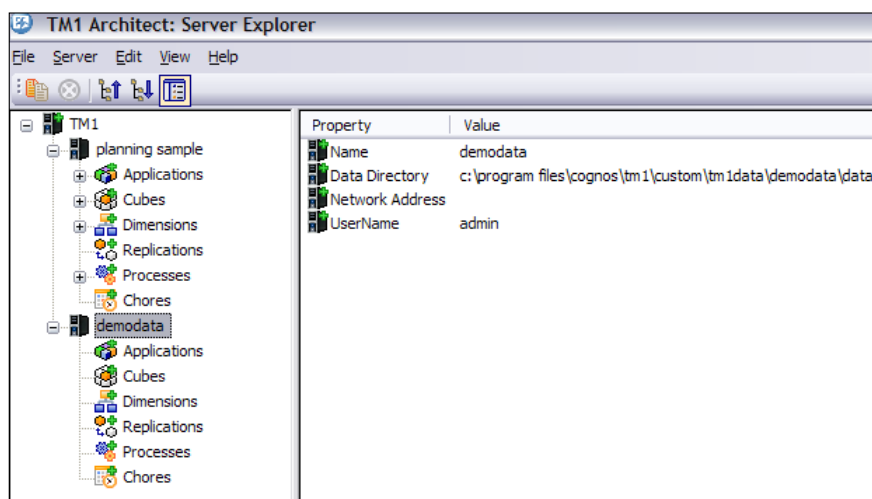
7. Double-click on **planning sample** and provide the login credentials as admin/apple. Please note that TM1 is not case sensitive, while typing the user credentials lower/uppercase is irrelevant.



8. Various sample applications, cubes, and dimensions created on **planning sample** are visible:



- Similarly, log in to **DemoData** using `admin` as the user and no password. It doesn't have any objects created (as we have created the TM1 Data Server from scratch) yet the different nodes are visible.



- The servers are ready for business now and can be accessed through one of the client interfaces as per user preferences. In subsequent modules, we will always access the server explorer using the **Architect**.
- TM1 Admin Server** keeps track of all the TM1 Data Servers running on a network. When a TM1 Data Server starts, it registers itself with the Admin Server that runs on a specified Admin Host. Single or multiple Admin Hosts need to be specified as comma separated values in the Admin Host field for every TM1 Data Server in the `tm1s.cfg` file for the Data Server.
- In our case Admin Server (localhost) and TM1 Data Servers/Application Servers (Planning Sample) are installed on the same server along with the clients. We have created one more TM1 Data Server as part of the demo scenario above (DemoData).

Setting up TM1 Data Server to run as a Windows service

- Open a DOS box/command prompt: **Start | Run | cmd** and navigate to the TM1 bin directory using the `cd` command and go to the TM1 bin directory.

2. Please note that the default installation path is `C:\Program Files\Cognos\TM1\bin`. Type the following command:

```
tm1sd.exe -install -n"NameOfTM1Server" -z"PathtoTm1s.cfgFile"
```

3. In the preceding command `NameOfTM1Server` (after the `-n`) is the name we want to give to this TM1 Server service. If the name we are going to use contains any spaces, this value must be specified in double quotes. `PathtoTm1s.cfgFile` (after the `-z`) is the path to the `tm1s.cfg` file for the TM1 Server we are setting up. For example:

```
tm1sd.exe -install -n"DemoData" -z"C:\Program Files\Cognos\TM1\Custom\TM1Data\DemoData"
```

4. If the preceding command is successful, it will return the following message: **TM1 Server / tm1 production server installed**. We can then go into the services listing, and see the new TM1 Server service you set up. Following the preceding example, the service would be listed as: **TM1 Server / DemoData**. We can then configure the service to run automatically and to run under a specific domain user.



If we are setting up a completely new TM1 Server for which we don't already have a `tm1s.cfg` file, you can copy the `tm1s.cfg` from the **Planning Sample** TM1 Server and modify it accordingly. As a minimum, we will need to modify the `ServerName`, `DataBaseDirectory`, and `PortNumber` parameter values. If the service does not start (error message stating the service cannot be started), starting the same TM1 Logical Server as a desktop application will usually provide more detailed information on the error.

Removal of the TM1 service

1. To remove a TM1 Server service (again, following the preceding example), follow steps 1 and 2, then type the following command:

```
tm1sd.exe -remove -n"NameOfTM1Server"
```

2. Where the `NameOfTM1Server` (after the `-n`) is the name of the TM1 Server service we wish to remove. For example:

```
tm1sd.exe -remove -n"DemoData"
```