

**IBM Cognos TM1:  
Design and Develop Models (v9.5)**  
Instructor Guide Volume 1  
**CourseCode: P6502**

*IBM® Cognos® TM1: Design and Develop  
Models (v9.5)*

P6502

ERC: 2.0

Published March 2010

Licensed Materials – Property of IBM

© Copyright IBM Corp. 2007, 2010

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM corp.

IBM, the IBM logo, ibm.com and Cognos are trademarks of international Business Machines Corp., registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

# Contents

<b>PREFACE.....</b>	<b>IP-1</b>
CONTENTS.....	IP-3
COURSE OVERVIEW .....	IP-15
IMPORTANT COURSE CHANGES .....	IP-17
COURSE OUTLINE .....	IP-21
INSTRUCTIONAL MATERIALS .....	IP-29
GENERAL SETUP AND INSTRUCTOR PREPARATION.....	IP-34
DOCUMENT CONVENTIONS .....	IP-36
WORKSHOPS .....	IP-37
POWERPOINT TIPS .....	IP-38
VMWARE KEYBOARD SHORTCUTS.....	IP-39
GET THE CLASS STARTED .....	IP-41
POST-CLASS AGENDA .....	IP-42
SUBMIT FEEDBACK AND LOCATE ADDITIONAL PRODUCT INFORMATION ...	IP-43
IBM PRODUCT HELP .....	IP-44
<b>SETUP INSTRUCTIONS .....</b>	<b>SI-1</b>
INSTALL JRE 6.0 .....	SI-4
INSTALL MICROSOFT .NET FRAMEWORK VERSION 3.5 SP1 .....	SI-5
INSTALL APACHE TOMCAT VERSION 6.0.18.....	SI-6
ENSURE YOU HAVE INTERNET INFORMATION SERVICES INSTALLED .....	SI-7
SET EXCEL MACRO SECURITY .....	SI-8
INSTALL TM1 9.5 .....	SI-9
CREATE A SHORTCUT TO PERSPECTIVES ON THE DESKTOP .....	SI-11
SETUP ADVANCED RULES EDITOR.....	SI-12
CHANGE TM1 PASSWORD FOR PLANNING SAMPLE SERVER.....	SI-13
SET UP TM1 SERVERS .....	SI-14
DEPLOY TM1 CONTRIBUTOR TO THE TOMCAT SERVER .....	SI-15
TEST TM1 CONTRIBUTOR.....	SI-16
TEST TM1 CONTRIBUTOR.....	SI-17

CREATE USERS IN THE NTLM AUTHENTICATION PROVIDER .....	SI-18
INSTALL DB2 EXPRESS VERSION 9.5.....	SI-19
CREATE THE IBM COGNOS 8 BI CONTENT STORE .....	SI-21
SET UP WEB ALIASES .....	SI-23
CONFIGURE IBM COGNOS 8 .....	SI-24
ENSURE SIMPLE FILE SHARING IS OFF.....	SI-25
INITIALIZE THE IBM COGNOS TM1 APPLICATION.....	SI-26
CREATE A NEW INCOME_STATEMENT PLANNING APPLICATION ALONG WITH APPROPRIATE SECURITY .....	SI-30
INSTALL TM1 PORTLETS .....	SI-35
INSTALL TM1 WORKFLOW .....	SI-38
<b>INTRODUCTION.....</b>	<b>INTRO-1</b>
COURSE OBJECTIVES .....	INTRO-3
IBM COGNOS TM1: DESIGN AND DEVELOP MODELS (v9.5) .....	INTRO-5
ADDITIONAL TRAINING RESOURCES.....	INTRO-8
<b>INTRODUCTION TO IBM COGNOS TM1 .....</b>	<b>1-1</b>
OBJECTIVES .....	1-3
THE IBM COGNOS SOLUTION .....	1-4
FINANCIAL PERFORMANCE MANAGEMENT .....	1-5
TM1 OLAP SOLUTION.....	1-6
USE DIFFERENT TM1 USER INTERFACES.....	1-7
EXAMINE ROLES OF TM1 USERS .....	1-8
EXAMINE TM1 TECHNOLOGY.....	1-9
IMPORTANT FILES IN TM1 .....	1-10
DEMO 1: CONNECT TO AN EXISTING TM1 SERVER .....	1-11
DEMO 2: CREATE A NEW TM1 SERVER .....	1-14
SUMMARY .....	1-17

<b>CREATE DIMENSIONS .....</b>	<b>2-1</b>
OBJECTIVES .....	2-3
WHAT IS A DIMENSION?.....	2-4
CLASSES OF DIMENSIONS .....	2-5
NAMING DIMENSIONS .....	2-6
WHAT IS AN ELEMENT? .....	2-7
USING WEIGHTS WITH ELEMENTS .....	2-8
USING ATTRIBUTES.....	2-9
CREATE DIMENSIONS.....	2-10
DEMO 1: CREATE DIMENSIONS MANUALLY .....	2-11
DEMO 2: MODIFY EXISTING DIMENSIONS .....	2-15
WHAT IS TURBO INTEGRATOR? .....	2-22
KEY STEPS IN TI.....	2-23
DEMO 3: CREATE A DIMENSION USING TURBO INTEGRATOR .....	2-24
SUMMARY .....	2-31
WORKSHOP 1: CREATE A DIMENSION USING TURBO INTEGRATOR.....	2-32
<b>BUILD CUBES AND VIEWS.....</b>	<b>3-1</b>
OBJECTIVES .....	3-3
VIEW THE MODEL .....	3-4
STORE DATA IN CUBES .....	3-5
BUILD A CUBE.....	3-6
CONSIDER DIMENSION ORDER.....	3-7
NAME THE CUBE .....	3-8
CREATE A VIEW .....	3-9
DEMO 1: BUILD A CUBE.....	3-10
WHAT IS A PICK LIST .....	3-15
CREATING A SIMPLE PICK LIST.....	3-16
DEMO 2. CREATE A PICK LIST .....	3-18
SUMMARY .....	3-23
WORKSHOP 1: BUILD THE SALES PLAN CUBE .....	3-24

<b>LOAD AND MAINTAIN DATA .....</b>	<b>4-1</b>
OBJECTIVES .....	4-3
VIEW THE MODEL .....	4-4
DETERMINE THE DATA SOURCE .....	4-5
REVIEW THE DATA STRUCTURE .....	4-6
LOAD DATA WITH TURBO INTEGRATOR .....	4-7
DEMO 1: IMPORT DATA FROM AN ASCII FILE .....	4-8
USING FORMULAS IN VARIABLES .....	4-13
DEMO 2: IMPORT DATA FROM AN ODBC SOURCE .....	4-14
USE TURBO INTEGRATOR TO MAINTAIN THE MODEL .....	4-24
DEMO 3: CLEAR DATA FROM A CUBE USING A PROCESS .....	4-25
DEMO 4: ACCUMULATE DATA .....	4-29
DEMO 5: SAVE DATA .....	4-32
SUMMARY .....	4-34
WORKSHOP 1: CREATE THE FRANCHISE REVENUE CUBE .....	4-35
<b>ADD BUSINESS RULES .....</b>	<b>5-1</b>
OBJECTIVES .....	5-3
ADD BUSINESS LOGIC .....	5-4
VIEW THE MODEL .....	5-5
CREATE A RULE .....	5-6
USE A MODIFIER IN A RULE .....	5-7
REVIEW RULE PROCESSING .....	5-9
DEMO 1: CREATE SIMPLE RULES .....	5-10
USE A RULE TO OVERRIDE AGGREGATION .....	5-22
DEMO 2: CREATE A RULE TO SUPPRESS A CONSOLIDATION .....	5-23
USE A FUNCTION IN A RULE .....	5-31
DEMO 3: CREATE A RULE TO SHARE DATA BETWEEN CUBES .....	5-32
POPULATE A PICK LIST USING A RULE .....	5-37
CREATING A PICK LIST CONTROL CUBE .....	5-38
DEMO 4: CREATE A PICK LIST USING A RULE .....	5-39
USING RULE WORKSHEETS .....	5-49

DEMO 5: REVIEW A RULE WORKSHEET (INSTRUCTOR ONLY).....	5-50
DETERMINE LOCATION OF LOGIC .....	5-52
DEMO 6: CALCULATE GROSS MARGIN .....	5-53
SUMMARY .....	5-56
WORKSHOP 1: CALCULATE AVERAGE UNIT COST .....	5-57
WORKSHOP 2: COMPLETE CALCULATIONS FOR SALES PLAN MEASURES .....	5-62
<b>OPTIMIZE RULE PERFORMANCE.....</b>	<b>6-1</b>
OBJECTIVES .....	6-3
VIEW THE MODEL .....	6-4
UNDERSTAND CONSOLIDATIONS AND SPARSITY .....	6-5
EFFECT OF RULES ON SPARSE CONSOLIDATION .....	6-6
OPTIMIZE YOUR RULES USING SKIPCHECK .....	6-7
SKIPCHECK WITH CONSOLIDATIONS .....	6-8
DISCUSS A FEEDER STATEMENT .....	6-9
USING FEEDER STATEMENTS .....	6-10
INTER-CUBE FEEDERS .....	6-12
DEMO 1: ADD SKIPCHECK AND FEEDERS .....	6-13
TROUBLESHOOTING RULES AND FEEDERS.....	6-18
DEMO 2: TROUBLESHOOT RULES AND FEEDERS .....	6-19
FEEDING STRING RULES .....	6-24
SUMMARY .....	6-25
WORKSHOP 1: FEED RULES IN THE GREATOUTDOORS MODEL .....	6-26
<b>TRANSFER DATA INTO YOUR MODEL USING SCRIPTS.....</b>	<b>7-1</b>
OBJECTIVES .....	7-3
VIEW THE MODEL .....	7-4
LINK CUBES WITH DIFFERENT DIMENSIONS .....	7-5
REVIEW TURBO INTEGRATOR .....	7-6
ADDITIONAL TI PROCESS COMPONENTS .....	7-7
BENEFITS OF CUSTOM SCRIPTS.....	7-8
DEMO 1: LOAD DATA FROM MULTIPLE COLUMNS TO A SINGLE COLUMN.....	7-9
UPDATE A DIMENSION .....	7-19

<b>DEMO 2: CREATE A DIMENSION INCLUDING A SUBSET WITH TURBO INTEGRATOR .....</b>	<b>7-20</b>
DEALING WITH DATA.....	7-25
<b>DEMO 3: CREATE A DIMENSION WITH UNEVEN HIERARCHY .....</b>	<b>7-26</b>
USE TM1 AS A DATA SOURCE .....	7-35
<b>DEMO 4: EXPORT DATA TO ASCII TEXT USING TURBO INTEGRATOR .....</b>	<b>7-36</b>
SCRIPTING IN TURBO INTEGRATOR TIPS .....	7-41
<b>DEMO 5: MOVE DATA BETWEEN VERSIONS.....</b>	<b>7-42</b>
USE CHORES WITH TURBO INTEGRATOR .....	7-55
<b>DEMO 6: CREATE AND SCHEDULE A CHORE.....</b>	<b>7-56</b>
SUMMARY .....	7-57
<b>WORKSHOP 1: CREATE SUBSETS WITHIN A TURBO INTEGRATOR PROCESS ..</b>	<b>7-58</b>
<b>CUSTOMIZE DRILL PATHS .....</b>	<b>8-1</b>
OBJECTIVES .....	8-3
VIEW RELATED DATA.....	8-4
CREATE A DRILL PATH .....	8-5
<b>DEMO 1: CREATE A DRILL-THROUGH PATH TO A VIEW IN ANOTHER CUBE ....</b>	<b>8-6</b>
<b>DEMO 2: CREATE A DRILL PATH TO AN ODBC SOURCE .....</b>	<b>8-14</b>
SUMMARY .....	8-19
<b>WORKSHOP 1: CREATE DRILL PATHS FOR COMMISSIONS .....</b>	<b>8-20</b>
<b>USING RULES FOR ADVANCED MODELING.....</b>	<b>9-1</b>
OBJECTIVES .....	9-3
VIEW THE MODEL .....	9-4
DISCUSS A VIRTUAL CUBE.....	9-5
UTILIZE A LOOKUP CUBE.....	9-6
USE RELATIVE SPREADING .....	9-7
USE A SPREAD PROFILE CUBE .....	9-8
<b>DEMO 1: CREATE THE EXPENSES CUBE .....</b>	<b>9-9</b>
<b>DEMO 2: CREATE SPREAD PROFILE CUBE .....</b>	<b>9-26</b>
USE ATTRIBUTES IN RULES.....	9-38
<b>DEMO 3: MOVE BALANCES FROM LAST MONTH TO NEXT MONTH.....</b>	<b>9-40</b>
SUMMARY .....	9-56

<b>CONVERT CURRENCIES .....</b>	<b>10-1</b>
OBJECTIVES .....	10-3
VIEW THE MODEL .....	10-4
CONVERT CURRENCY .....	10-5
REVIEW CONTROL CUBES.....	10-6
DEMO 1: INCLUDE REPORTING CURRENCIES IN A CUBE.....	10-7
SUMMARY .....	10-25
WORKSHOP 1: BUILD THE INCOME STATEMENT CUBE WITH REPORTING CURRENCIES .....	10-26
<b>MODEL FOR DIFFERENT FISCAL REQUIREMENTS.....</b>	<b>11-1</b>
OBJECTIVES .....	11-3
VIEW THE MODEL .....	11-4
UNDERSTANDING TIME.....	11-5
DISCRETE TIME DIMENSIONS.....	11-6
DEMO 1: ADD A YEAR DIMENSION .....	11-7
CONTINUOUS TIME DIMENSIONS .....	11-12
DEVELOP CONTINUOUS TIME MODEL .....	11-13
DEMO 2: CREATE A CONTINUOUS TIME MODEL.....	11-14
SUMMARY .....	11-27
WORKSHOP 1: ADD LAST YEAR CALCULATIONS TO FRANCHISE REVENUE3 .....	11-28
WORKSHOP 2: ADD FRANCHISE FEES TO THE INCOME_STATEMENT CUBE.	11-33
<b>INTRODUCTION TO MANAGED PLANNING APPLICATIONS.....</b>	<b>12-1</b>
OBJECTIVES .....	12-3
TM1 CONTRIBUTOR (MANAGED PLANNING APPLICATIONS) .....	12-6
TM1 CONTRIBUTOR WEB CLIENT .....	12-7
TM1 CONTRIBUTOR ADMINISTRATION .....	12-8
EXAMINE TM1 THREE TIER ARCHITECTURE.....	12-9
PURPOSE AND BENEFITS OF MANGED PLANNING APPLICATIONS.....	12-10
INITIALIZE THE MANAGED PLANNING ENVIRONMENT .....	12-12
IBM COGNOS TM1 SUPPORTED PLATFORMS .....	12-13
SUMMARY .....	12-14

## CONTRIBUTE DATA TO MANAGED PLANNING APPLICATIONS 13-1

OBJECTIVES .....	13-3
ACCESS A TM1 CONTRIBUTOR APPLICATION.....	13-4
OPEN AN APPLICATION .....	13-5
EXAMINE WORKFLOW .....	13-6
EXAMINE THE ADMINISTRATOR ROLE.....	13-7
EXAMINE THE CONTRIBUTOR ROLE.....	13-8
EXAMINE THE REVIEWER ROLE.....	13-9
IDENTIFY WORKFLOW STATES .....	13-10
DEMO 1: EXAMINE THE WORKFLOW SCREEN.....	13-11
NAVIGATE THE CONTRIBUTOR WORKSPACE.....	13-13
REORGANIZE THE GRID.....	13-14
ADD CHARTS .....	13-15
EXPORT DATA.....	13-16
DEMO 2: OPEN A NODE AND MODIFY THE VIEW .....	13-18
TAKE OWNERSHIP OF A GRID .....	13-24
ENTER AND DISTRIBUTE DATA.....	13-25
USE SHORTCUTS TO MANUALLY ENTER DATA .....	13-26
USE SHORTCUTS TO MANUALLY ENTER DATA (CONT'D).....	13-27
COPY DATA QUICKLY .....	13-28
ENTER AND CHANGE DATA QUICKLY .....	13-29
COMPARE DATA ENTRY COMMANDS .....	13-31
EXAMINE DATA COLORS .....	13-33
COMMIT DATA .....	13-34
COMMIT AND SUBMIT DATA.....	13-35
TAKE OWNERSHIP FROM A USER .....	13-36
DEMO 3: ENTER, SAVE AND SUBMIT DATA .....	13-37
DEMO 4: REVIEW PLAN DATA .....	13-43
SUMMARY .....	13-45
WORKSHOP 1: ENTER, SUBMIT, AND REVIEW DATA .....	13-46

<b>CREATE AND DEPLOY MANAGED PLANNING APPLICATIONS TO THE WEB .....</b>	<b>14-1</b>
OBJECTIVES .....	14-3
IBM COGNOS TM1 CONTRIBUTOR ADMINISTRATION.....	14-4
MANAGED PLANNING ADMINISTRATION .....	14-5
MANAGED PLANNING APPLICATION.....	14-6
CREATE A MANAGED PLANNING APPLICATION.....	14-7
DEMO 1: CREATE A MANAGED PLANNING APPLICATION.....	14-8
ASSIGN SECURITY .....	14-16
TM1 OBJECT ACCESS .....	14-18
DEMO 2: ASSIGN SECURITY TO A MANAGED PLANNING APPLICATION .....	14-19
SUMMARY .....	14-27
WORKSHOP 1: CREATE A MANAGED PLANNING APPLICATION FOR THE GREATOUTDOORS BUDGET PROCESS.....	14-28
<b>DEFINE WORKFLOW .....</b>	<b>A-1</b>
OBJECTIVE .....	A-3
CONFIGURE TM1 FOR WORKFLOW.....	A-4
REVIEW TM1 SECURITY .....	A-5
DEMO 1: ADD USERS AND GROUPS TO TM1 SECURITY .....	A-6
INSTALL WORKFLOW .....	A-11
DEMO 2: ADD TM1 WORKFLOW TO A TM1 SERVER .....	A-12
CREATE A WORKFLOW PROCESS .....	A-14
DEMO 3: ADD A NEW PROCESS AND TASKS .....	A-15
USE THE TOOLBAR.....	A-27
DEMO 4: SUBMIT BUDGET FOR GO AMERICAS EXPENSES .....	A-28
SUMMARY .....	A-38

<b>INTEGRATE WITH IBM COGNOS 8 BI AND IBM COGNOS 8 PLANNING .....</b>	<b>B-1</b>
OBJECTIVES .....	B-3
EXAMINE PERFORMANCE MANAGEMENT (PM).....	B-4
IBM COGNOS 8 BI OVERVIEW.....	B-5
IBM COGNOS 8 BI COMPONENTS .....	B-6
IBM COGNOS 8 GROUPS AND ROLES.....	B-7
WHAT IS FRAMEWORK MANAGER? .....	B-8
BENEFITS OF IBM COGNOS 8 INTEGRATION.....	B-9
INTEGRATE IBM COGNOS TM1 WITH IBM COGNOS 8: STEPS .....	B-10
DEMO 1: INTEGRATE IBM COGNOS TM1 WITH IBM COGNOS 8 BI .....	B-11
PORTLETS OVERVIEW .....	B-20
PORTLET OPTIONS .....	B-21
DEMO 2: CREATE A PORTAL WITH TM1 WEB.....	B-22
EXAMINE IBM COGNOS 8 PLANNING .....	B-28
IBM COGNOS 8 PLANNING COMPONENTS .....	B-29
BUILD MODELS IN ANALYST .....	B-30
DEPLOY APPLICATIONS USING CONTRIBUTOR .....	B-31
USE TM1 DATA IN AN IBM COGNOS PLANNING APPLICATION .....	B-32
USE IBM COGNOS PLANNING DATA IN A TM1 APPLICATION.....	B-33
SUMMARY .....	B-34
<b>DESIGN FOR REPORTING.....</b>	<b>C-1</b>
OBJECTIVES .....	C-3
ORGANIZE CONTENT IN A TM1 APPLICATION .....	C-4
USING APPLICATION FOLDERS: EXAMPLE.....	C-5
DEMO 1: ADD TM1 OBJECTS AND EXTERNAL FILES TO AN APPLICATION .....	C-6
BUILD WORKSHEETS WITH ACTION BUTTONS .....	C-13
ADD AN ACTION BUTTON TO A WORKSHEET .....	C-14
DEMO 2: ADD AN ACTION BUTTON TO A WORKSHEET.....	C-15
USE WORKSHEET CONTROLS TO PRESENT DATA .....	C-21
USE CONTROLS FOR DATA VALIDATION .....	C-22

USE TOOLBARS TO ADD CONTROLS .....	C-23
DEMO 3: ADD A DATA VALIDATION CONTROL TO A MICROSOFT EXCEL SPREADSHEET .....	C-24
FORMAT REPORTS.....	C-28
DEMO 4: CHANGE THE FORMATTING ON A REPORT AND CREATE A CUSTOM MENU .....	C-29
SUMMARY .....	C-33
<b>OPTIMIZE AND TUNE TM1 MODELS.....</b>	<b>D-1</b>
OBJECTIVES .....	D-3
TM1 MODEL SIZE .....	D-4
MODELS MAY GROW RAPIDLY.....	D-5
IMPACT TO THE MODEL PERFORMANCE .....	D-6
IMPROVE INDIVIDUAL CUBE PERFORMANCE.....	D-7
SIMPLE RULE & FEEDER ENHANCEMENTS .....	D-8
OVERVIEW OF TM1 SERVER OPERATIONS .....	D-9
IDENTIFY SERVER OPERATIONS ON STARTUP.....	D-10
IDENTIFY SERVER OPERATIONS FOR DATA REQUESTS.....	D-11
SCALING.....	D-12
MIRRORED SERVERS .....	D-13
IMPROVE SERVER PERFORMANCE.....	D-14
SUMMARY .....	D-16

<b>IDENTIFY COMMON DATA STRUCTURES.....</b>	<b>E-1</b>
OBJECTIVES .....	E-3
EXAMINE THE ROLE OF AN IBM COGNOS TM1 METADATA MODEL .....	E-4
DISTINGUISH BETWEEN OPERATIONAL AND REPORTING DATABASES.....	E-5
IDENTIFY FEATURES OF AN OPERATIONAL DATABASE .....	E-6
IDENTIFY ISSUES WITH OPERATIONAL DATABASES .....	E-7
EXAMINE REPORTING DATABASES (STAR SCHEMA DESIGN).....	E-8
CREATE A STAR SCHEMA.....	E-9
EXAMINE OPERATIONAL DATA .....	E-10
EXAMINE REPORTING DATA .....	E-11
EXAMINE FACT TABLES.....	E-12
EXAMINE DIMENSION TABLES.....	E-13
IDENTIFY ISSUES WITH A STAR SCHEMA .....	E-14
WHAT IS OLAP? .....	E-15
OLAP VS. OLTP (RELATIONAL).....	E-16
DIMENSIONAL ANALYSIS EXAMPLE .....	E-17
EXAMINE OLAP DATA STRUCTURES .....	E-18
THE THREE CORE DESIGN PRINCIPLES .....	E-19
CREATE A MODEL IN TM1.....	E-20
VIEW THE MODEL .....	E-21
SUMMARY .....	E-22

# Course Overview

## Course Overview

**IBM Cognos TM1: Design and Develop Models (v9.5)** is a five-day, instructor-led course designed to teach modelers how to build a complete model in TM1. Through a series of lectures and hands-on exercises, students will learn how to set up and verify structures, manually enter data into these structures, and define the data users can see.

## Intended Audience

- Modelers, who design and build multidimensional models, map data flow, and apply business logic. May analyze data and perform predictive analysis. May create and customize reports, templates, and applications

## Topics Covered

Topics covered in this course include:

- Examine the components of a TM1 model
- Create dimensions
- Build cubes and views
- Load and maintain data
- Add business rules
- Optimize business rules
- Transfer data into the model using scripts
- Customize drill paths

- Use rules for advanced modeling
- Convert currencies
- Model for different fiscal requirements
- Analyze data in TM1 Web
- Contribute Data to Managed Planning Applications
- Create and Deploy Managed Planning Applications
- Define Workflow (Optional)
- Integrate TM1 with IBM Cognos 8 BI and IBM Cognos Planning (Optional)
- Present Data and Reports (Optional)
- Optimize and Tune TM1 Models (Optional)
- Identify Common Data Structures

## **Course Prerequisites**

Participants should have:

- Successfully completed the IBM Cognos TM1: Analyze Data (v9.5) course
- Basic knowledge of OLAP and TM1 (TM1 Analyst)
- Significant experience with Excel spreadsheets (functions, macros, etc.)
- Understanding of the metrics and drivers of your business

## Important Course Changes

### New features for IBM Cognos TM1: Design and Develop Models (v9.5)

#### General Changes

- TM1 9.5 includes the new TM1 Contributor component, that lets modelers create and deploy managed planning applications (MPA). MPAs allow multiple users to review, edit and contribute to plans on the Web using built-in workflow. This course shows modelers how to contribute to a TM1 Contributor MPA and how to create a simple managed planning application using TM1 Contributor Administration.
- TM1 9.5 now supports the use of pick lists in cubes. This course shows modelers how to create pick lists using attributes and rules.
- This course primarily uses TM1 Architect and the Advanced Rules Editor for model and rule development. The Advanced Rules Editor will not run under TM1 Perspectives on the VMWare images provided, instead, use TM1 Architect.
- Text files (\*.rux) have been provided in order to copy rules and then to paste them into the rules editor. There are also some (\*.txt) text files containing SQL or TI scripts.
- Creating business rules and feeders are now in separate modules.
- This course will also be offered as an instructor led on-line (ILO) course.
- There are ILO Interactions at the bottom of several pages to facilitate delivery of the course.
- TM1 Workflow is discussed as an optional appendix.

- Design for Reporting has been moved from IBM Cognos TM1: Analyze Data to Appendix C in this course.
- There is now a new slide, View the Model, depicting source file and cube relationships.
- There is a new Appendix E: Identify Common Data Structures that reviews data from data sources and how it can be mapped to TM1 objects. It should be taught after Module 1 but before Module 2.
- There is a new optional Appendix D that discusses some tips and techniques for optimizing and maintaining your TM1 model.

## **Introduction:**

- This course is now aimed at modelers. There is no longer an "analyst" role.

## **Module 1: Introduction to IBM Cognos TM1**

- Update to demo: Create a New TM1 Server.

## **Module 3: Build Cubes and Views**

- New slide: View the Model.
- New slide: What is a Pick List?
- New slide: Creating a Simple Pick List.
- New demo: Create a Simple Pick list in a cube.

## **Module 4: Load and Maintain Data**

- New slide: View the Model.

## **Module 5: Add Business Rules**

- New slide: View the Model.
- New slide: Populate a Pick List Using a Rule.
- New slide: Creating a Pick List Control Cube.
- New demo: Create a Pick List Using a Rule.

## Module 6: Optimize Rule Performance

- New module.

## Module 7: Transfer Data into Your Model Using Scripts

- Module renamed from Automate Common Processes Using Scripts.
- New slide: View the Model.
- New slide: Dealing with Data.
- New slide: Scripting in Turbo Integrator Tips.
- Updated demo: Move Data Between Versions.

## Module 9: Using Rules for Advanced Modeling

- New slide: View the Model.

## Module 10: Convert Currencies

- New slide: View the Model.

## Module 11: Model for Different Fiscal Requirements

- New slide: View the Model.
- New workshop: Add Franchise Fees to the Income\_Statement cube.

## Module 12: Introduction to Managed Planning Applications

- New module.

## Module 13: Contribute Data to Managed Planning Applications

- New module.

## Module 14: Create and Deploy Managed Planning Applications to the Web

- New module.

## **Appendix A: Define Workflow**

- New model for demos.

## **Appendix C: Design for Reporting**

- New module.

## **Appendix D: Optimize and Tune TM1 Models**

- New module.

## **Appendix E: Identify Common Data Structures**

- New module. This module should be discussed prior to Module 2: Create Dimensions.

# Course Outline

The following table outlines the high-level topics for each module, and the number of slides, demos and workshops included, as well as the estimated teaching time.

<b>Module 1: Introduction to IBM Cognos TM1</b>				
Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Review financial performance management</li> <li>• Identify the TM1 position in the financial performance management system</li> <li>• Describe the architecture and components of TM1</li> <li>• Connect to a TM1 server</li> </ul>	12	2	0	1 hrs.
<b>Module 2: Create Dimensions</b>				
Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Review cubes and dimensions</li> <li>• Create dimensions manually</li> <li>• Edit dimensions</li> <li>• Create dimensions using Turbo Integrator</li> </ul>	17	3	1	1.5 hrs.

## Module 3: Build Cubes and Views

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss cubes and data points</li> <li>• Create cubes</li> <li>• Construct views of data in cubes</li> <li>• Create and use a pick list</li> </ul>	14	2	1	1 hr.

## Module 4: Load and Maintain Data

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Identify data sources</li> <li>• Create processes to load data</li> <li>• Create a process to delete data in a cube</li> <li>• Create processes to update and maintain the model</li> </ul>	15	5	1	2 hrs.

## Module 5: Add Business Rules

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss a rule</li> <li>• Construct rules for elements or consolidations</li> <li>• Use functions in rules</li> <li>• Populate a pick list using a rule</li> <li>• Discuss rules in a Rule Worksheet</li> </ul>	20	6	2	3 hrs.

## Module 6: Optimize Rule Performance

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss consolidations and sparsity</li> <li>• Enhance consolidation performance using SKIPCHECK</li> <li>• Use FEEDERS to optimize aggregations</li> <li>• Check accuracy of rules and feeders</li> </ul>	17	2	1	1.5 hrs.

## Module 7: Transfer Data into Your Model Using Scripts

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Review Advanced tabs in Turbo Integrator</li> <li>• Load custom data into a cube</li> <li>• Add a subset to a dimension</li> <li>• Use logic in scripts</li> <li>• Export data to ASCII</li> <li>• Move data between versions</li> <li>• Construct chores</li> </ul>	19	6	1	2.5 hrs.

## Module 8: Customize Drill Paths

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• View related data</li> <li>• Create a Drill Process</li> <li>• Create a Drill Assignment Rule</li> <li>• Edit a drill process</li> </ul>	8	2	0	1 hr.

## Module 9: Using Rules for Advanced Modeling

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss virtual cubes</li> <li>• List uses for lookup cubes</li> <li>• Create and use a spread profile cube</li> <li>• Implement moving balances in a cube</li> </ul>	12	3	0	2.5 hrs.

## Module 10: Convert Currencies

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss currency challenges</li> <li>• Create currency dimension</li> <li>• Create currency cube</li> <li>• Create rules for currency conversion</li> <li>• Use TM1 techniques to reduce maintenance</li> </ul>	8	12	1	1.5 hrs.

## Module 11: Model for Different Fiscal Requirements

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Discuss time considerations</li> <li>• Use discreet time dimensions</li> <li>• Implement a continuous time dimension model</li> </ul>	12	2	2	1.5 hrs.

## Module 12: Introduction to Managed Planning Applications

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• review the purposes and benefits of managed planning applications</li> <li>• review supported platforms</li> </ul>	11	0	0	1 hr.

## Module 13: Contribute Data to Managed Planning Applications

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Access a TM1 Contributor application</li> <li>• Identify the user roles and workflow states</li> <li>• Enter data using shortcuts and spreading</li> <li>• Submit and reject data</li> <li>• Reorganize the grid</li> </ul>	30	4	1	3 hrs.

## Module 14: Create and Deploy Managed Planning Applications to the Web

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>Understand the purpose and benefits of a Managed Planning Application</li> <li>Create a Managed Planning Application</li> <li>Apply security to a Managed Planning Application</li> <li>Examine design techniques and best practices for building Managed Planning Applications</li> </ul>	13	2	1	2.5 hrs.

## Appendix A: Define Workflow

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>Configure TM1 for Workflow</li> <li>Install Workflow</li> <li>Create Workflow processes</li> <li>Use Workflow toolbars to review versions</li> </ul>	13	4	0	2.5 hrs.

## Appendix B: Integrate with IBM Cognos 8 BI and IBM Cognos Planning

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Integrate TM1 cubes within a IBM Cognos 8 business intelligence (BI) environment</li> <li>• Integrate IBM Cognos TM1 in IBM Cognos 8 BI portals</li> <li>• Describe how to use TM1 data in an IBM Cognos 8 Planning application</li> <li>• Describe how to use IBM Cognos 8 Planning data in a TM1 application</li> </ul>	13	4	0	2 hrs.

## Appendix C: Design for Reporting

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Organize content in TM1 using application folders</li> <li>• Use Action buttons to enhance functionality in Microsoft Excel and TM1 Web reports</li> <li>• Add worksheet controls to change the presentation of data</li> </ul>	15	4	0	2 hrs.

## Appendix D: Optimize and Tune TM1 Models

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• Identify characteristics of large models</li> <li>• Discuss strategies to improve model performance</li> </ul>	15	0	0	1 hrs.

## Appendix E: Identify Common Data Structures

Topics	Slides	Demos	Workshops	Est. Time
<ul style="list-style-type: none"> <li>• identify characteristics of operational and reporting databases</li> <li>• discuss Online Analytical Processing (OLAP)</li> <li>• examine core model design principles</li> <li>• identify the basic tasks required to create a TM1 model and deploy it</li> </ul>	22	0	0	1 hrs.

# Instructional Materials

## Student Guide

The Student Guide contains material that helps to explain features of the product, along with the presentation slides that are presented by the instructor. Student demos and workshops are incorporated in the course to enrich the learning experience through hands-on practice.

### Demos

Demos appear after covering one or more topics or features of the application. While not every product function is demonstrated, participants work with the more important and complex features through a series of tasks. Demo tasks contain a number of steps related to a specific action or feature of the product.

### Workshops

In most of the modules, a supplementary workshop is included. If participants followed the concepts in class without difficulties, they can probably complete the workshop with no additional information. The second section for each workshop contains a task table that identifies each task, where to work in the application, and any applicable hints to help the participants. The third section of the workshop contains screen captures of the expected results. The fourth section contains a step-by-step solution to the workshop. Participants may want to follow these instructions if they are not able to complete the workshop or if they require a little more practice with the application.

## Instructor Guide

The Instructor Guide contains the same content presented in the Student Guide, along with additional notes to supplement and add value to the lecture. The information can be generic, non-technical information, such as multiple ways to perform the same command or a more in-depth discussion of a topic. It may also be used to address more technical questions from participants or as supplementary technical discussion, at the discretion of the instructor. It helps to provide the appropriate level of information to a specific audience.

## Instructor Data Download

The Instructor Data Download is an executable (EXE) file that can install any or all of the following files. By launching this executable and following the prompts as the install runs, these files will be installed in C:\Edcognos\P6502.

## Instructor Slides

These files contain the Microsoft PowerPoint slide presentation for each module of the course as presented in the Student Guide:

- P6502\_Start.ppt
- P6502\_Intro.ppt
- TMD-01-Intro\_to\_TM1.ppt
- TMD-02-Create\_Dimensions.ppt
- TMD-03-Build\_Cubes\_and\_Views.ppt
- TMD-04-Load\_and\_Maintain\_Data.ppt
- TMD-05-Add\_Business\_Rules.ppt
- TMD-06-Optimize\_Rule\_Performance.ppt
- TMD-07-Transfer\_Data\_into\_Your\_Model\_Using\_Scripts.ppt
- TMD-08-Customize\_Drill\_Paths.ppt
- TMD-09-Using\_Rules\_for\_Advanced\_Modeling.ppt
- TMD-10-Convert\_Currencies.ppt
- TMD-11-Model\_for\_Different\_Fiscal\_Requirements.ppt

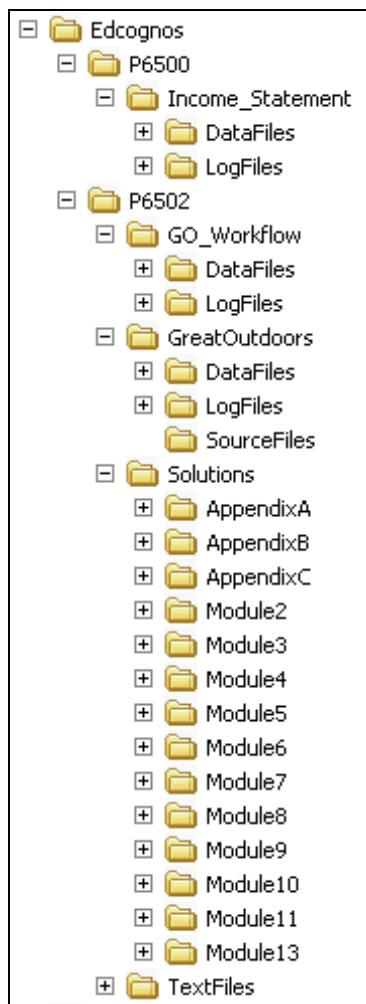
- TMD-12-Introduction\_to\_Managed\_Planning\_Applications.ppt
- TMD-13-  
Contribute\_Data\_to\_Managed\_Planning\_Applications.ppt
- TMD-14>Create\_and\_Deploy\_Managed\_Planning\_  
Applications\_to\_the\_Web.ppt
- TMD-A-Define\_Workflow.ppt
- TMD-B-Integrate\_with\_C8\_and\_CP.ppt
- TMD-C-Design\_for\_Reporting.ppt
- TMD-D-Optimize\_and\_Tune\_TM1\_Models.ppt
- TMD-E-Identify\_Common\_Data\_Structures.ppt

## PDF version of the Instructor Guide

These files are the Instructor Guide Microsoft Word documents in PDF.

## Student Data

The EXE contains the files that are necessary to complete the demos and workshops. By installing on the student PCs, the following files and folders will be installed to the C:\Edcognos directory:



- P6500\Income\_Statement subfolder (contains a complete TM1 application including a Managed Planning Application)
- P6502\GO\_Workflow (contains a complete TM1 application for TM1 Workflow)
- P6502\GreatOutdoors subfolder (contains a complete TM1 application)
- P6502\TextFiles (contains text files containing rules, SQL and scripts to use in demos and workshops)

## Solution Files

Solution files will be installed to C:\Edcognos\P6502\Solutions. To teach a module where participants have not completed all demos and workshops up to that point, delete the DataFiles folder from C:\Edcognos\P6502\GreatOutdoors, and replace it with the DataFiles folder from the last module that should have been completed. Module 13 uses the TM1 server in P6500\Income\_Statement. Appendix A contains the DataFiles for the GO\_Workflow TM1 server.

## Demos, Workshops, and Workshop Solutions

The course is designed to be easily customized for on-site training with customer data. The files contain Microsoft Word files that you can use to modify the demonstrations and workshops as required.

## General Setup and Instructor Preparation

### Pre-Class Agenda

To ensure that the class runs smoothly, you should know the answers to the following questions. If you need help in obtaining answers, contact the customer or customer's sales representative (if the course is scheduled for a client site), or the local office responsible for course logistics.

- Who is the contact person for class setup?
- What is the classroom setup? Is there a white board? Is there a flip chart? Is there a computer for the instructor, a PC viewer, overhead projector, and screen?
- Will the physical environment be set up prior to your arrival (product loaded and PowerPoint files on the computer)?
- What time does the class start?
- What hours are available for accessing the teaching site, copying the files to the hard disk, tuning the color on the PC viewer, and so on?
- What Cognos office is responsible for sending the Student Guides?
- If the course has been previously taught on the computers you are using, have the Preferences been reset to their defaults, and have student files been deleted?

## Prepare to Teach

After you have configured the instructor and student computers, consider the following:

- Run through at least one module in a classroom with a PC viewer.
- Run through the full course at least once on a computer.
- Make sure you complete each of the demos before teaching the course so that you become familiar with each step required.
- Have a set of product reference manuals in the classroom.
- Make sure that there is a Student Guide for each participant.

## Document Conventions

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

### **Bold**

Bold style is used in demo and workshop step-by-step solutions to indicate either:

- actionable items  
(Point to **Sort**, and then click **Ascending**.)
- text to type or keys to press  
(Type **Sales Report**, and then press **Enter**.)
- UI elements that are the focus of attention  
(In the **Format** pane, click **Data**)

### *Italic*

Used to reference book titles.

### CAPITALIZATION

All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application.

To keep capitalization consistent with this guide, type text exactly as shown.

# Workshops

## Workshop Format

Workshops are designed to allow you to work according to your own pace. The workshops are structured as follows:

### The Business Question Section

The first page of each workshop presents a business-type question followed by a series of steps. These steps provide additional information to help guide you through the workshop. Within each step, there may be numbered questions relating to the step. Solve the tasks by using the skills you learned in this module and in previous ones. If you need more assistance, you can refer to the Task Table section that provides more detailed instruction.

### The Task Table Section

The second page of the workshop is a Task Table that presents the question as a series of numbered tasks to be accomplished. The first column in the table states the task to be accomplished. The second column, "Where to Work", indicates the area of the product to work in. Finally, the third column provides some hints that may help you complete the workshop. If you need more assistance to complete the workshop, please refer to the Step-by-Step Instructions at the end of the workshop.

### The Workshop Results Section

This section will contain a screen capture(s) of interim or final results and/or answers to the questions asked in the Business Question section.

### The Step-by-Step Section

The Step-by-Step instructions for completing all of the tasks are located at the end of the workshop following the Workshop Results section. Each task in the Task Table is expanded into numbered steps, scripted like the demos.

## PowerPoint Tips

Here are valuable keyboard commands you can use to improve your presentation.

Command	Key(s)
Advance to next slide	Left-click, Page Down, Space, N, Right or Down Arrow, right-click/Next, Enter
Return to previous slide	Backspace, Page Up, P, Left or Up Arrow, right-click/Previous
Change pointer to a pen	Right-click/Pen or Ctrl+P
Erase drawings on screen	E
Make the screen white	W or ',' (toggle to restore)
Make the screen black	B or '.' (toggle to restore)
Help	?
End the slide show	Esc, Ctrl+Break, '-'
Move between PowerPoint and the product	Alt+Tab or click the application name on the status bar

You can also jump to a specific slide by typing its slide number and pressing the Enter key. However the slide number is not the same as the printed page number because a page may be built from several slides to produce an animation sequence.

### Important Tips:

- A page containing an animation slide (multiple clicks to complete the slide) will also include an Instructor Guide note indicating the number of clicks needed to complete the slide.

## VMware Keyboard Shortcuts

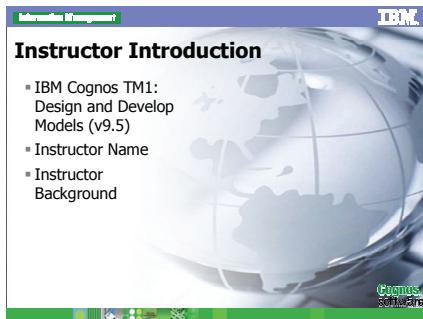
Below is a list of shortcuts that can be used when in a VM image.

Shortcut	Action
Ctrl-B	Power on.
Ctrl-E	Power off.
Ctrl-R	Reset the power.
Ctrl-Z	Suspend.
Ctrl-N	Create a new virtual machine.
Ctrl-O	Open a virtual machine.
Ctrl-F4	Close the summary/console view for the selected virtual machine. A confirmation dialog appears only if the virtual machine is powered on.
Ctrl-D	Edit the virtual machine's configuration.
Ctrl-G	Grab input from keyboard and mouse.
Ctrl-P	Edit preferences.
Ctrl-Alt-Enter	Go to full screen mode.
Ctrl-Alt	Return to normal (windowed) mode.
Ctrl-Alt-Tab	Switch among open virtual machines while mouse and keyboard input are grabbed.
Ctrl-Tab	Switch among open virtual machines while mouse and keyboard input are not grabbed. VMware Workstation must be the active application.

Shortcut	Action
Ctrl-Shift-Tab	Switch among open virtual machines while mouse and keyboard input are not grabbed. VMware Workstation must be the active application.
Ctrl-Alt-Fx	<p>Linux hosts: Switch among open virtual machines while using full screen mode. Fx is a function key corresponding to the virtual machine you want to use. The key combination to use for a virtual machine is shown in the VMware Workstation title bar when that virtual machine is active and in normal (windowed) mode.</p> <p>Windows hosts: For an additional similar functionality, see <a href="#">Using Full Screen Switch Mode</a>.</p>

## Get the Class Started

Welcome participants to the course. Use the slide show, P6502\_Start.PPT, to introduce yourself, the participants, and the agenda for the training (optional).



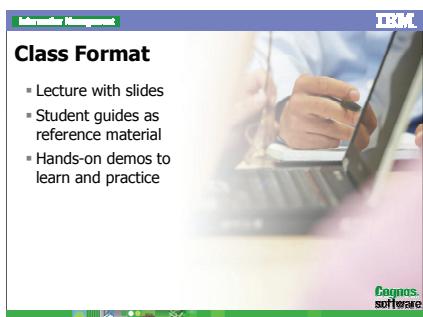
Use this slide to welcome the participants to the course, to introduce yourself, and to mention your background (for example, how long you have been teaching the course, your teaching experience overall, how long you have been working with the product, and so on). Make sure you have customized the slide ahead of time.



Have the participants take turns introducing themselves with respect to the items listed on this slide. The intent of the slide is to act as an icebreaker and to encourage participation.



Use this slide to go through the list of administrative items that participants often ask about.



Use this slide to explain the class format and emphasize that participants are encouraged to actively perform the hands-on demos while following along with the instructor. Mention that the Student Guide contains copies of the slides and further supporting notes for the participants to use as reference material in the future.

## Post-Class Agenda

- Have participants complete the Course Evaluation forms.
- Hand out certificates.
- Leave the classroom clean.
- If you brought any hardware or course media, take them with you when you leave. Erase any files copied to the hard disks of the computers in the classroom. Change the Preferences back to their initial settings.
- Complete the Instructor Feedback form, and return it to the Education Coordinator. Ensure that the Coordinator receives the Course Evaluation forms.
- If you are at a customer site, thank the course administrator by letter.
- List any outstanding questions, and ensure that participants receive answers in writing.
- Report any sales leads to your sales representative.
- Make notes for yourself about what went well during the course and what needs improvement. When you are preparing for your next teach, you can refer to these.

## Submit Feedback and Locate Additional Product Information

### Submit Feedback

Your feedback is important and valuable. We are interested in your comments or questions. If you have feedback for a course, you may directly submit it to our online database using the following link:  
<http://sottgesops.ottawa.ibm.com/feedback/default.htm>. The link to this feedback database can also be found in the Instructor Support section of the IBM Cognos Education wiki.

You may also send general comments or concerns to Global Education Services at [Cognos.ges@ca.ibm.com](mailto:Cognos.ges@ca.ibm.com).

In the Instructor Support area on the IBM Cognos Education wiki, your suggested course corrections and course-related supplementary information (e.g., additional demos you use, diagrams you created) will be posted within the Instructor Zone under Course Updates & Corrections, and teaching tips and techniques will be posted within the Instructor Zone under Share Your Knowledge. This gives other worldwide instructors immediate use of the information. Technical course developers will also use this information when the course is updated.

Access the Instructor Support Community on the IBM Cognos Education wiki at

<https://w3.tap.ibm.com/w3ki07/display/IBMCogEd/Instructor+Support>. You will also find additional course information (e.g., course release status or course development templates) in the Instructor Support Community, which you may find useful when preparing for courses.

Suggestions for significant course enhancements will be reviewed for possible implementation in future course updates.

## IBM Product Help

Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	<i>IBM Product- Help link</i>
Books for Printing (.pdf)	<p>You want to use search engines to find information. You can then print out selected pages, a section, or the whole book.</p> <p>Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book.</p> <p>The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.</p>	Start/All Programs/ <i>IBM Product/Documentation</i>
IBM on the Web	<p>You want to access any of the following:</p> <ul style="list-style-type: none"> <li>• Services and Training Web site</li> <li>• Online support</li> <li>• IBM Web site</li> </ul>	<p><a href="http://www.ibm.com/Cognos/services">http://www.ibm.com/Cognos/services</a></p> <p><a href="http://www.ibm.com/Cognos/support">http://www.ibm.com/Cognos/support</a></p>

**Information Management**



# Setup Instructions

IBM Cognos TM1 9.5



**Cognos**  
software  
*Confidential*

© 2009 IBM Corporation

## Setup Instructions for IBM Cognos TM1 9.5 P6502

Use the following checklist when configuring both the instructor and student computers. The total time to set up the instructor and student computers is listed in the table.

<b>Setup</b>	<b>Page</b>	<b>Est. Time (per Computer)</b>	<b>Complete</b>
Ensure you have a file compression utility.		1 min	
Install JRE 6.0	SI-4	2 min	
Install Microsoft .NET Framework version 3.5 SP1	SI-5	5 mins	
Install Apache Tomcat version 6.0.18	SI-6	2 mins	
Ensure IIS is installed	SI-7	1 mins	
Set Excel Macro Security	SI-8	1 mins	
Install TM1 9.5	SI-9	5 mins	
Create a shortcut to Perspectives on the desktop	SI-11	1 min	
Setup the Advanced Rules Editor	SI-12	5 mins	
Change TM1 password for Planning Sample server	SI-13	5 mins	
Set up TM1 servers.	SI-14	3 mins.	
Deploy TM1 Contributor to the Tomcat Server	SI-15	2 mins	
Test TM1 Contributor	SI-16		
Adjust Internet Explorer Security	SI-17	1 min	
Create Users in the NTLM authentication provider	SI-18	2 mins	

Install DB2 Express Version 9.5	SI-19	1 mins	
Perform a default install of IBM Cognos 8.4 (from \\sotpprodbldf\Cognos_Prod\Cognos_8_BI_v4\multilingual) <ul style="list-style-type: none"> <li>• IBM Cognos 8 BI Intelligence Modeling</li> <li>• IBM Cognos 8 BI Server</li> </ul>		15 mins	
Create the IBM Cognos 8 Content Store	SI-21	2 mins	
Set up Web Aliases	SI-23	2 mins	
Configure IBM Cognos 8	SI-24	10 mins	
Ensure Simple File Sharing is off	SI-25	1 min	
Initialize the IBM Cognos TM1 Application	SI-26	10 mins	
Create a new Income_Statement Planning application along with the appropriate security.	SI-30	20 mins	
Install TM1 Portlets	SI-35	5 mins.	
Install TM1 Workflow	SI-38	5 mins.	

## Install JRE 6.0

### Task 1. Install JRE 6.0

1. Double-click **jre-6u16-windows-i586.exe**.
2. On the welcome screen, click **Install**.
3. After the installation is complete, click **Close**.

# Install Microsoft .NET Framework version 3.5 SP1

## Task 1. Install and configure Microsoft .NET Framework version 3.5 SP1.

If you have Microsoft .NET version 3.5 SP1 installed, skip this task.

If you do not already have Microsoft .NET Framework version 3.5 SP1 installed, you will need to download it from Microsoft.com.

1. In **Internet Explorer** browse to <http://www.microsoft.com/downloads>.
2. In the search box, type **.NET 3.5**, and then press **Enter**.
3. From the list of search results, click **.NET Framework 3.5 Service pack 1**.  
The Download Center page appears for Microsoft .NET Framework 3.5 SP1
4. Click **Download**, and then click **Run**.
5. Click **I have read and ACCEPT the terms of the License Agreement**.
6. Click **Install**.

The required files are downloaded to your machine, and the installation process begins. You may be required to close your internet browser to proceed with the installation.

7. When the install is finished, click **Close**.  
You are prompted to restart your machine.
8. Click **Restart Later**.

## Install Apache Tomcat version 6.0.18

### Task 1. Install Tomcat version 6.0.18.

1. Double-click **apache-tomcat-6.0.18.exe**.
2. Click **Next**, and then click **I Agree**.
3. Click **Next**, and then click **Next** to accept the default installation folder.
4. In the **HTTP/1.1 Connector Port** box, type **8085**.
5. Ensure that the **User Name** is **admin**, and then in the **Password** box, type **Education1!**.
6. Click **Next**, and then click **Install**.  
Apache Tomcat is installed on the machine.
7. Clear the checkboxes, and then click **Finish**.

# Ensure You Have Internet Information Services Installed

1. Open **Control Panel**, and then double-click **Add or Remove Programs**.

2. Click **Add/Remove Windows Components**.

3. Ensure the **Internet Information Services (IIS)** check box is selected.

4. Highlight **Internet Information Services (IIS)**, and then click **Details**.

5. Ensure all of the check boxes for the subcomponents are selected.

If any of the check boxes are grayed out, highlight the subcomponent, click Details, and then select all of the check boxes.

6. When you are finished, close all of the dialog boxes, except for the Control Panel.

7. Double-click **Administrative Tools**.

8. Double-click **Services** and verify two services have started:

**IIS Admin**

**World Wide Web Publishing**

9. If either service has not been started, please select it and click **Start**.

10. Right-click each service, on the **General** tab next to Startup type: click **Automatic**.

11. Close the Administrative Tools window and the Control Panel window.

## **Set Excel Macro Security**

1. Start **Excel**.
2. From the **Tools** menu, click **Macro**, and then click **Security**
3. Click **Low**, and then click **OK**.
4. Close Excel.

## Install TM1 9.5

If TM1 9.5 is installed, skip this step. You may need to extract the TM1 setup program. Please do so before proceeding.

1. Double-click **setup.exe** to run the installation program for TM1.
2. Select **English (United States)**, and then click **OK**.
3. In the warning box regarding upgrading to TM1 9.5 from previous versions, click **OK**.
4. Click **Next**.
5. Select **I accept the terms of the license agreement**, and then click **Next**.
6. Click **TM1 Contributor**, and then click **Next**.
7. Click **Standard - Complete...**, and then click **Next**.

8. On the **TCP/IP Port Configuration** page, click **Next**.
9. Enter the **Username** and **Password** for this machine or network, and then click **Next**.

If you are using a VM Image, the username is Student and the password is Education1!.

The TM1 Server will use this login information when it runs as a service. If you are unsure of this information, please ask your system administrator what account can run services on this machine.

10. Click **Next**.
11. Click **Install**.



12. Click **Finish**.
13. Restart your computer, or go to **Start\Control Panel\Administrative Tools\Services**, and manually start the **TM1 Admin**, **TM1 Excel**, and **TM1 Planning Sample** services.

## Create a Shortcut to Perspectives on the desktop

1. Right-click the desktop, point to **New**, and then click **Shortcut**.
2. In the location box, type C:\Program Files\Cognos\TM1\bin\tm1p.xla and then click **OK**.
3. Rename the newly created shortcut **Perspectives for MS Excel**.

## Setup the Advanced Rules Editor

This task will configure TM1 to use the Advanced Rules Editor rather than the default rules editor.

1. In Windows Explorer, navigate to C:\Documents and Settings\All Users\Application Data\Applix\TM1, and then double-click tm1p.ini to open the file in Notepad.
2. Scroll to the bottom, find the **AdvancedRulesEditor** setting, and then change the **F** to **T**.
3. From the **File** menu, click **Save**, and then close Notepad.

Note: This changes the setting from False to True, which enables the Advanced Rules Editor in Architect only. The Advanced Rules Editor in Excel will NOT be enabled.

## Change TM1 password for Planning Sample server

In order to make both the greatoutdoors server and the planning sample server available in TM1 Contributor, both admin passwords must be the same.

1. From the **Start** menu, navigate to **All Programs\IBM Cognos\TM1\Architect**.
- TM1 Architect: Server Explorer appears.
2. In the left pane, expand **TM1**, and then double-click **planning sample**.
3. In the **Username** box, type **admin**, in the **Password** box, type **apple**, and then click **OK**.

The tree on the left expands to show the contents of the server. You are now logged on to this server.

4. Right-click **planning sample**, point to Security, and then click **Change Password**.
5. Ensure the **Password** box is blank (delete any contents), and then click **OK**.
6. In the **Password Verify** box, ensure the box is blank, and then click **OK**.
7. Close **TM1 Architect**.

## Set up TM1 servers.

1. From the **Course Materials Vault**, download the **P6502\_INST\_Data.exe** Instructor Download file.
2. Run the **.exe** and extract the files to **C:\Edcognos\P6502**.
3. Navigate to **C:\Edcognos\P6502\GreatOutdoors\DataFiles** and then copy the **GreatOutdoors** shortcut to your desktop.
4. Navigate to **C:\Edcognos\P6500\Income\_Statement\DataFiles** and then copy the **Income\_Statement** shortcut to your desktop.
5. Close **Windows Explorer**.
6. On the desktop, double-click the shortcut to start the **GreatOutdoors** server.
7. On the desktop, double-click the shortcut to start the **Income\_Statement** server.

The server window remains open while the servers are loading. Once the servers are running, the windows minimize and remain on your taskbar.

## Deploy TM1 Contributor to the Tomcat Server

After installing TM1 Contributor, you need to manually deploy the web application.

1. Go to **Start\Control Panel\Administrative Tools\Services**, and stop the **Apache Tomcat** service if it is started.
2. In **Windows Explorer**, browse to **C:\Program Files\Cognos\TM1\Cadmin**.
3. Copy the **pmpsvc.war** file to **C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps**.
4. Browse back to **C:\Program Files\Cognos\TM1\Cadmin**.
5. Copy the **xercesImpl.jar** file to **C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib**.
6. Go to **Start\Control Panel\Administrative Tools\Services** and set the Apache Tomcat service to start **Automatically**.
7. Start the **Apache Tomcat** service.
8. Start the following services if they are not already started:
  - **TM1 Admin Server**
  - **TM1 Excel Service**
  - **TM1 Server – Planning Sample**
9. Change the properties on the above services so they will start Automatically.

## Test TM1 Contributor

1. Open **Internet Explorer**, and then go to <http://localhost:8085/pmpsvc>.  
It may take a moment for the web site to appear.  
If the IBM Cognos Planning Configuration page appears, then everything is working properly, and you can close Internet Explorer.  
If the IBM Cognos Planning Configuration page does not appear, then follow the steps below.
2. Go to **Start\Control Panel\Administrative Tools\Services**, and stop the **Apache Tomcat** service.
3. Open **Windows Explorer**, and navigate to **C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf**.
4. Open the **tomcat-users.xml** file in **Notepad**.
5. At the end of the first line, change '**cp1252**' to '**UTF-8**' so it reads:  
**<?xml version='1.0' encoding='UTF-8'?>**
6. Save and close the file.
7. Restart the **Tomcat** service, and then repeat step 1.  
The IBM Cognos Planning Configuration page should now appear.
8. Close Internet Explorer.

## Adjust Internet Explorer Security

1. Open **Internet Explorer**, and from the menu click **Tools**, and then click **Internet Options**.
2. Click the **Security** tab, and then click **Custom Level**.
3. In the **Settings** box, under **Downloads**, change **Automatic prompting for file downloads** to **Enable**.
4. Click **OK**, and then click **OK**.
5. Close **Internet Explorer**.

## Create Users in the NTLM authentication provider

1. From the **Start** menu, click **Control Panel**, and then double-click **Administrative Tools**.
2. Double-click **Computer Management**.
3. Expand **Local Users and Groups**, and then click **Users**.

You can see the current list of users. You can now run a script to add users and their information to the list.

4. In **Windows Explorer**, navigate to **C:\Edcognos\<course code>\Instructor Files**, and then double-click **Create NTLM Users.vbs**.

The new users are added to the list.

**Note:** If you are not able to create the users by running the script, create them manually by following the steps below:

5. Right-click the **Users** folder, and click **New User**.
6. Enter the user information as follows:

User Name: **C8User**

Full Name: **C8 User**

Password: **Education1!**

Confirm Password: **Education1!**

Ensure the **Password never expires** box is selected.

7. Click **Create**, and then click **Close**.

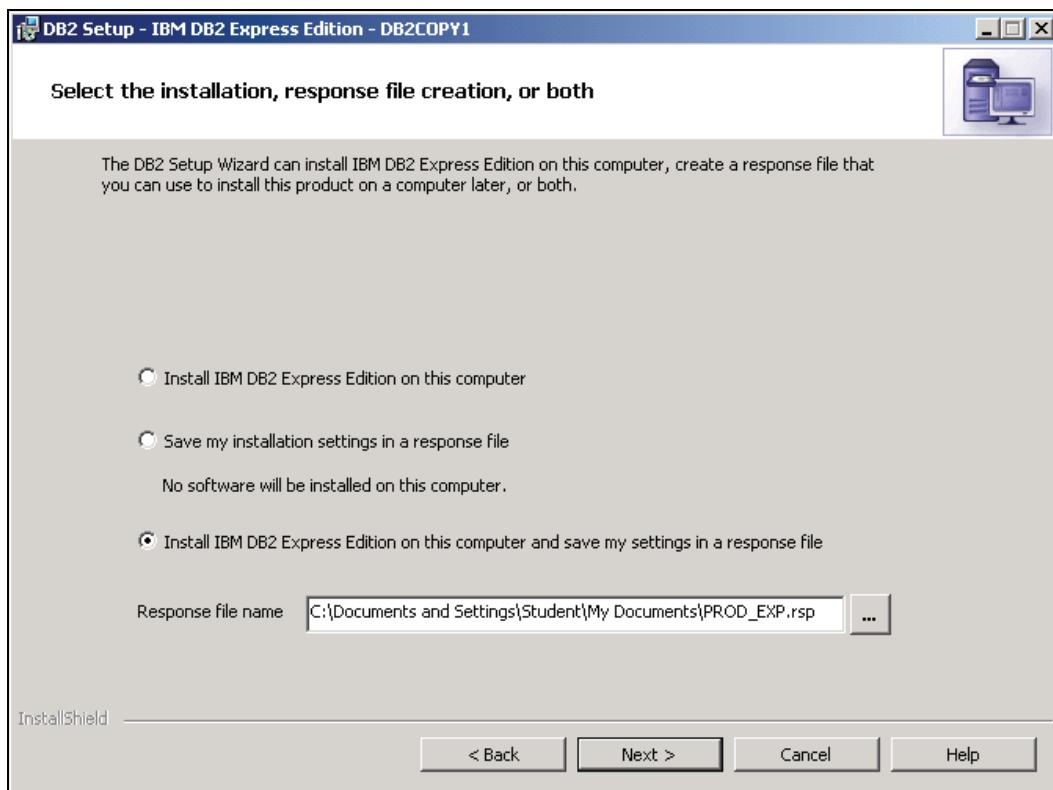
The new user is added.

# Install DB2 Express Version 9.5

You must be logged on to the local machine with Administrative privileges to perform the installation.

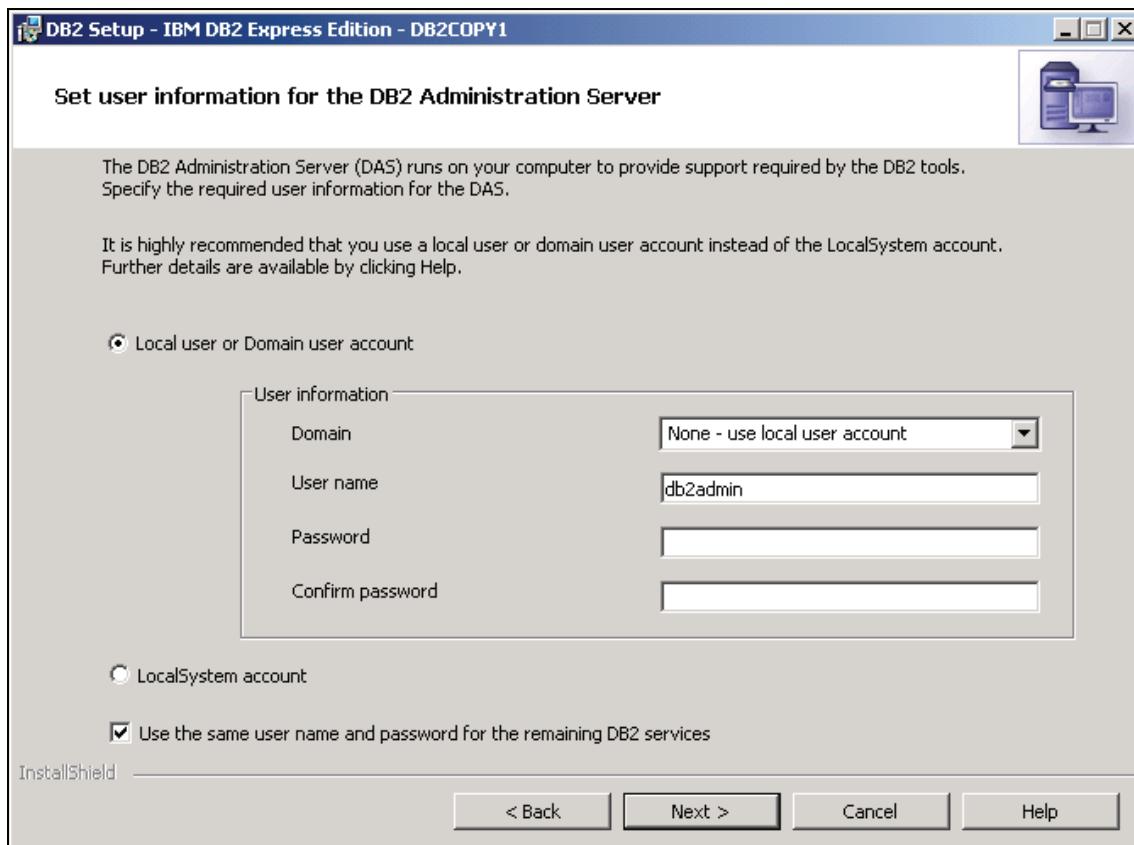
1. In **Windows Explorer**, navigate to **C:\Edcognos\<course code>\Instructor Files**, and then double-click **DB2\_Exp\_Ed\_V95\_Win\_x86.exe**.
2. Click **Browse**, navigate to **C:\Edcognos\<course code>\Instructor Files**, and then click **Unzip**.
3. Click **OK**, and then click **Close**.
4. In **Windows Explorer**, navigate to **C:\Edcognos\<course code>\Instructor Files**, and then double-click **setup.exe**.
5. Click **Install a Product**, and then click **Install New**.
6. Click **Next**.
7. Click **I accept the terms in the license agreement**, and then click **Next**.
8. Ensure **Typical** (default) is selected, and then click **Next**.

The results appear as follows:



9. Click **Install IBM DB2 Express Edition on this computer**, and then click **Next**.
10. Leave the default directory, and then click **Next**.

The results appear as follows:



11. In the **Password** and **Confirm password** boxes, type **Education1!**, and then click **Next**.
12. Leave **DB2** in the **DB2 instances** box, and then click **Next**.
13. Click **Install**.
14. Click **Next**, click **Finish**, and then click **Exit**.

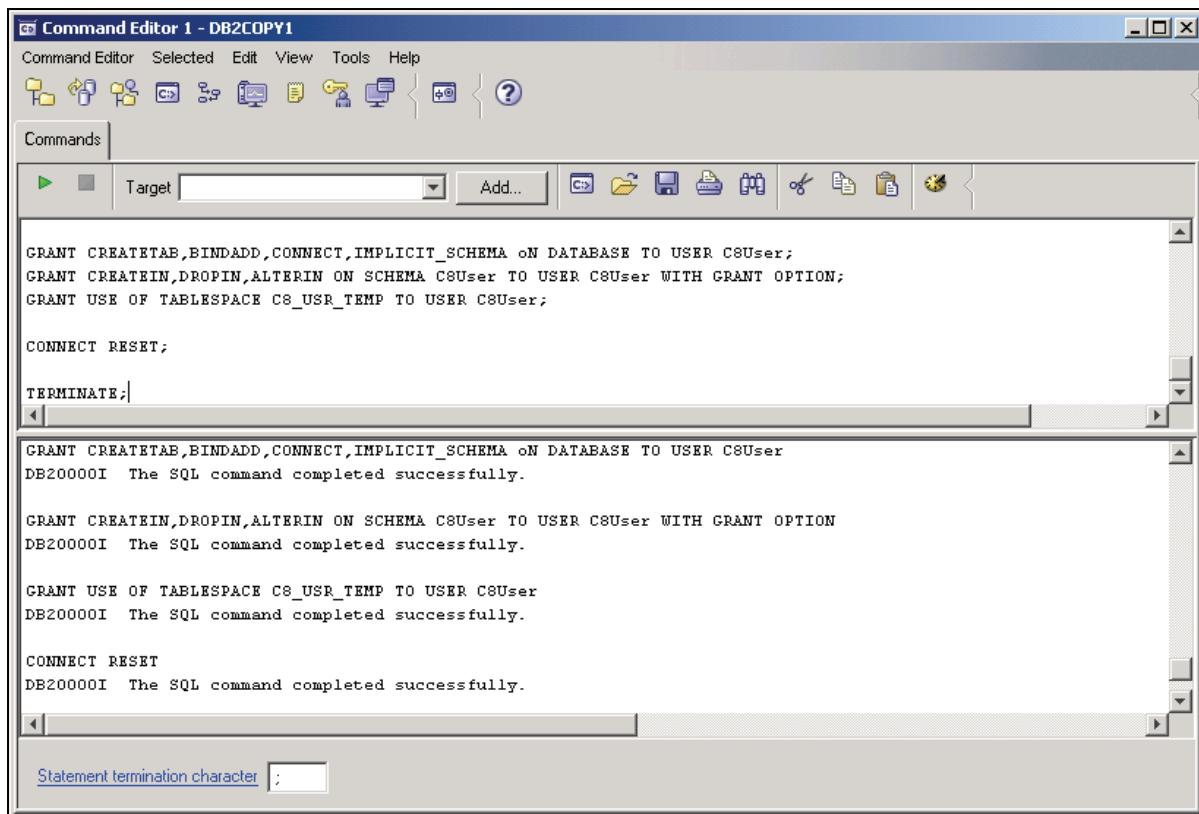
# Create the IBM Cognos 8 BI Content Store database

## Task 1. Create the IBM Cognos 8 BI Content Store database.

You must be logged on to the local machine with administrative privileges to perform this task.

1. In Windows Explorer, navigate to **C:\Edcognos\<course code>\Instructor Files**, and then double-click **DB2 Script - Create & Config C8 Content Store.txt**, copy the contents to the clipboard, and then close Notepad.
2. From the Start menu, point to **All Programs\IBM DB2\DB2COPY1 (Default)\Command Line Tools**, and then click **Command Editor**.
3. At the prompt, paste the contents of the clipboard, and then from the Selected menu, click **Execute**.

The results appear as follows:



The screenshot shows the Command Editor 1 - DB2COPY1 window. The interface includes a toolbar with various icons, a menu bar with 'Command Editor', 'Selected', 'Edit', 'View', 'Tools', and 'Help', and a toolbar below it with icons for opening files, saving, and executing commands. The main area is divided into two panes: 'Commands' on the left and a large text pane on the right. The text pane contains the following SQL commands and their execution results:

```

GRANT CREATETAB,BINDADD,CONNECT,IMPLICIT_SCHEMA ON DATABASE TO USER C8User;
GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA C8User TO USER C8User WITH GRANT OPTION;
GRANT USE OF TABLESPACE C8_USR_TEMP TO USER C8User;

CONNECT RESET;

TERMINATE;|
```

DB20000I The SQL command completed successfully.

```

GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA C8User TO USER C8User WITH GRANT OPTION
DB20000I The SQL command completed successfully.

GRANT USE OF TABLESPACE C8_USR_TEMP TO USER C8User
DB20000I The SQL command completed successfully.

CONNECT RESET
DB20000I The SQL command completed successfully.
```

Statement termination character : |

4. Close **Command Editor**.
5. In **Windows Explorer**, navigate to **C:\Program Files\IBM\SQLLIB\java**, and then copy the **db2jcc.jar** and **db2jcc\_license\_cu.jar** files.
6. Navigate to **C:\Program Files\cognos\c8\webapps\p2pd\WEB-INF\lib**, and then paste the **db2jcc.jar** and **db2jcc\_license\_cu.jar** files.

## Set up Web Aliases

### Task 1. Set up alias for IBM Cognos 8 BI.

1. From Control Panel, open Administrative Tools, and then open Internet Information Services.
2. In the left pane, expand <servername>, and then expand Web Sites.
3. Expand Default Web Site, right-click Default Web Site, point to New, and then click Virtual Directory.
4. Click Next.
5. Under Alias, type cognos8, and then click Next.
6. Browse to <IBM Cognos 8 install location>\Program Files\cognos\c8\webcontent, click OK, and then click Next.
7. Clear the Run scripts check box, so only Read is selected, and then click Next.
8. Click Finish.

### Task 2. Set up alias for cgi-bin.

1. Right-click the cognos8 virtual directory folder, point to New, and then click Virtual Directory.
2. Click Next.
3. Under Alias, type cgi-bin, and then click Next.
4. Browse to <IBM Cognos 8 install location>\Program Files\cognos\c8\cgi-bin, click OK, and then click Next.
5. Select the Execute check box, deselect the Read and Run scripts check boxes, and then click Next.
6. Click Finish.

## Configure IBM Cognos 8

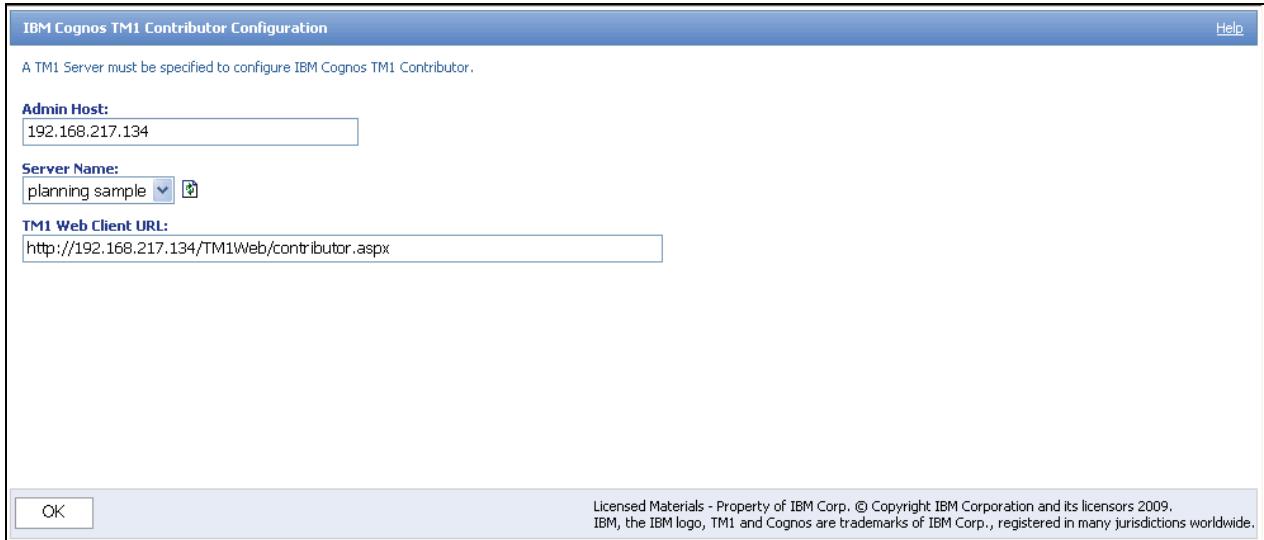
1. From the **Start** menu, navigate to **All Programs\IBM Cognos 8\IBM Cognos Configuration**.
2. In the **Explorer** pane, under **Content Manager**, click **Content Store**.
3. In the **Resource Properties** pane on the right, set the following properties:
  - Database server with port number or instance name: **localhost:50000**
  - Database name: **cm**
  - User ID and password:
    - User ID: **C8User**
    - Password: **Education1!**
4. Right-click **Content Store**, and then click **Test**.
5. Click **Close** when the test is complete.
6. Click **Save configuration**  to save the new configuration.
7. Click **Close** to close the **IBM Cognos Configuration** dialog box after the configuration is saved and all items have green check marks beside them.
8. Click **Start**  to start the IBM Cognos 8 service.  
You will receive a warning about the mail server message. It can be ignored.
9. When all services are started, click **Close**.
10. Close **IBM Cognos Configuration**.

## Ensure Simple File Sharing is Off (for VMWare images only)

1. From the **Start** menu, navigate to **Control Panel\Folder Options**.
2. Click the **View** tab, and then ensure the **Use Simple File Sharing (Recommended)** check box is cleared.
3. Click **OK**.
4. Close the **Control Panel**.

## Initialize the IBM Cognos TM1 Application

1. Open Microsoft Windows Explorer.
2. In the address bar, type <http://localhost:8085/pmpsvc>.
3. On the **IBM Cognos TM1 Contributor Configuration** dialog, click **OK**.  
Note that you must have at least one server configured for IBM Cognos TM1 Contributor.

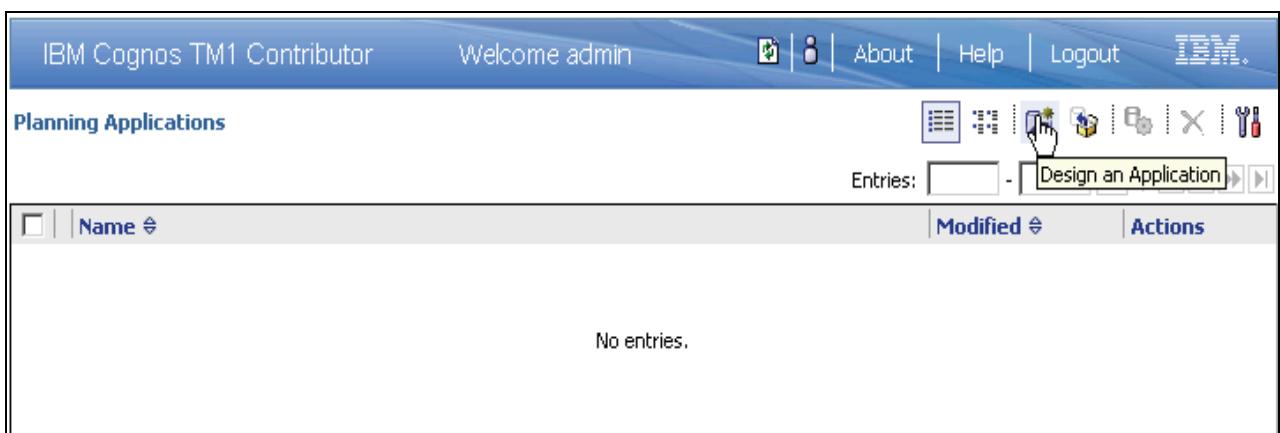


4. Log on as follows:

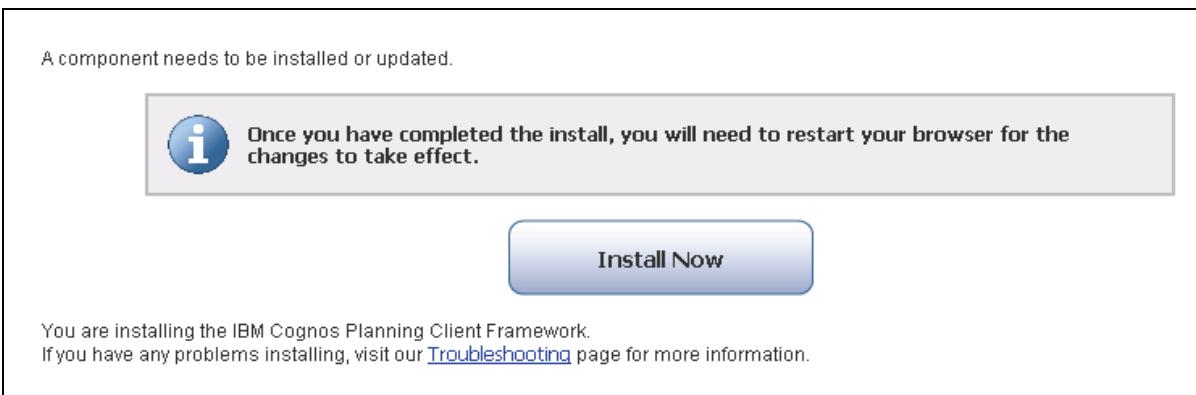
**UserName:** admin

**Password:** a password is not needed.

5. Click **Login**.
6. On the **IBM Cognos TM1 Contributor, Planning Applications** portal select **Design an Application**.



7. Click **Install Now**. As this is the first time you are accessing the TM1 Contributor Administrator application you must first install it. You will be presented with the following dialog.



8. In the **File Download - Security Warning** dialog click **Run**.
9. In the **Internet Explorer - Security Warning** dialog click **Run**.
10. When the installation is complete, close the browser.
11. On the Desktop, double-click **Income\_Statement**.  
This is the TM1 Server that we will be using in TM1 Contributor.
12. Launch the browser and enter the address  
**http://localhost:8085/pmpsvc**

13. Login as **admin/no password** and then click **Login**.

**Login**  
Please enter your information.

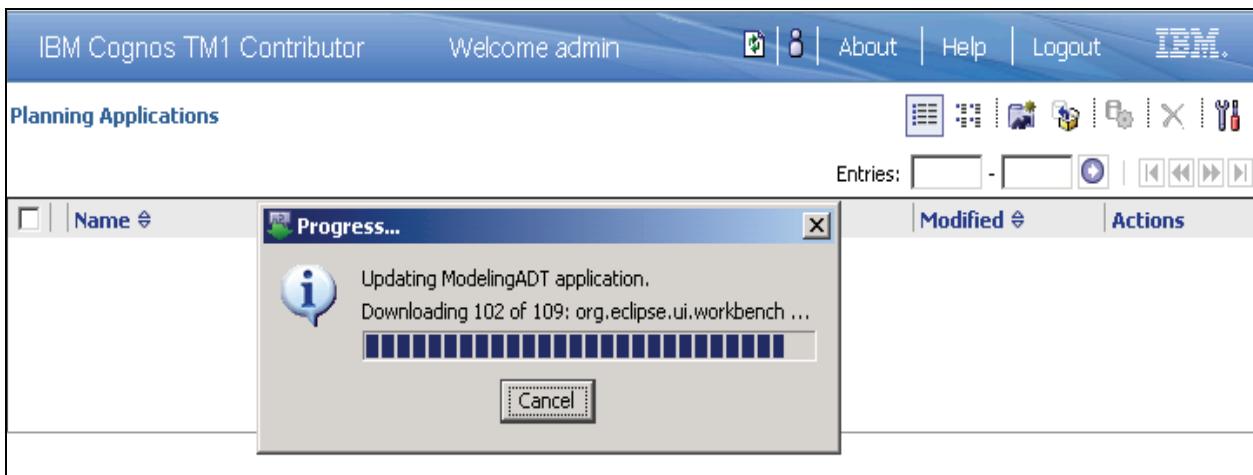
User name:

Password:

Please note that after some time of inactivity, the system will log you out automatically and ask you to sign in again.

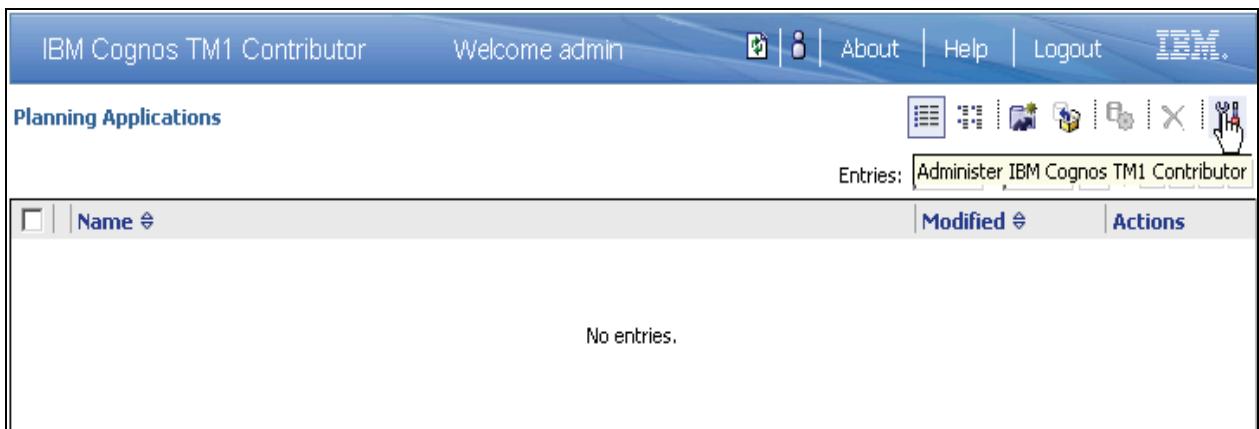
Licensed Material - Property of IBM Corp. © Copyright IBM Corporation and its licensors 2009.  
IBM, the IBM logo, TM1 and Cognos are trademarks of IBM Corp., registered in many jurisdictions worldwide.

14. In the **IBM Cognos TM1 Contributor Application** portal select **Design an Application**. Again as this is the first time you are designing an application the process will update. Note that every time you launch the IBM Cognos TM1 Contributor Administration application the process will check for updates.



You will be in the IBM Cognos TM1 Contributor Administration application.

15. Return to IBM Cognos TM1 Contributor and select **Administer IBM Cognos TM1 Contributor**.



16. In the **IBM Cognos TM1 Contributor Configuration** dialog, under **Server Names:** click **Add**.



17. In the **Add Server** dialog, under **Admin Host:** type **localhost**.
18. Select the **Refresh** button and from the **Server Name** drop down select **income\_statement** and then click **OK**.  
Notice that the localhost you entered has been converted to the IP address.
19. Click **OK** to close the **IBM Cognos TM1 Contributor Configuration** dialog.
20. Select **Logout** on the IBM Cognos Planning dialog.

## Create a new Income\_Statement planning application along with the appropriate security.

1. Open Internet Explorer, and then go to <http://localhost:8085/pmpsvc>.
2. Log on as follows:
 

User ID: **admin**

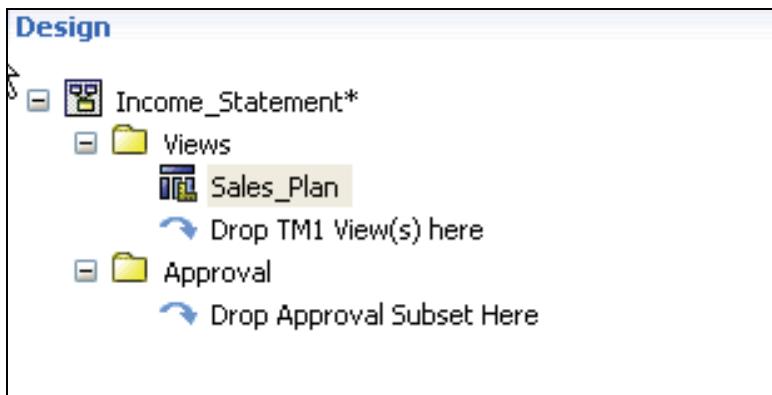
Password: <blank>
3. In the **IBM Cognos TM1 Contributor** portal, on the menu bar, click **Design an Application**.  Expand this window.
4. Within the **IBM Cognos TM1 Contributor Administration** window, select **File -> New**.
5. In the **Create new Planning Application** dialog, for the **Application Name:**, enter **Income\_Statement**.
6. For TM1 Server Name: **select IP Address:income\_statement**.

Your results should look as follows:



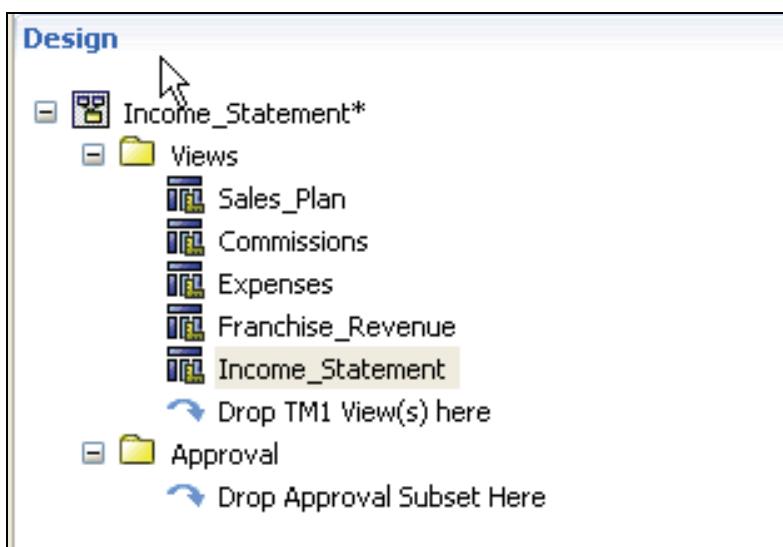
7. Click **OK**.
8. In the bottom left had side of this window, under **TM1 Objects**, expand the **Cubes** folder.
9. Expand **Sales\_Plan**.
10. Expand **Views**.

- Click and drag the **Sales\_Plan** view to the top left hand side of this window, under **Design**, and under **Views** and place the Sales\_Plan view on top of **Drop TM1 View(s) here**. Your results should look as follows:



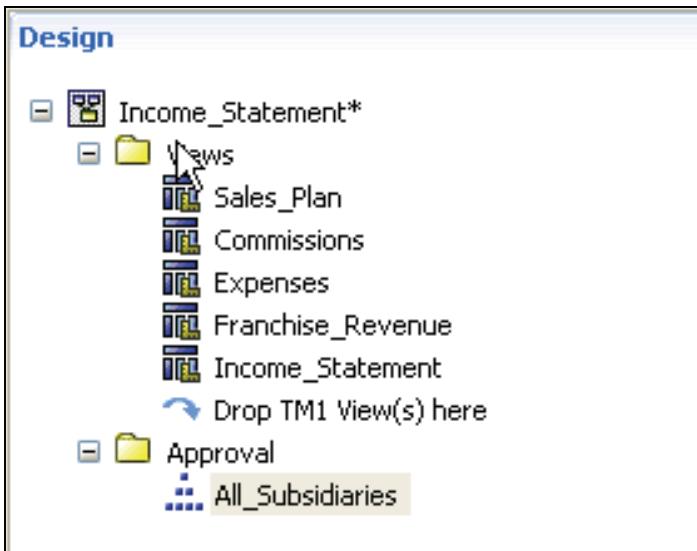
- Repeat steps 9-11 for **Commissions**, **Expenses**, **Franchise\_Revenue** and **Income\_Statement**.

Your results should look as follows:



- Collapse the **Cubes** folder in the bottom pane.
- Expand the **Dimensions** folder.
- Expand **Subsidiaries** and expand **Subsets**.
- Click and drag **All\_Subsidiaries** to the top left hand side of this window, under **Design**, and under the **Approval** folder and place on top of **Drop Approval Subset here**.

17. Click **Yes** to the warning message. Your results should look as follows:



18. On the **File** menu, click **SaveAs**.
19. Navigate to **C:\Edcognos\P6500\Income\_Statement** and name the file **Income\_Statement**. Click **Save**.
20. On the toolbar, click **Validate**
21. In the **Validation Successful** box, click **OK**.
22. On the toolbar, click **Deploy**
- If you get a message stating that the approval hierarchy is being used by another application, click **OK** and continue with step 23. If you do not get this message, skip to step 32.
23. Click **Start, All Programs, IBM Cognos, TM1, Architect**.
24. Expand **TM1**. Double click the **Income\_Statement** server.
25. In **UserName**: enter **admin**. Click **OK**.
26. Select the **View** menu and ensure the **Display Control Objects** is selected.
27. Expand **Processes**.
28. Right click the **{tp\_admin\_delete\_all** process and select **Run**.
29. In the **Parameters** box, click **OK**.

30. Click **OK** in the confirmation dialog, click **View**, click **Display Control Objects** and then close **TM1 Architect**.
31. On the toolbar, click **Deploy**.
32. On the successful deployment confirmation message, click **OK**.
33. On the **IBM Cognos TM1 Contributor Administration** window, under **4 Define Security**, click **Define**.  
The Manage Rights - Income\_Statement window opens.
34. In the **1 Select Node Box** on bottom left of the window, expand all Nodes.
35. Click **GO Americas**.
36. Under **2 Select Group** click **Contribute-Americas**.
37. Under **3 Define Security**, under **Right:** select **Submit**. Leave **Review Depth** and **View Depth** as **All**. Click **Add**.
38. Repeat steps 35-37 for all of the other child nodes:
  - GO Asia Pac, GO Accessories GmbH, GO Central Europe, GO Northern Europe, GO Southern Europe
  - Map to the corresponding Contribute... Group under number 2.
39. Under **1 Select Node Box**, click **GO AMERICAS REGION**, under **2 Select Group** click **Review-Americas** and under **3 Define Security**, under **Rights**, select **Review**. Click **Add**.
40. Repeat step 39 for all the other parent nodes except **TOTAL COMPANY**.

41. Under **1 Select Node Box**, click **TOTAL COMPANY** under **2 Select Group** click **View\_Company** and under **3 Define Security**, under **Rights**, select **View**. Click Add.

The results are as follows:

IBM Cognos TM1 Contributor				
				<input checked="" type="checkbox"/> Allow Reviewer Edit
Node	Group	Right	Review Depth	View Depth
TOTAL COMPANY	View_Company	View	None	All
GO AMERICAS REGION	Review_Americas	Review	All	All
GO Americas	Contribute_Americas	Submit	All	All
GO ASIA PACIFIC REGION	Review_Asia_Pacific	Review	All	All
GO Asia Pacific	Contribute_Asia_Pacific	Submit	All	All
GO EUROPE GMBH	Review_Europe	Review	All	All
GO Accessories GmbH	Contribute_Accessories_GmbH	Submit	All	All
GO Central Europe	Contribute_Central_Europe	Submit	All	All
GO Northern Europe	Contribute_Northern_Europe	Submit	All	All
GO Southern Europe	Contribute_Southern_Europe	Submit	All	All

42. Click **Save** and then click **Close** on the **Save Rights** confirmation window.

43. Click **Return** .

44. Click **Activate Application - Income\_Statement** .

45. Click **Logout**.

46. Log in using **Gradin** as the user id.

Bengt Gradin is a contributor in Sweden thus when you log in as Gradin, you should only see the Northern Europe.

47. Click on the **Income\_Statement** application. Notice you only have access to the GO Northern Europe region.



You are a reviewer or contributor for:		
Name	State	Ownership
<a href="#">GO Northern Europe</a>	 Not Started	 None

48. Click **Logout**.

49. Close the Windows Explorer window.

50. Shutdown the **Income\_Statement** server. Save changes when prompted.

## Install TM1 portlets

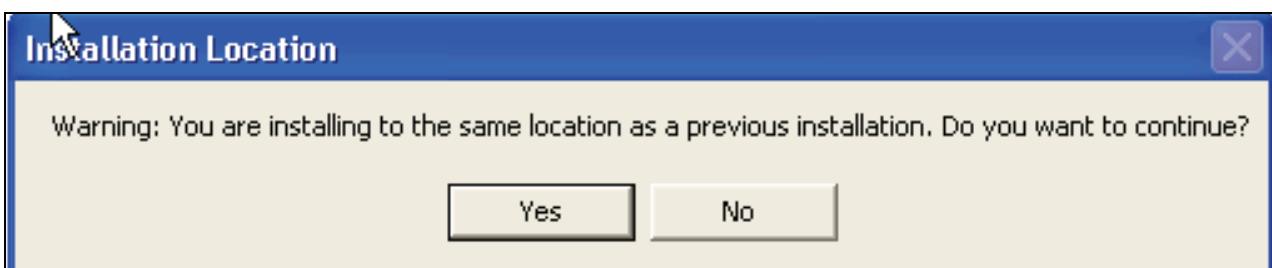
You may need to extract the TM1 portlet setup program, **C8\_TM1\_Portlets\_win32\_8.4.28.4.tar.gz** to a temporary location. Please do so before proceeding.

1. Double-click **issetup.exe**.
2. Select the language to install and then click **Next**.

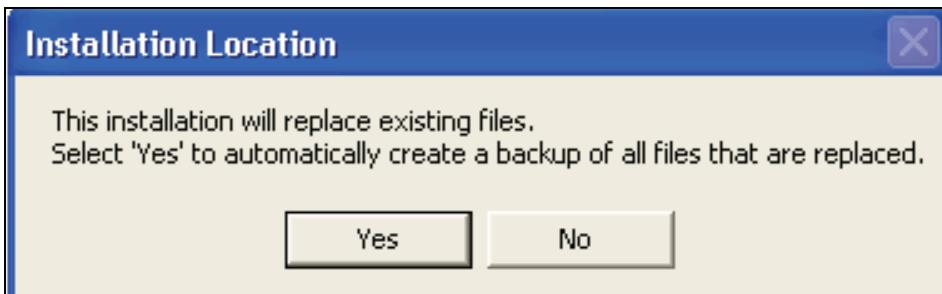


Regardless of the language selected, they will all be installed

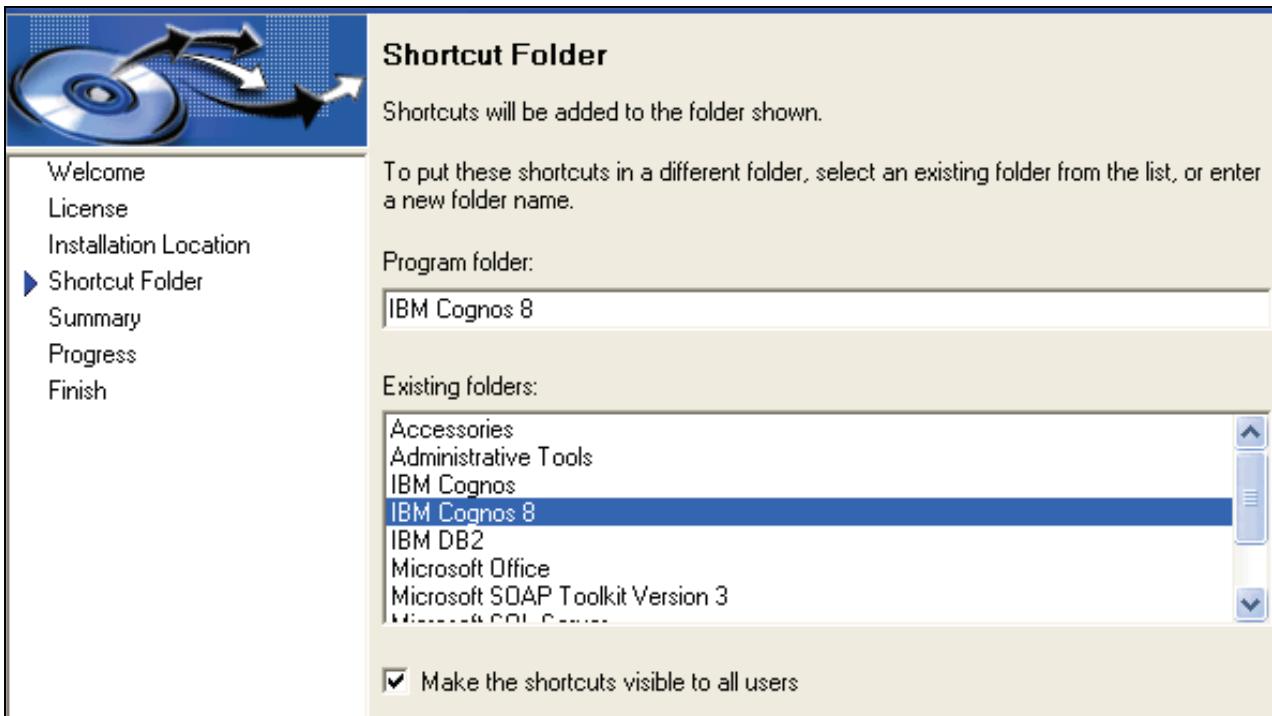
3. Under the License Agreement, click **I accept** and then click **Next**.
4. Under Installation Location, keep the default location and then click **Next**.
5. Click **Yes** to continue.



6. Click **Yes** to create backup files.

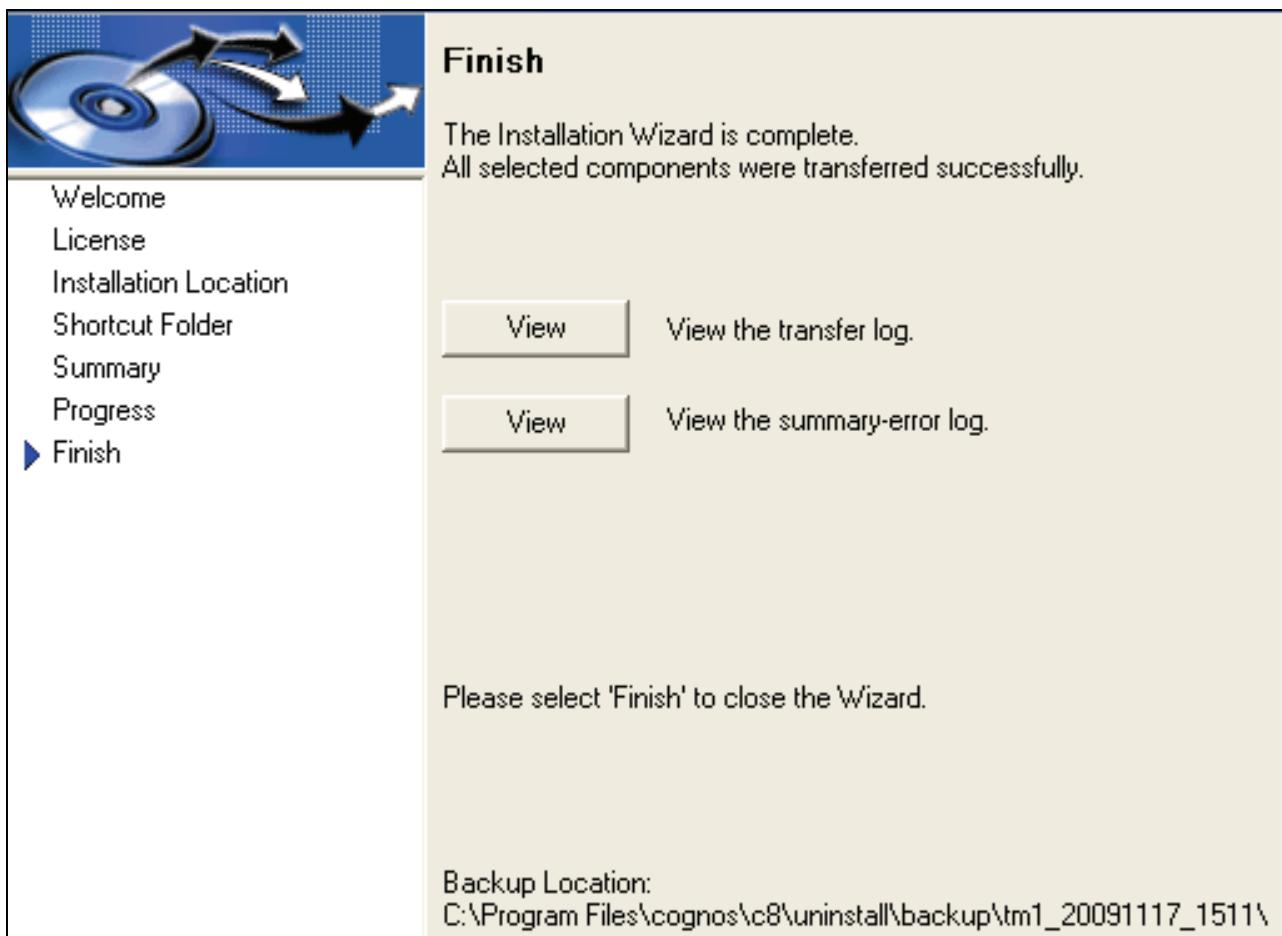


7. Click Next to create a Shortcut.



8. Under the Installation Summary, click **Next**.

9. Click **Finish**.

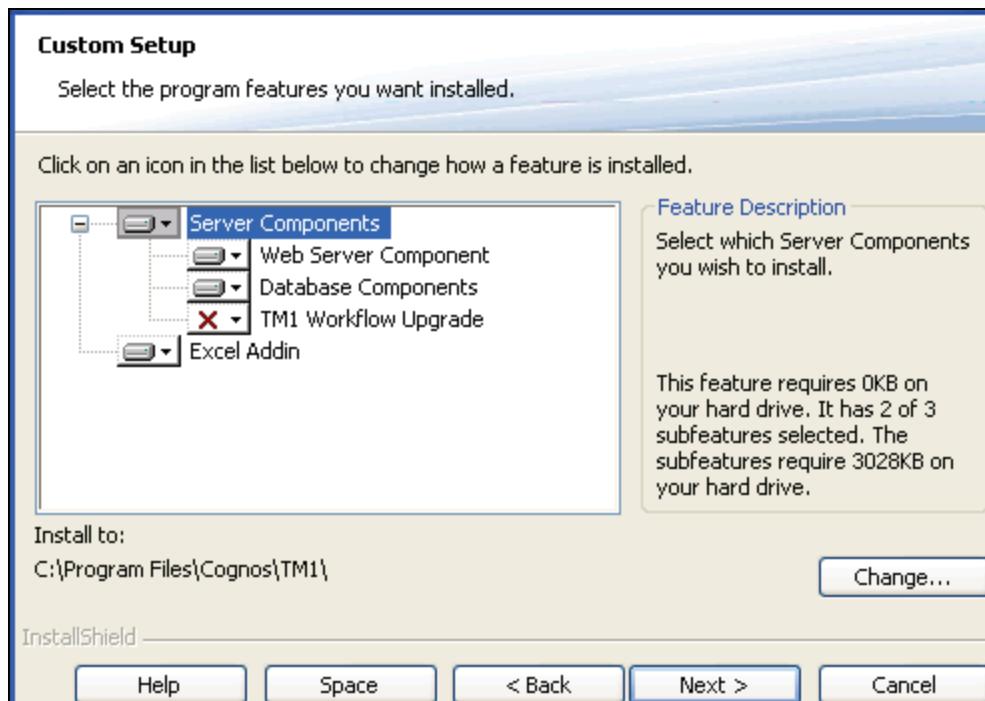


## Install TM1 Workflow

If TM1 Workflow is already installed, you are finished. You may need to extract the TM1 Workflow setup program to a temporary location. Please do so before proceeding.

1. Double-click **setup.exe** in the temporary location.
2. In the **Welcome** screen, click **Next**.
3. Click **I accept the terms in the license agreement** and then click **Next**.
4. Under the **Custom Setup**, click **Next** to accept the defaults.

The files will be copied to a default location in the TM1 install directory, C:\Program Files\Cognos\TM1\Custom\TM1Data\TM1 Planning Manager V2\. You could also choose to install the files directly to the greatoutdoors server. However, if you wanted to set up workflow on another server you would need to reinstall Workflow.



5. Under **Excel Addin**, leave the defaults and then click **Next**.
6. Click **Install**.
7. Click **Finish**.

**Information Management**



# Introduction

IBM Cognos TM1 (v9.5)



**Cognos.**  
software

© 2009 IBM Corporation



## Course Objectives

- At the end of this course, you should be able to:
  - examine the components of a TM1 model
  - create dimensions
  - build cubes and views
  - load and maintain data
  - add business rules
  - optimize business rules
  - transfer data into the model using scripts
  - customize drill paths
  - use rules for advanced modeling

**Cognos.**  
software



IBM Cognos TM1: Design and Develop Models (v9.5) is a five-day, instructor-led course designed to teach modelers how to build a complete model in TM1. Through a series of lectures and hands-on exercises, students will learn how to set up and verify structures, manually enter data into these structures, and define the data users can see.

### Audience

This course is intended for modelers.

### Recommended prerequisites

- Successfully completed the IBM Cognos TM1: Analyze Data (v9.5) course
- Basic knowledge of OLAP and TM1
- Significant experience with Excel spreadsheets (functions, macros, etc.)
- Understanding of the metrics and drivers of your business

## Course Objectives (cont'd)

- At the end of this course, you should be able to:
  - convert currencies
  - model for different fiscal requirements
  - analyze data in TM1 web
  - contribute data to managed planning applications
  - create and deploy managed planning applications
  - define workflow (optional)
  - integrate TM1 with IBM Cognos 8 BI and IBM Cognos Planning (Optional)
  - present data and reports (Optional)

Cognos.  
software





## **Introduction to TM1**

### **Create Dimensions**

### **Build Cubes and Views**

### **Load and Maintain Data**

### **Add Business Rules**

### **Optimize Rule Performance**

## **Transfer Data Into Your Model Using Scripts**

IBM Cognos TM1 (v9.5)

**Cognos**.  
software

Information Management

© 2009 IBM Corporation



**Customize Drill Paths**  
**Using Rules for Advanced Modeling**  
**Convert Currencies**  
**Model for Different Fiscal Requirements**  
**Introduction to Managed Planning Applications**  
**Contribute Data to Managed Planning Applications**

IBM Cognos TM1 (v9.5)

**Cognos.**  
software

Information Management

© 2009 IBM Corporation

## Create and Deploy Managed Planning Applications to the Web



### Appendix A: Define Workflow

### Appendix B: Integrate with C8 and Cognos Planning

### Appendix C: Design for Reporting

### Appendix D: Optimize and Tune TM1 Models

### Appendix E: Identify Common Data Structures

IBM Cognos TM1 (v9.5)

Cognos.  
software

Information Management

© 2009 IBM Corporation

## Additional Training Resources

Bookmark [www.ibm.com/cognos/training](http://www.ibm.com/cognos/training) for details on:

- Instructor-led training in a classroom or online
- Self-paced training that fits your needs & schedule
- Comprehensive curricula & training paths that help you identify the courses that are right for you
- The IBM Cognos Certification program
- Other resources that will enhance your success with IBM Cognos software

**Cognos.**  
software



**Information Management**



# **Introduction to IBM Cognos TM1**

IBM Cognos TM1 9.5



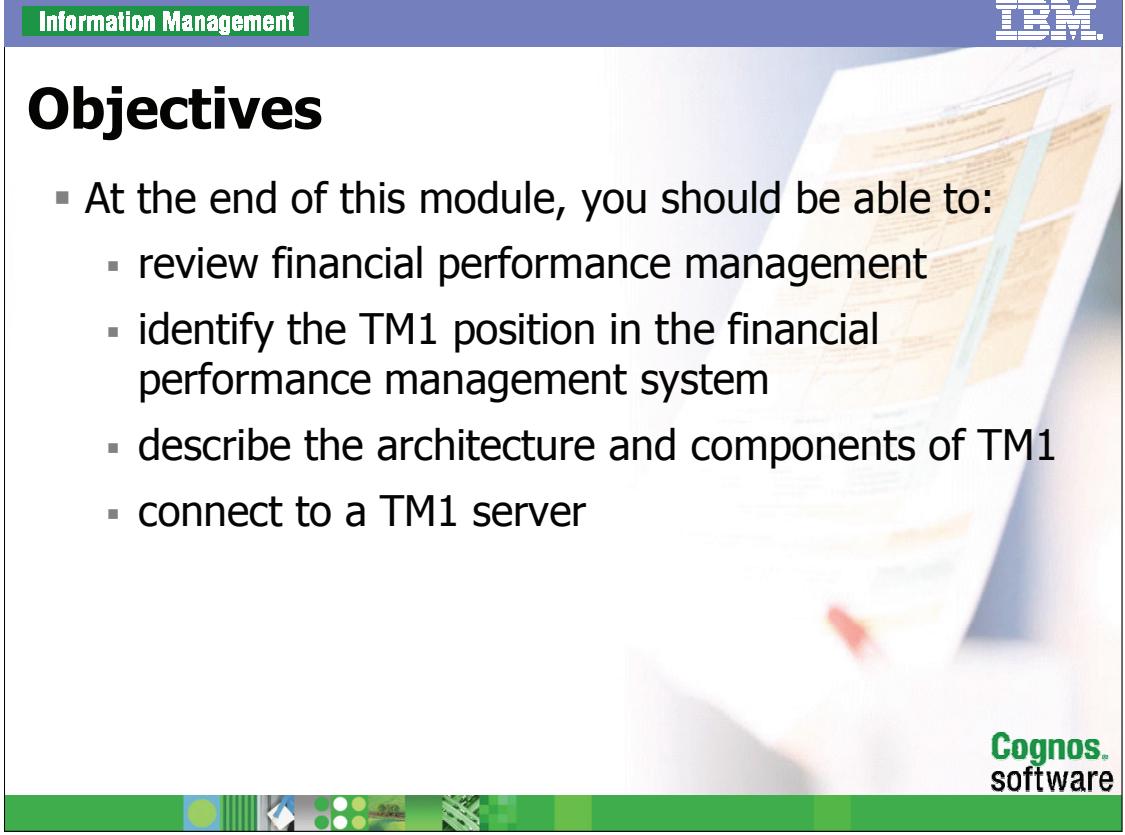
**Cognos.**  
software

© 2009 IBM Corporation



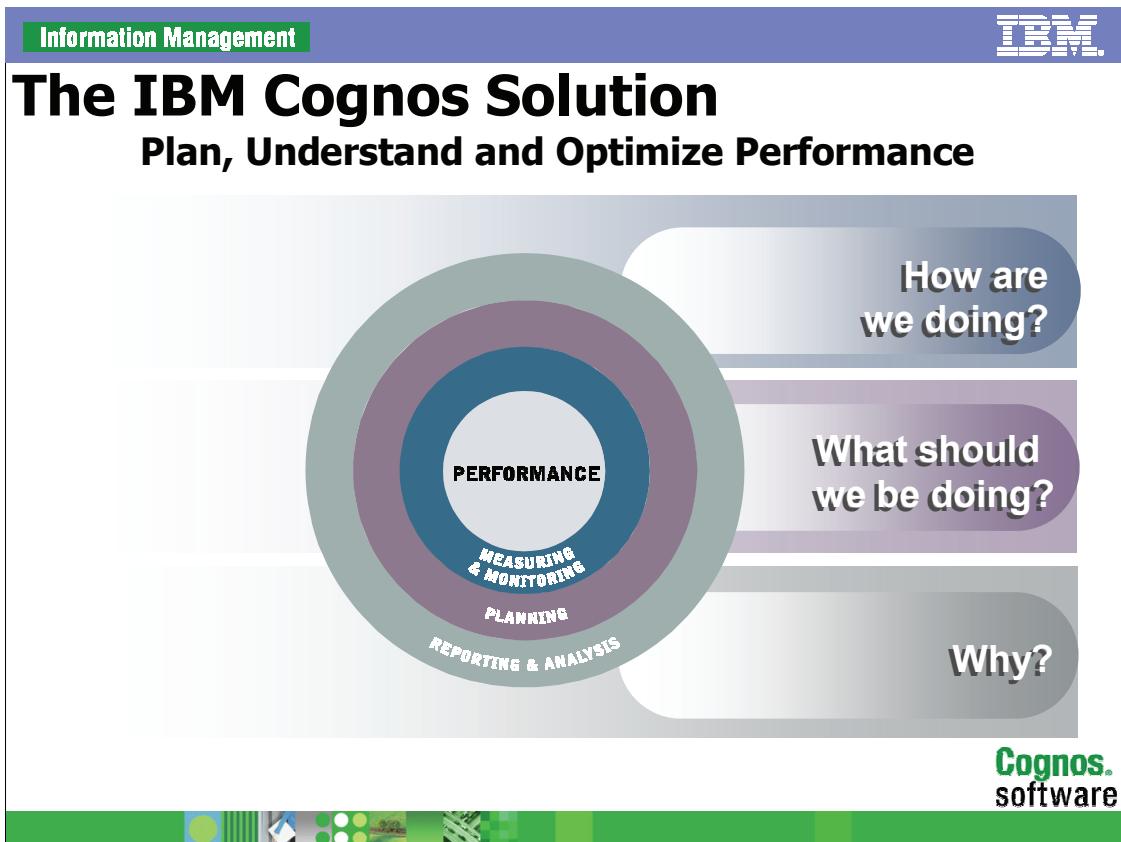
# Objectives

- At the end of this module, you should be able to:
  - review financial performance management
  - identify the TM1 position in the financial performance management system
  - describe the architecture and components of TM1
  - connect to a TM1 server



Cognos  
software





The Cognos performance management solution consists of:

1. **Enterprise Planning** - an integrated solution for planning, budgeting, forecasting, modeling, consolidation, and financial reporting
2. **Enterprise Business Intelligence** - track operational performance against your strategic plans and monitor the key metrics of your day-to-day operation, and report and analyze your data, to understand the *why* behind your business performance

### Instructor Notes

#### 1) Enterprise Planning

Use Cognos Planning or TM1 for planning, budgeting, forecasting, and modeling. Use Cognos Controller for consolidation and financial reporting.

#### 2) Enterprise Business Intelligence

Use Cognos 8 Metric Studio to track operational performance against strategic plans and monitor the key metrics. Use Cognos 8 Query Studio, Report Studio, Analysis Studio, and Event Studio to report and analyze your data.

# Financial Performance Management

- The IBM Cognos vision for financial performance management focuses on the relationship between three critical processes in the Office of Finance:
  - close, consolidate, and report
  - plan, forecast, and control
  - analyze and optimize



**Cognos.**  
software

Finance organizations around the world face the same challenges. They struggle to drive growth and performance while simultaneously ensuring compliance with a myriad of legal, tax, and accounting requirements. Financial performance management solutions from IBM Cognos give Finance the tools to win that struggle.

**INTERACTION - Whiteboard:** Put up a whiteboard and ask participants to choose a color and list some of the goals of their TM1 applications. Planning, forecasting, reporting, finance, operations, etc.

## TM1 OLAP Solution

- TM1 is an OLAP tool with:
  - Powerful database engine (TM1 Server)
  - Familiar user interface:
    - Excel add-on
    - Web browser

**Cognos.**  
software



Remote TM1 Servers provide access to shared data and objects in your organization.

You can have more than one data server running.

### Instructor Notes

A single Admin Server can control all the TM1 Data Servers that are running, OR, you can use multiple Admin servers. This will still allow all servers to be accessed by those clients who list multiple admin servers separated by commas in their Admin Host field. This also allows specific Admin servers to be aligned with specific logical TM1 servers aligned by function: For example; Production, Development, HR or Finance.

**INTERACTION - Survey:** Create a survey asking if participants are developing models and deploying them in via Excel, the web, or a combination

## Use Different TM1 User Interfaces

- Depending on your needs, you can use a different user interface:
  - TM1 Perspectives, uses Excel add-on
  - TM1 Architect
  - TM1 Web
  - TM1 Contributor
  - TM1 Contributor Administration



Cognos  
software

TM1 Perspectives is a client application that lets you create, manage, access, and edit TM1 databases that reside on local or remote servers. The Perspectives application is run as an add-on to Excel.

TM1 Architect is a stand-alone windows application. TM1 Architect is also a client application that lets you create, manage, access, and edit TM1 databases that reside on local or remote servers. Architect is not run as an add-on to Excel, and is thus useful for administrative functions where no overhead for Excel activities is incurred. It is also useful for those environments where Excel is not allowed to be present on the server.

TurboIntegrator is executed from within TM1 Architect or Perspectives.

The interface you choose depends on your role in a company and what you need to do.

### Instructor Notes

More details about each interface are available in the *TM1 Users Guide*.

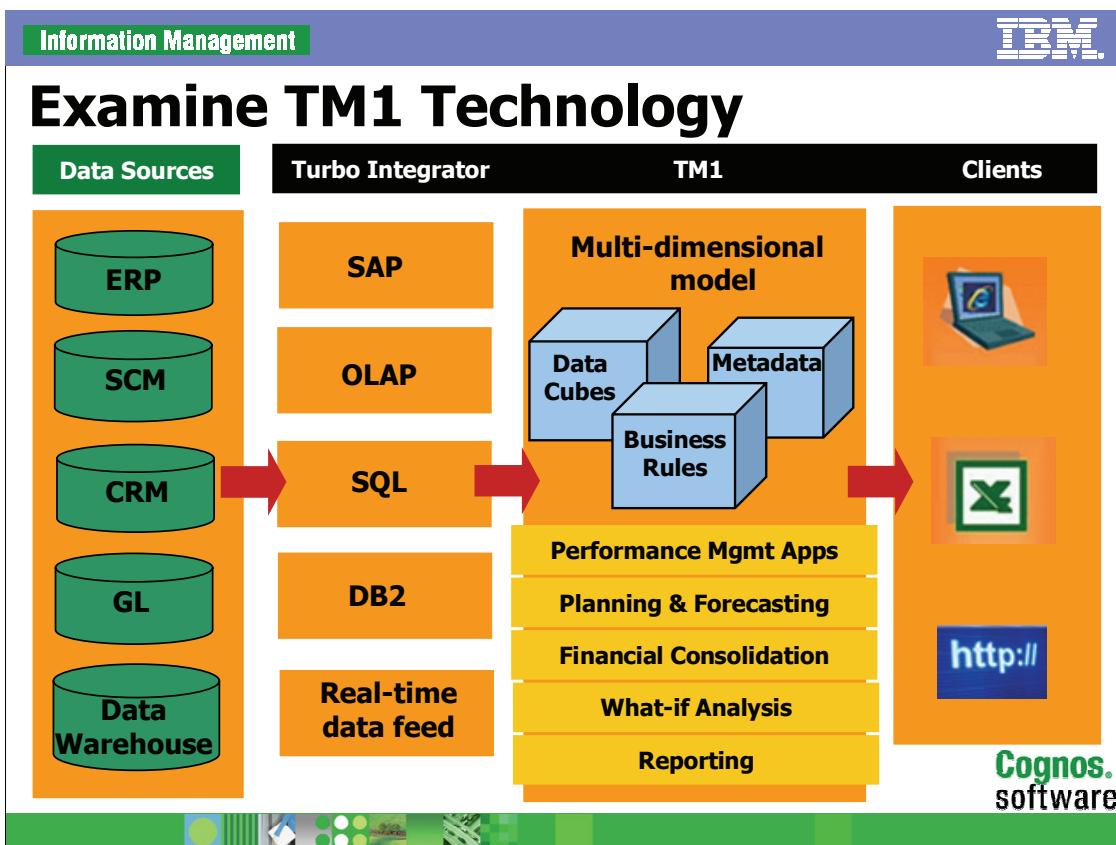
## Examine the Roles of TM1 Users

- The components you use and the tasks you perform in TM1 depend on your role.
  - Modeler
  - Administrator
  - Contributor
  - Explorer

**Cognos.**  
software



Role	Description
Modeler	Designs and builds multidimensional models, maps data flow, and applies business logic. May analyze data and perform predictive analysis. May create and customize reports, templates, and applications.
Administrator	Manages the technical deployment of applications, reviews hardware and software requirements, plans and supports the overall system infrastructure, users, groups, and security.
Contributor	Uses TM1 to answer questions by viewing reports, and may input data into templates.
Explorer	Views reports, presentations, and dashboards.



## Instructor Notes

**ERP** (Enterprise Resource Planning): planning of how business resources (materials, employees, customers etc.) are acquired and moved from one state to another (SAP is an example).

**SCM** (Supply Chain Management): planning and management of all activities involved in sourcing, procurement, conversion, and logistics management activities.

**CRM** (Customer Relationship Management): processes that handle key customer information. CRM software is used to support these processes, storing information on customers and prospective customers.

**GL** (General Ledger): collection of the group of accounts that supports the items shown in the major financial statements.

**SAP**: a large software enterprise company focusing on large ERP systems.

**DB2** and **SQL**: databases designed for the retrieval and management of data in relational database management systems.

## Important Files in TM1

- You can have multiple TM1 data servers on a machine.
- Each TM1 data server must have a tm1s.cfg file in the TM1 data server directory.
- All TM1 data files (cubes, dimensions, rules, etc) are maintained in memory, until you save them.

Cognos.  
software

Because all objects reside in memory, processing time is very fast.

When objects are saved, only the leaf-level element is saved. Consolidations and calculations are not stored, they are performed as required. They do remain in memory on the server until the server is recycled. If another user requests the same calculations, they will receive the results much faster than the first time because TM1 will give them the value that resides in memory.

You are prompted to save files when you stop the TM1 data server, or you can choose to save data manually or using Turbo Integrator. You will see this later in the course.

## Demo 1: Connect to an Existing TM1 Server

### Purpose:

You want to add a connection to your TM1 server and browse through the default items that appear in Server Explorer.

### Task 1. Browse through the different TM1 Services that are installed.

1. From the **Start** menu, click **Control Panel**, **Administrative Tools**, **Services**.
2. Scroll down to **TM1 Admin Server**.
3. Ensure that the **TM1 Admin Server** is **Started**.

The Admin Server runs as a service and keeps track of all TM1 servers running on a network. When a TM1 server starts, the server registers itself with an Admin Server that is running on a specified Admin Host.

The TM1 server will not run if the Admin server is not started.

4. Close **Services**, and then close **Administrative Tools**.

### Task 2. Start the Great Outdoors TM1 Server.

1. On the **Desktop**, right-click the **Great Outdoors** icon, and then click **Properties**.

The Target box shows that tm1s.exe is run, and indicates the location of the tm1s.cfg configuration file. This file must be in the same directory as the server's data files.

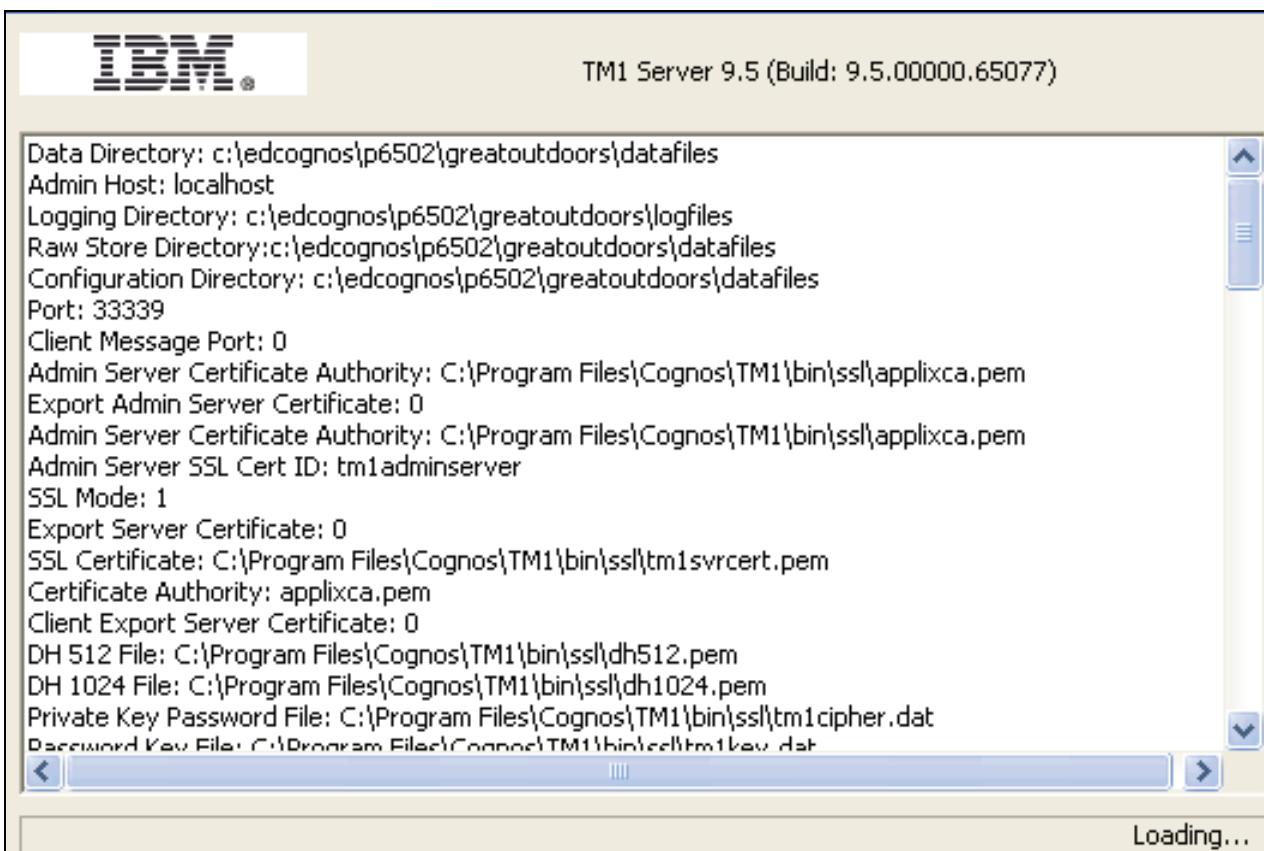
---

The TM1 Application server(s) which are run as a service will run even if the Admin service is not started, as long as it is set to Automatic. If not set to Automatic, and not already started, TM1 servers set up as services that point to this Admin service will not start as they depend on this service. TM1 Desktop application servers will run regardless of an Admin service running or not, as they will automatically start a Desktop Admin Server if an Admin Service is not available.

2. Click **Cancel**, and then double-click **Great Outdoors**.

The Great Outdoors TM1 server is started as a desktop application.

The result appears as shown below:



When the server is running, the TM1 Server appears minimized on the toolbar.

You can now start working with TM1, either by using Architect, Perspectives (through Excel) or a web browser.

## Task 3. Connect to TM1 using Architect.

1. From the **Start** menu, click **All Programs**, **IBM Cognos**, **TM1**, and then click **Architect**.
2. In **Server Explorer** click to expand **TM1**, double-click **greatoutdoors**, and then log in with:  
UserName: **admin**  
Password: **<blank>**  
You can now create an application. An application can contain cubes, dimensions, replications, processes, and chores. Throughout this course, you will modify this application.
3. Click **OK**.
4. Minimize **TM1 Architect** for the next demo.

### Results:

You added a connection to your GreatOutdoors TM1 server and then browsed through the default items that appear in Server Explorer.

---

You can start Perspectives if you will need to slice data into Microsoft Excel. We will not be slicing in this course so we will use Architect.

You may wish to have students create a shortcut on the desktop instead of navigating the menu.

## Demo 2: Create a New TM1 Server

### Purpose:

You want to create a new TM1 Server for the SData sample data and add a desktop icon to connect to it. Finally, you will test the new server that you created.

Note: The server that you need for subsequent demos has already been created on the image.

### Task 1. Copy the configuration file.

1. Browse to **C:\Program Files\Cognos\TM1\Custom\TM1Data\PlanSamp** and copy **tm1s.cfg**.
2. Browse to **C:\Program Files\Cognos\TM1\Custom\TM1Data\SData** and then paste the file.
3. Create the subfolder **Logfiles** under the **SData** folder.
4. Right-click **tm1s.cfg**, and then open with **WordPad**.

No two TM1 data servers can be running from the same port or server name, so you need to change them.

You also need to indicate the correct DataBaseDirectory and LoggingDirectory. The directory indicates where the server stores its data and log files.

5. Change the **ServerName** from **ServerName=Planning Sample** to **ServerName=SData**.  
You cannot have two data servers with the same name on the same machine.
6. Change the **DataBaseDirectory** path to **C:\Program Files\Cognos\TM1\Custom\TM1Data\SData\**.

7. Change the **PortNumber=12345** to **PortNumber=11111**.
8. Change **AdminHost=VCLASSBASE** to **AdminHost=localhost**.
9. Add a return after **DataBaseDirectory=** and then type the following:  
**LoggingDirectory= C:\Program Files\Cognos\TM1\Custom\TM1Data\SData\Logfiles\**

It is a recommended practice that you create a separate folder to place TM1 log files. This will allow data in the TM1 data folder to be backed up while the TM1 server is running, otherwise the server must be shut down in order to back up the server.

10. Save and close the file.

## **Task 2. Create a desktop shortcut to the new data server.**

1. Browse to **C:\Program Files\Cognos\TM1\bin** and create a shortcut of **tm1s.exe**.
2. Drag the shortcut to the desktop.
3. Rename the shortcut **SData**, and then right-click the shortcut and then click **Properties**.
4. In the **Target** box, at the end of the expression type **-z "C:\Program Files\Cognos\TM1\Custom\TM1Data\SData"**.

The complete expression appears as shown below:

**"C:\Program Files\Cognos\TM1\bin\tm1s.exe" -z  
 "C:\Program Files\Cognos\TM1\Custom\TM1Data\SData"**

The **-z** switch indicates the location of the server configuration file.

5. Click **OK**.

## Task 3. Start and test the server.

1. Double-click the **SData** shortcut on the desktop, to start the new TM1 server.
2. Open **TM1 Architect**, and then from the **File** menu, click **Refresh Available Servers**.  
The new testserver appears.
3. Double-click **sdata**.
4. Log in with UserName: **admin**, Password: **apple**.  
You have created a new data server in TM1.
5. Right-click **SData**, click **Security** and then click **Change Password**.
6. Click **OK** twice.
7. Right-click **SData** and then click **Log Out**.
8. Log in with UserName: **admin**, Password: <blank>
9. Close **TM1 Architect: Server Explorer**, and then close all open windows and both servers, saving changes when prompted.

### Results:

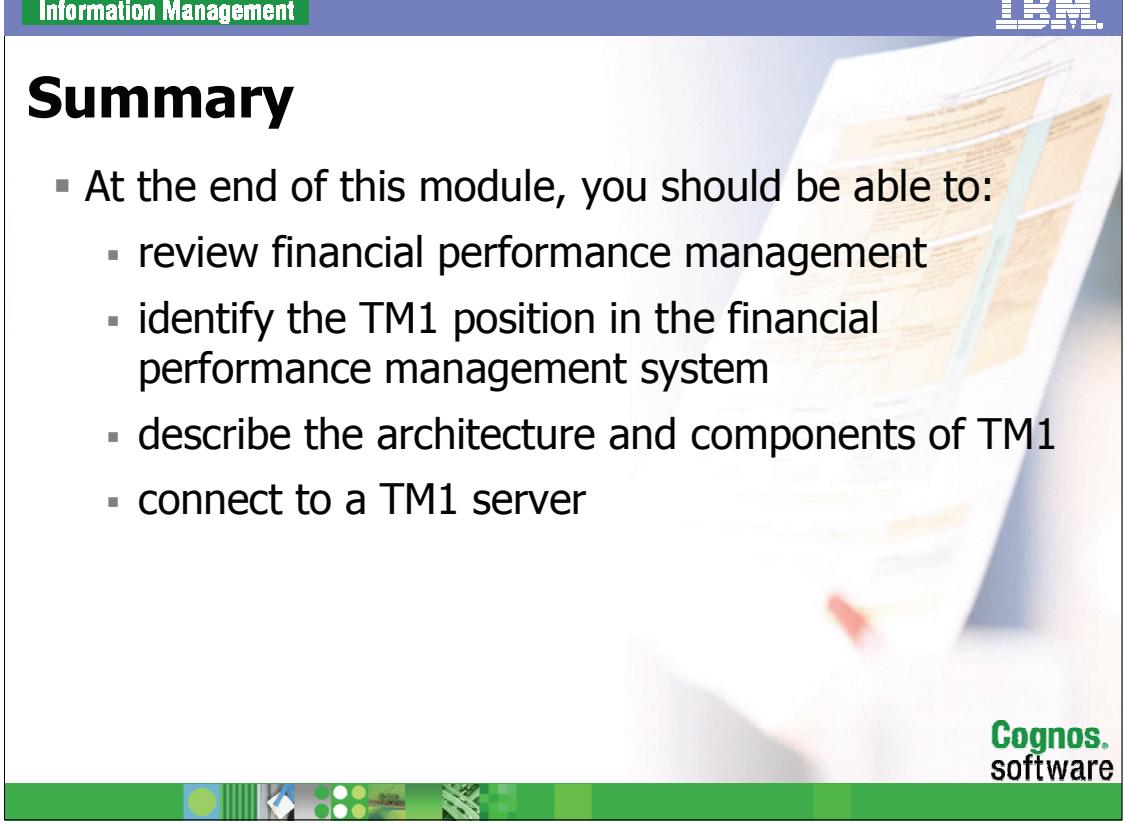
You created a new TM1 Server, and added desktop icon to connect to it. Finally, you tested the connection.

---

You can start a data server as a windows service or as a desktop shortcut. In this example, you create a desktop shortcut because error messages are clearer and more readily available. If there are problems starting the windows service, you need to open the tm1server.log files.

## Summary

- At the end of this module, you should be able to:
  - review financial performance management
  - identify the TM1 position in the financial performance management system
  - describe the architecture and components of TM1
  - connect to a TM1 server



Cognos  
software



---

Before continuing to Module 2: Create Dimensions, please review Appendix E: Identify Common Data Structures.



**Information Management**



## Create Dimensions

IBM Cognos TM1 9.5



**Cognos.**  
software

© 2009 IBM Corporation



# Objectives

- At the end of this module, you should be able to:
  - review cubes and dimensions
  - create dimensions manually
  - edit dimensions
  - create dimensions using Turbo Integrator

Cognos.  
software



---

Please review Appendix E: Identify Common data Structures before proceeding with the current module.

**Information Management**

**IBM**

# What is a Dimension?

- Fundamental building blocks of TM1 cubes.
- Contain one or more hierarchies.

All Products

- Camping Equipment
  - Tents
  - Lanterns
  - Sleeping Bags

Sales

- Unit Price
- Quantity Sold
- Discounts

Total Year

- Q2
  - April
  - May
  - June

**Cognos. software**

Dimensions are made up of elements which are structured lists of related items that define the data stored in cubes.

The slide shows that the time dimensions contain hierarchies where months roll up into quarters and quarters into the year. The product dimension also contains hierarchies where product types roll into a product line and product line into all product lines.

---

TM1's aggregation is implicit. You do not need to specify the children roll up to a parent. The consolidation is assumed (don't need to be defined by calcs)

## Classes of Dimensions

- Dimensions may be classified in the following categories:
  - Foundation
    - Example: Account, Customer, Product
  - Control
    - Example: Version, Scenario
  - Time
    - Example: Year, Month, Hour
  - Measure



Cognos.  
software

Foundation dimensions describe the data.

Control dimensions are most often used to implement business processes.

Time dimensions represent the critical time of the data.

The measures dimension represents the cardinal values of interest. There should be a measures dimension for almost every cube. It may be a good idea to name it after the cube for which it will be used along with a prefix or suffix, marking it as the measures.

---

Note: these classifications are theoretical and not enforced or marked in TM1.

Cubename\_m or cubename\_measures may be used. In this course, we will use cubename\_Measures to identify the measures dimensions.

## Naming Dimensions

- Effective and consistent naming are vital for maintainability
- Dimension names should be:
  - indicative of the dimension granularity
  - concise
  - unambiguous

 Cognos.  
software

Dimensions identify the location of a data point in a cube. If a dimension name were to change, that changes the location of the data. For this reason, naming of objects is crucial to the model.

Consistent naming conventions should be established and enforced in the model.

---

Objects in TM1 may not be renamed. Instead they must be deleted and recreated. It is often recommended the data is exported to text before deleting.

This concept is true for all objects in TM1 not just dimensions including Elements, Cubes, Processes, etc.

**INTERACTION - Text Chat:** Ask for examples of poorly named dimensions. Ask people to suggest better names.

## What is an Element?

- Elements are used to define data in cubes.
- Every element has a type associated with it:
  - simple (number)
  - string
  - consolidated element
- As application developers, you will create, edit and format the attributes of the elements (precision, format).

**Cognos**  
software



Symbol	Element Type	Description
	Simple	An element without any children. Also known as a leaf element. It is the lowest level in the hierarchy.
	String	A simple element that identifies one or more cells as containing strings.
	Consolidated	An element that has aggregated data in a cube. Two or more elements roll up to a consolidated element.

String elements are not aggregated and are usually used for descriptive purposes. If these exist, they must be created in the last dimension in the cube.

It is best practice to make the Measures dimension the last dimension.

TM1 only stores data at the leaf level element, consolidations are done in RAM. This allows TM1 to quickly compute calculations.

## Using Weights with Elements

- Weights identify the contribution of an element to a consolidation.
- Default weight factor is 1.0.

$$\begin{array}{rcl} \text{Total Q1} & = & \text{January} \\ \text{Revenue} & & \text{Revenue} + \text{February} \\ \text{Weight 1.0} & & \text{Weight 1.0} + \text{March} \\ & & \text{Revenue} \\ & & \text{Weight 1.0} \end{array}$$

Can also consolidate elements by subtracting the values associated with the element.

$$\begin{array}{rcl} \text{Net} & = & \text{Total} \\ \text{Revenue} & & \text{Revenue} + -\text{Total Sales} \\ \text{Weight 1.0} & & \text{Weight 1.0} \quad \text{Deductions} \\ & & \text{Weight -1.0} \end{array}$$

Cognos software



TM1 assumes all children will be aggregated (addition).

Credit balances may be stored unsigned and negative weights allow you to apply a reduction.

Weighting is a property of a child's relationship to a parent. It allows the value of the child element to be subtracted from the parent value (instead of aggregated).

---

Net Rev = Total Revenue + (Total Sales Ded. \* (-1))

**INTERACTION - Shapes > Oval:** Use the oval to highlight the expression being discussed.

# Using Attributes

- Element attributes describe an element.
- Use attributes to:
  - describe features of elements
  - provide alternative names, or aliases
  - format the display for numeric data

Cognos.  
software

If you want to:	Attribute Type:	Example:
describe features of an element	text or numeric	<ul style="list-style-type: none"> <li>• square footage of a store</li> <li>• the engine size of a car model</li> </ul>
provide alternative names, or aliases	alias	<ul style="list-style-type: none"> <li>• descriptive names of general ledger accounts</li> <li>• element name in another language</li> </ul>
format the display for numeric data	format	<ul style="list-style-type: none"> <li>• number decimal places</li> <li>• currency or percentage</li> <li>• how negatives appear</li> </ul>

**INTERACTION - Text Chat:** Ask for other examples of each of the element types. May also create a whiteboard, make three columns and have participants add examples of each.

## Create Dimensions

- You can create dimensions:
  - manually
  - using Turbo Integrator
  - using Dimension Worksheets

Cognos.  
software



It is best practice to name the dimension with an intuitive noun that indicates it is a leaf-level element.

TM1 supports the use of spaces and some special characters as names. However, it is better to use underscores (\_) instead of spaces in order to minimize conflict with other applications.

## Demo 1: Create Dimensions Manually

### Purpose:

You want to create the Price and Cost, Months, and Versions dimensions manually.

### Task 1. Create the Price\_and\_Cost\_Measures dimension.

TM1 Server: greatoutdoors

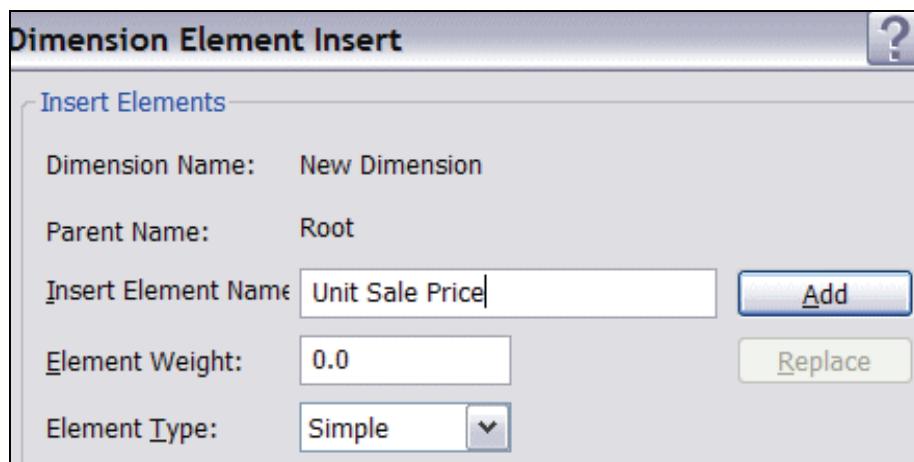
TM1 - Architect: Server Explorer

UserName: admin

Password: <blank>

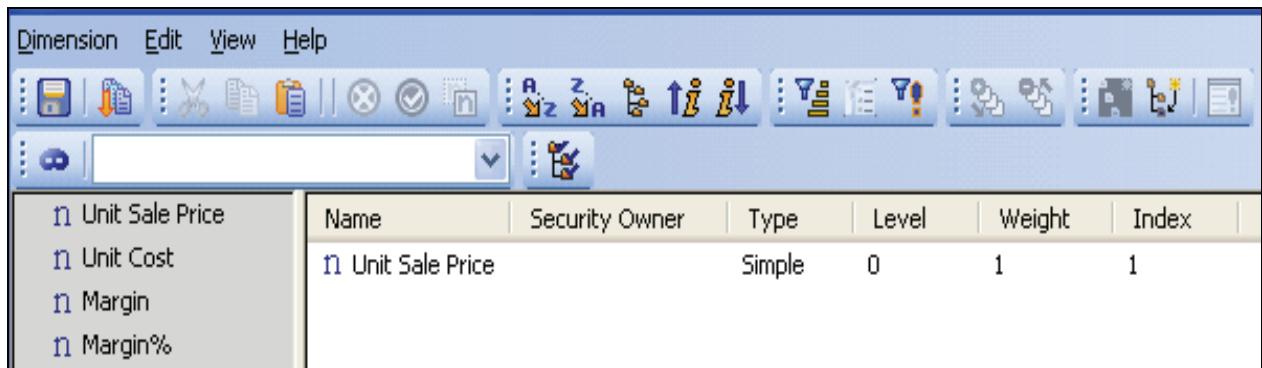
1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, and then double-click **greatoutdoors**.
3. In the **UserName** box, type **admin**, and then click **OK**.
4. Right-click **Dimensions**, and then click **Create New Dimension**.
5. From the **Edit** menu, click **Insert Element**.
6. In the **Insert Element Name** box, type **Unit Sale Price**.

The result appears as shown below:



7. Leave the default Element Weight, and Element Type, and then click **Add**.  
The new element appears in the Inserted Elements pane.
8. Repeat Steps 5 and 6 to add **Unit Cost**, **Margin** and **Margin%**.
9. Click **OK**.

The result appears as shown below:



The screenshot shows the IBM Cognos TM1 Dimension Editor window. The menu bar includes Dimension, Edit, View, and Help. The toolbar contains various icons for file operations, search, and navigation. The main pane displays a table of dimensions with the following data:

	Name	Security Owner	Type	Level	Weight	Index
Unit Sale Price	Unit Sale Price		Simple	0	1	1
Unit Cost						
Margin						
Margin%						

If the Dimension Editor does not appear with two panes as shown, on the toolbar, click Display Properties Window .

10. From the **Dimension** menu, click **Save**.
11. In the **Name** box, type **Price\_and\_Cost\_Measures**, and then click **OK**.  
Next you will create the Months dimension. Instead of typing each month, use Excel to populate the months.
12. Click **OK** to close the Dimension Editor.

## Task 2. Create the Months dimension.

1. In **TM1 - Architect: Server Explorer**, right-click **Dimensions**, and then click **Create New Dimension**.
2. Open **Excel**, create a new blank workbook, and then in cell **A1**, type **Jan**.  
You will use autofill to populate the remaining months of the year.
3. Click cell **A1**, and then place the pointer over the lower right corner of cell A1, until you see a plus sign.

---

Task 1 Step 7. You can also press <Enter>, to add elements to the Inserted Elements pane.

4. Click and drag the cursor to cell **A12**, and then release the mouse.
- A list of all 12 months appears.
5. Copy the 12 months in **Excel**, and paste them in the grey pane in the **Dimension Editor**.
6. Save the dimension with the name **Months**.

The result appears as shown below:

Name	Security	Ownership
Jan		
Feb		
Mar		
Apr		
May		
Jun		
Jul		
Aug		
Sep		
Oct		
Nov		
Dec		

The **n** beside each element indicates that these are leaf level numeric elements, at the lowest level of detail in the dimension. In a cube that contains only numbers, all the lowest level elements are numeric.

7. Close the Dimension Editor.

Next you will create the Versions dimension.

It is a proven practice to include the measures dimension as the last in the cube. However, dimensions can be created in the order of your choice.

The measure dimension is usually named <cubename>\_m.

In the next module you will create the Price and Cost cube.

TM1 supports string elements. They are not aggregated and are usually used for descriptive purposes. If these exist, they must be defined in the last dimension in the cube.

A dimension doesn't actually contain anything, it maps to data points when used in a cube. The text string "Jan" really doesn't point to text. It points to a value that "Jan" describes.

## Task 3. Create Versions dimension.

1. Right-click **Dimensions** and click **Create New Dimension**.
2. In the **Dimension Editor**, insert the following elements:
  - **Budget Version 1**
  - **Budget Version 2**
3. Save the dimension with the name **Versions**, and then close the **Dimension Editor**.

### Results:

You have created the **Price\_and\_Cost\_Measures**, **Months** and **Versions** dimensions.

## Demo 2: Modify Existing Dimensions

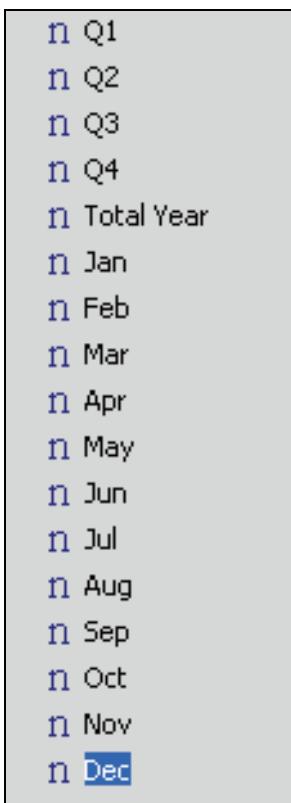
### Purpose:

You want to modify the Months dimension, to include the elements for quarters and year, and then sort the Months dimension. You will also add descriptive attributes for each month, to identify the next and previous months as well as an alias for each month to be able to refer to each month in either short or long form. You will also edit the attribute elements to format the Price\_and\_Cost\_Measures dimension.

### Task 1. Add Quarters and Total Year elements to the Months dimension and then sort the dimension.

1. In Server Explorer, right-click Months, and then click **Edit Dimension Structure**.
2. From the **Edit** menu, click **Insert Element**, and then insert the following elements with the default Element Weight and Element Type:
  - Q1
  - Q2
  - Q3
  - Q4
  - Total Year

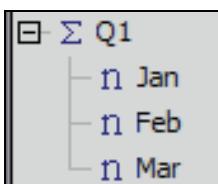
The result appears as shown below:



3. Click **Jan**, and then Shift+click **Mar**, to select Jan to Mar, and then drop the items beside Q1 when the following icon appears .

The icon indicates that the selected items will be dropped as child elements.

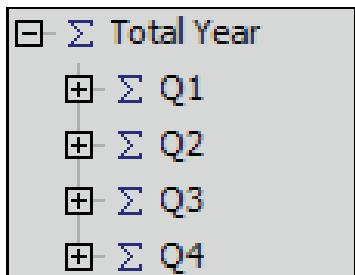
The result appears as shown below:



The "Σ" indicates consolidated items. Jan, Feb, and Mar are summed into Q1.

4. Right-click **Q1**, and then click **Element Properties**.  
Notice that the element type is "Consolidated".
5. Under **Element Type**, click the arrow beside **Consolidated**.  
Notice that the other available element types are Simple and String.
6. Click **Cancel**, and then view the **Element Properties** for **Jan**.  
Notice that the element type is Simple.
7. Click **Cancel**.
8. Select **Apr**, **May**, and **Jun**, and then repeat step 3 beside **Q2**.
9. Select **Jul**, **Aug**, and **Sep**, and then repeat step 3 beside **Q3**.
10. Select **Oct**, **Nov**, and **Dec**, and then repeat step 3 beside **Q4**.
11. Collapse and select **Q1**, **Q2**, **Q3**, and **Q4**, and then drop the selection beside **Total Year**.

The result appears as shown below:



Next, you want to make sure that these items will maintain this element order when you open the dimension again.

12. Click the **All** button and then click **Hierarchy Sort** button.

Instead of dragging and dropping items, you can cut (Edit/Cut) and paste (Edit/Paste) them.

13. On the toolbar, click **Set Dimension order** 

This button will reset the Index number of each element. Each time you add a new element, the next available index number is assigned to it. This number is often used in ordering the dimension. If you moved the elements around, you may wish to reset the index number to reflect the new order.

However, index numbers are sometimes used in TM1 functions so you want to be sure you will not alter the result of these functions if you reorder the dimension.

14. Save the **Months** dimension, and then click **OK** to close the Dimension Editor.

## **Task 2. Add element attributes for Next and Previous Months.**

Next you will add descriptions that identify next and previous months.

1. In **Server Explorer**, right-click the **Months** dimension and then select **Edit Element Attributes**.
2. From the **Edit** menu, click **Add New Attribute**.
3. In the **Name** box, type **Next**, under **Type**, ensure that **Text** is selected, and then click **OK**.
4. Repeat steps 2 and 3, and in the **Name** box, type **Previous**.

You will need to identify the Next and Previous month.

5. Enter the values as shown below:

Note: You can paste autofill values from Excel.

	Previous (Text)	Next (Text)	Format (Text)
Tot	N/A	N/A	
Q1	Q4	Q2	
Jan	Dec	Feb	
Feb	Jan	Mar	
Mar	Feb	Apr	
Q2	Q1	Q3	
Apr	Mar	May	
May	Apr	Jun	
Jun	May	Jul	
Q3	Q2	Q4	
Jul	Jun	Aug	
Aug	Jul	Sep	
Sep	Aug	Oct	
Q4	Q3	Q1	
Oct	Sep	Nov	
Nov	Oct	Dec	
Dec	Nov	Jan	

### Task 3. Add attributes to alias the month name in full name form.

- From the **Edit** menu, click **Add New Attribute**.
- In the **Name** box, type **Full Name**, below type click **Alias** and then click **OK**.  
The warning that TM1 will set initial values for all objects for an alias attribute appears.
- Click **OK**.

You will have to manually change the values in the column to their full names.

4. Modify the values in the Full Name (Alias) column to the full names.

The result appears as shown below:

	Full Name (Alias)	Previous (Text)	Next (Text)	Format (Text)
Tot	Total Year	N/A	N/A	
Q1	Quarter 1	Q4	Q2	
Jan	January	Dec	Feb	
Feb	February	Jan	Mar	
Mar	March	Feb	Apr	
Q2	Quarter 2	Q1	Q3	
Apr	April	Mar	May	
May	May	Apr	Jun	
Jun	June	May	Jul	
Q3	Quarter 3	Q2	Q4	
Jul	July	Jun	Aug	
Aug	August	Jul	Sep	
Sep	September	Aug	Oct	
Q4	Quarter 4	Q3	Q1	
Oct	October	Sep	Nov	
Nov	November	Oct	Dec	
Dec	December	Nov	Jan	

5. Click **OK**.

#### **Task 4. Format the elements in the Price\_and\_Cost\_Measures dimension.**

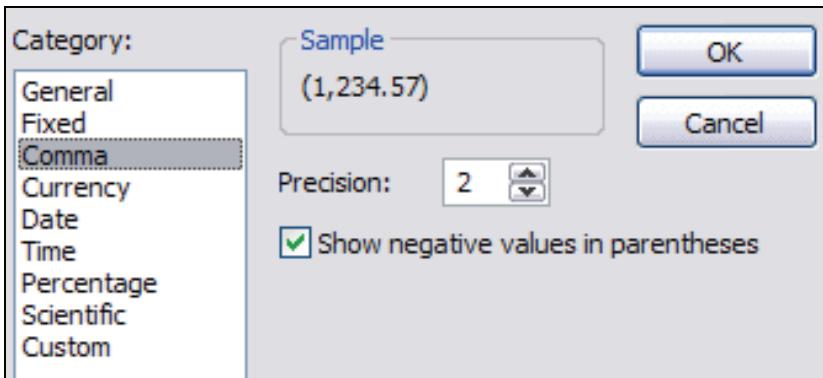
1. In the **Server Explorer** window, right click **Price\_and\_Cost\_Measures** and then click **Edit Element Attributes**.
2. Click and drag from the **Unit Sale Price** cell to the **Margin** cell to highlight all three cells.

---

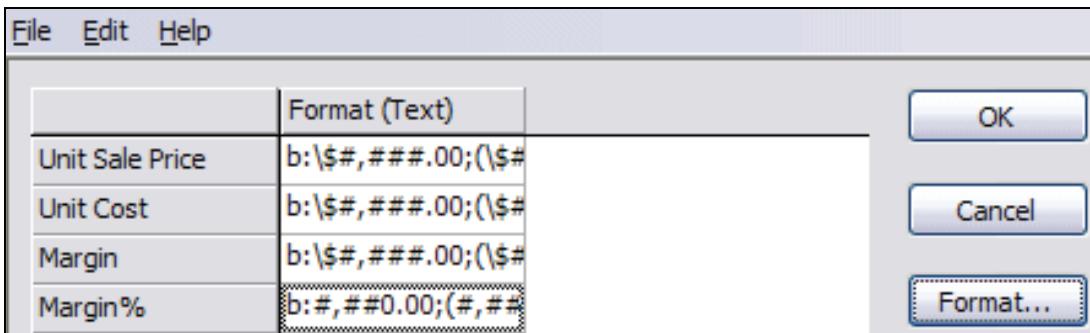
It is recommended that you format only the measures dimension. This way there is no conflict between formatting on dimensions in a cube.

Ctrl keys do not work in this window so you must click and drag to select multiple cells.

3. Click **Format**.



4. Click **Currency** and then click **OK**.
5. Click **Margin%**, and then click **Format**.
6. Click **Comma**, and then click **OK**.



7. Click **OK**.

**Results:**

You modified the Months dimension, to include the elements for quarters and year, and then sorted the Months dimension. You also added descriptive attributes for each month, to identify the next and previous months as well as an alias for each month to be able to refer to each month in either short or long form. You also edited the attribute elements to format the Price\_and\_Cost\_Measures dimension.

## What is Turbo Integrator?

- Turbo Integrator (TI) is an extraction, transformation, and loading (ETL) tool built into TM1
- Use TI to:
  - create and update dimensions
  - create and update cubes
  - load data
  - maintain model objects using TI functions

**Cognos.**  
software



Turbo Integrator is used to create a process that recognizes the data structure of the source and transform it into the appropriate structure for the model. Once the TI Process is developed, you can re-run it or schedule it to be used when importing data from a dynamic source.

TI may import data from a variety of sources including TM1, text files, ODBC and more.

In this module, the focus is creating dimensions however; additional uses will be discussed in the following modules.

Whenever creating an object using TI, be sure to review the resulting object for accuracy.

---

It is very possible a script will run without an error but the resulting structure is not what was intended. This may be due to bad or unexpected data or a number of other reasons so it is recommended you always review the resulting objects for accuracy. (Garbage in, garbage out)

## Key Steps in TI

- TI uses a number of windows to aid in the development of a process including:
  - Data Source
  - Define Variables
  - Map Data
  - Advanced
  - Schedule



Cognos.  
software

The data source tab allows you to select and preview the data from the data source as it appears to TI.

Define Variables is used to identify how the incoming data will be transformed to be used in the TM1 structures.

Map data allows for the more detailed mapping of data structures and objects.

The Advanced tab displays and is used to edit the scripts used by a TI process.

The Schedule tab is used to schedule the process to run repetitively or off hours.

### Instructor Notes

For even more detail, there is a separate TI Guide for Turbo Integrator called tm1\_turb.pdf. It is located in c:\program files\cognos\tm1\DocLibrary\eng (or your local language code).

## Demo 3: Create a Dimension Using Turbo Integrator

### Purpose:

You will create the Products dimension using Turbo Integrator. You will identify text data as the source, create variables to transform the data, map the variables to TM1 objects, review the scripts generated and run the process. Finally, you will review the resulting dimension for accuracy.

### Task 1. Add data source connection information.

1. In the Server Explorer window, right click **Processes**, and then click **Create New Process**.
2. On the **Data Source** tab, click **Text**.
3. Click **Browse** and navigate to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles** and then double-click **Products.csv** to open the file.
4. Click **OK** to dismiss the warning.

If you were working on a remote server, you would use a Universal Naming Convention (UNC) path, for files. The UNC path is of the form `\ComputerName\SharedFolder\Resource`.

5. Ensure that the **Delimiter Type** is **Delimited**, and the **Delimiter** is **Comma**.
6. Click **Preview**.

You can see the first 10 rows of data in your source.

## Task 2. Modify variable names.

1. Click the **Variables** tab.
2. In the first row, double-click **V1**, and rename it **vProducts**, and leave the Variable Type column unchanged, as String.

This column is used to identify to TI what type of data is coming in from the source. You define formulas to transform this Variable Type into the final data structure required for the TM1 object being created/updated in the TI process.

3. In the **Contents** column, in the **vProducts** row, click the down arrow , and then click **Element**.

The Contents column identifies how the incoming data will be used or transformed in the TI process. It is defined by selecting one of the following settings:

Contents	Description
<b>Ignore</b>	Ignore the contents of the column when processing the data source.
<b>Element</b>	The column contains simple elements.
<b>Consolidation</b>	The column contains consolidated elements.
<b>Data</b>	The column contains data values in a cube.
<b>Attribute</b>	The column contains element attributes for the dimension you want to create or update.
<b>Other</b>	The column contains data that does not fall into any of the previous four categories. Typically, this setting is used for columns containing data that will be processed through custom variables and formulas.

- Repeat steps 2 to 4 with the following information.

Old Variable Name	New Variable Name	Contents
V2	vProductTypes	Consolidation
V3	vTotalProduct	Consolidation

The result appears as shown below:

	Variable Name	Variable Type	Sample Value	Contents	F
1	vProducts	String	Cooking Gear	Element	▼
2	vProductTypes	String	CAMPING EQUIPMENT	Consolidation	▼
3	vTotalProduct	String	TOTAL PRODUCTS	Consolidation	▼

### Task 3. Identify how data is mapped.

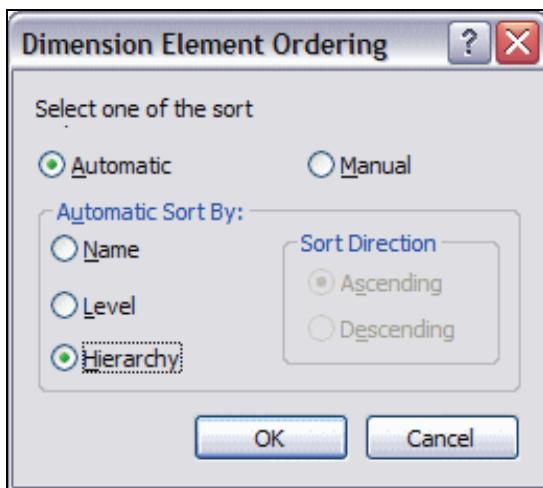
- Click the **Maps** tab, and then click the **Dimensions** tab.
- For the vProducts element variable, in the **Dimension** column, type **Products**.
- In the **Action** column, ensure that **Create** is selected, and in the **Element Type** column, ensure that **Numeric** is selected.

Numeric is the default and should only be changed if the data will be used for a String element. When you review the elements in a dimension in the Subset or Dimension Editors, they appear with either an N or an S (at the lowest level) identifying a numeric or a string element. This setting should reflect the N or S indicated here.

- In the Element Order column, click **By Input**.

5. Click **Automatic** and then click **Hierarchy**.

The result appears as shown below:



6. Click **OK**.
7. Click the **Consolidations** tab.

This tab identifies the parent-child relationship(s) in the dimension. You can map more than one hierarchy here if necessary.

It is often useful to diagram the parent-child relationships on paper or in a diagram tool prior to this screen:

vTotalProduct

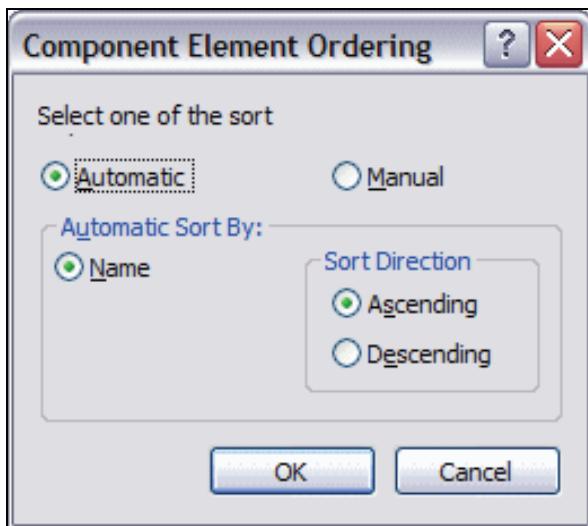
vProductTypes

vProducts

8. For the **vProductTypes** consolidation variable, click the child variable **vProducts**.
9. For the **vTotalProduct** consolidation variable, click the child variable **vProductTypes**.

10. Under Component Order for vProductTypes, click **By Input**.
11. Click **Automatic**.

The result appears as shown below:



12. Click **OK**.

The results appear as follows:

Cons. Variable	Dimension	Child Variable	Weight	Sample Value	Component Order
vProductTypes	Products ►	vProducts ►	1.000000	CAMPING EQUIP	By Name
vTotalProduct	Products ►	vProductTypes ►	1.000000	TOTAL PRODUC	By Input

13. Repeat steps **10 - 12** for the vTotalProduct.

---

It is recommended that you click on each tab under Advanced. This will ensure that TI scripts are updated (sometimes they do not but development has been unable to resolve this but clicking the tabs force the update). It is also a good idea to review the scripts that have been generated. You may want to mention these tabs correspond to 4 scripts run by TI when a process is executed. These scripts will be discussed in later modules.

## Task 4. Save and Execute Process.

1. Click the **Advanced** tab.
2. Click the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.

There are four scripts that are run when executing a TI process. There is a tab for each of them here (along with a tab to identify run-time variables) for the given process.

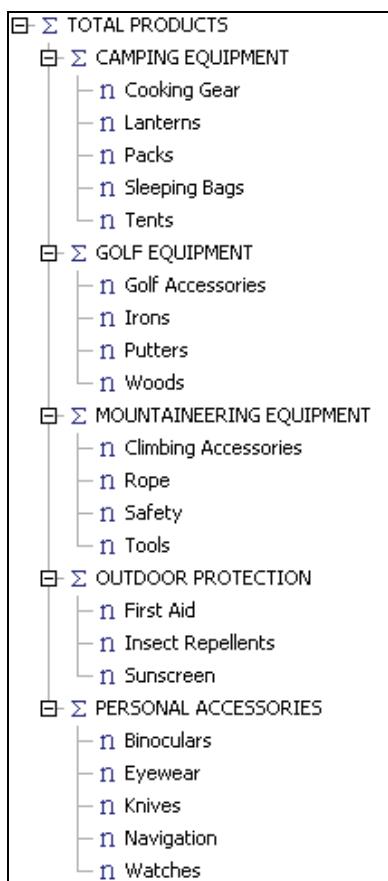
It is a good practice to review each script and ensure they have been updated, especially if you are modifying an existing process.

Clicking each tab will force the scripts on each tab to be updated and compiled.

3. Click **Save**.
4. In the **Name** box, type **CreateProductsCSV** and then click **OK**.
5. Click **Run** .
6. Click **OK** when the process has completed.
7. Close **Turbo Integrator**.

8. In the Server Explorer window, double-click the **Products** dimension.

The Subset Editor opens with all the elements that were loaded from the CSV file.



We ordered our elements in the TI process. If we had not chosen to order by Hierarchy & elements by name, we could reorder them using the Dimension Editor or in the Server Explorer window.

9. Close the Subset Editor.

### **Results:**

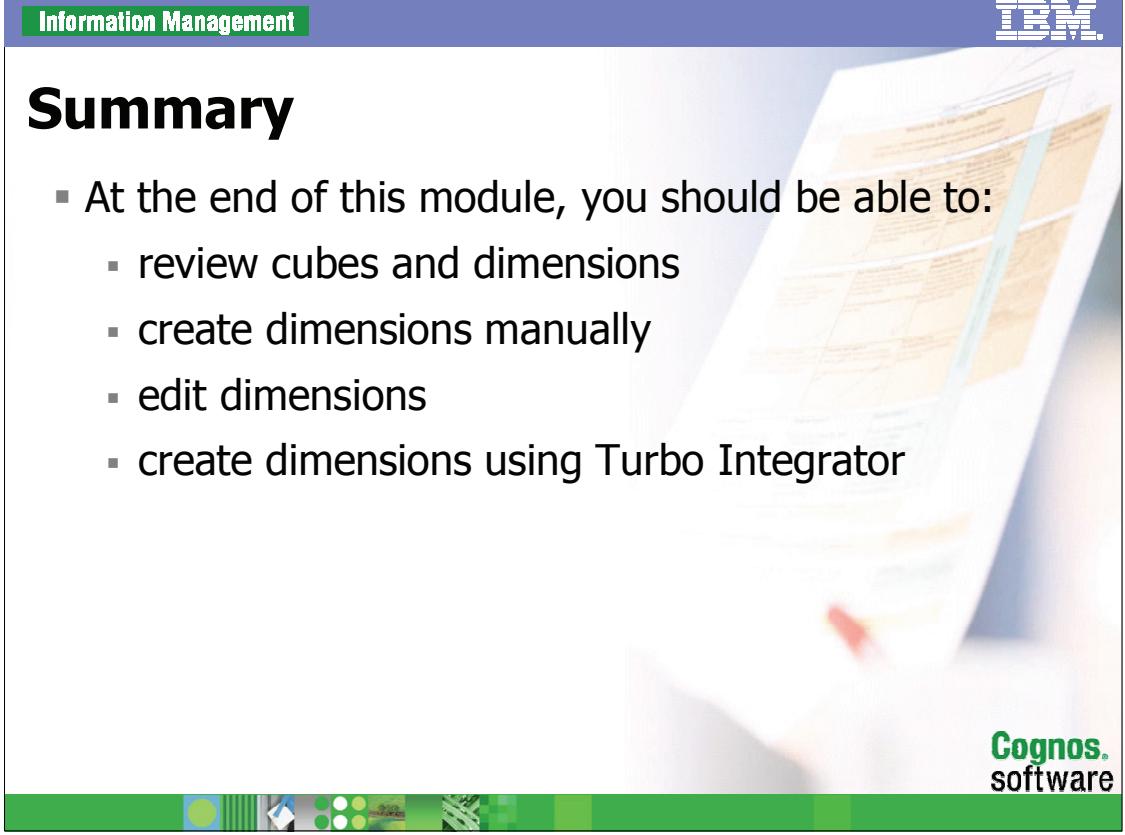
**You created a dimension using Turbo Integrator. You identified the variables, mapped the data, reviewed the script, executed the process and confirmed the resulting Products dimension.**

---

If you wish to order the dimension by hierarchy after running the TI process, in Server Explorer, right click the dimension name, click Set Elements Order, click Automatic, by Hierarchy and then click OK. Open the dimension in the Subset editor and it should now appear sorted by hierarchy.

## Summary

- At the end of this module, you should be able to:
  - review cubes and dimensions
  - create dimensions manually
  - edit dimensions
  - create dimensions using Turbo Integrator



Cognos  
software



## Workshop 1: Create a Dimension using Turbo Integrator

Analysts have mentioned that they need to be able to analyze sales revenue by distribution channel, or type of store. To satisfy this requirement, create the Channels dimension using TM1.

To accomplish this:

- Add data source connection information, to connect to the Channels.csv file on C:\Edcognos\P6502\greatoutdoors\SourceFiles.
- Modify variable names.
- Identify how data is mapped.
- Save the process as CreateChannelsCSV and execute the process.

For more detailed information outlined as tasks, see the Task Table section.

For the final query results, see the Workshop Results section that follows the Task Table section.

## Workshop 1: Task Table

Task	Where to Work	Hints
1. Add data source connection information.	Turbo Integrator/Data Source tab	<ul style="list-style-type: none"> <li>• Data source is a text file.</li> <li>• C:\Edcognos\P6502\greatoutdoors\SourceFiles\Channels.csv.</li> </ul>
2. Modify variable names.	Turbo Integrator/Variables tab	<ul style="list-style-type: none"> <li>• vChannels as Element.</li> <li>• vTotalChannels as Consolidation.</li> </ul>
3. Identify how data is mapped.	Turbo Integrator/ Maps tab/ Dimensions tab  Turbo Integrator/ Consolidations tab	<ul style="list-style-type: none"> <li>• Dimension - Channels.</li> <li>• Action - Create.</li> <li>• Element type - Numeric.</li> <li>• Element Order - by Hierarchy.</li> <li>• Child Variable - vChannels.</li> <li>• Component Order - by Name.</li> </ul>
4. Save and execute process.	Turbo Integrator/ Advanced tab	<ul style="list-style-type: none"> <li>• Click tabs.</li> <li>• Save as CreateChannels.</li> <li>• Run.</li> <li>• Hierarchy Sort.</li> </ul>

If you need more information to complete a task, see the Step-by-Step instructions at the end of the Workshop.

## Workshop 1:Workshop Results

After Task 1, the result appears as shown below:

	V1	V2
1	Golf Shop	ALL CHANNELS
2	Department Store	ALL CHANNELS
3	Direct Marketing	ALL CHANNELS
4	Warehouse Store	ALL CHANNELS
5	Equipment Rental Store	ALL CHANNELS
6	Outdoors Shop	ALL CHANNELS
7	Eyewear Store	ALL CHANNELS
8	Sports Store	ALL CHANNELS

After Task 2, the result appears as shown below:

Data Source	Variables	Maps	Advanced	Schedule
	Variable Name	Variable Type	Sample Value	Contents
1	vChannels	String	Golf Shop	Element
2	vTotalChannels	String	ALL CHANNELS	Consolidation

After Task 4 step 10, the result appears as shown below:



# Workshop 1: Step-by-Step Instructions

## Task 1. Add data source connection information.

1. In the Server Explorer window, right-click **Processes**, and then click **Create New Process**.
2. On the **Data Source** tab, click **Text**.
3. Click **Browse** and navigate to **C:\Edcognos\P6502\greatoutdoors\SourceFiles** and then double-click **Channels.csv** to open the file.
4. Click **OK** to dismiss the warning.
5. Ensure that the **Delimiter Type** is **Delimited**, and the **Delimiter** is **Comma**.
6. Click **Preview**.

You can see all 8 rows of data in your source.

## Task 2. Modify variable names.

1. Click the **Variables** tab.
2. In the first row, double-click **V1**, and rename it **vChannels**, and then leave the Variable Type column as String.
3. In the **Contents** column, in the **vChannels** row, click the down arrow  , and then click **Element**.
4. Click **V2**, and rename it **vTotalChannels**.
5. In the **Contents** column, in the **vTotalChannels** row, click the down arrow, and then click **Consolidation**.

## Task 3. Identify how data is mapped.

1. Click the **Maps** tab, and then click the **Dimensions** tab.
2. For the vChannels Element Variable, in the **Dimension** column, type **Channels**.
3. In the **Action** column, ensure that **Create** is selected, and in the **Element Type** column, ensure that **Numeric** is selected.
4. Click **By Input**, click **Automatic**, click Automatic Sort by **Hierarchy** and then click **OK**.
5. Click the **Consolidations** tab.
6. For the **vTotalChannels** consolidation variable, click the child variable **vChannels**.
7. Click **By Input**, click **Automatic**, click Automatic Sort by **Name** and then click **OK**.

## Task 4. Save and execute process.

1. Click the **Advanced** tab.
2. Click the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs, and then click **Save**.
3. In the **Name** box, type **CreateChannelsCSV** and then click **OK**.
4. Click **Run** , and then click **OK** when the process has completed.
5. Close **Turbo Integrator**.
6. In the Server Explorer window, double-click the **Channels** dimension.  
The Subset Editor opens with the elements that were loaded from the CSV file.
7. Close the Subset Editor.
8. Leave the Great Outdoors TM1 server running, but close all other open windows without saving.

**Information Management**



## **Build Cubes and Views**

IBM Cognos TM1 9.5



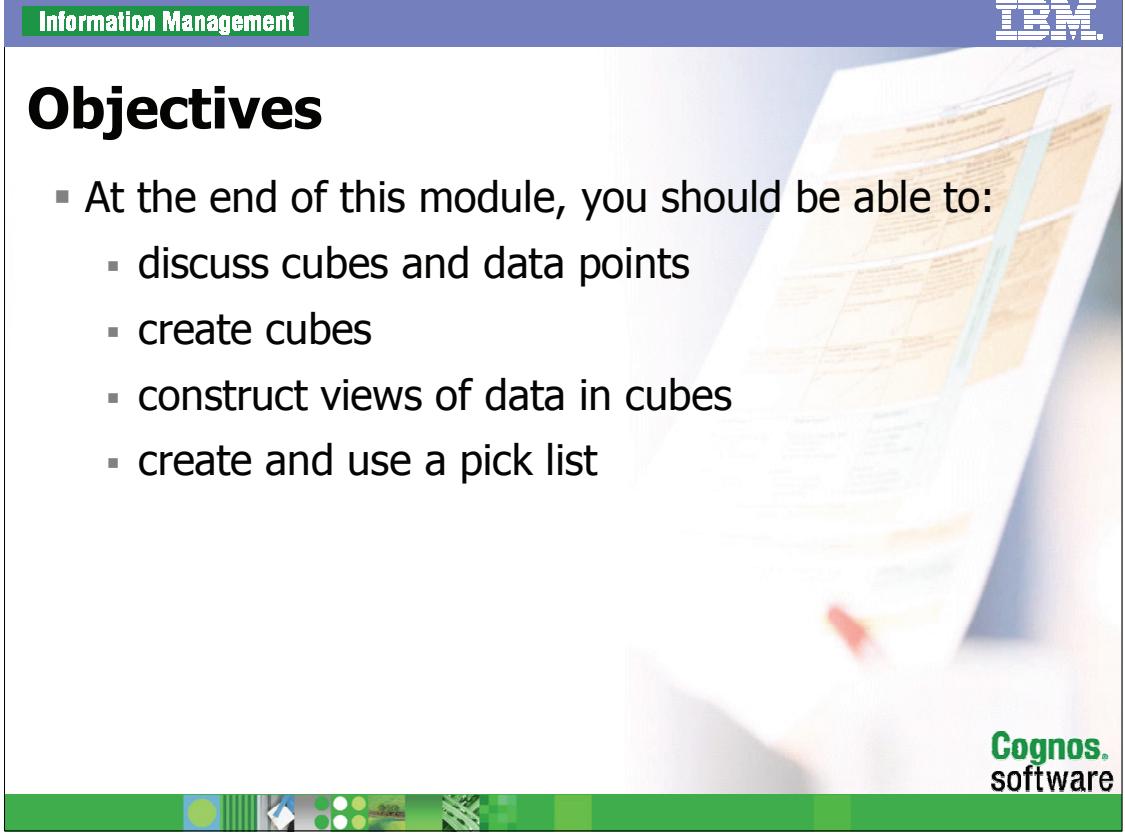
**Cognos.**  
software

© 2009 IBM Corporation



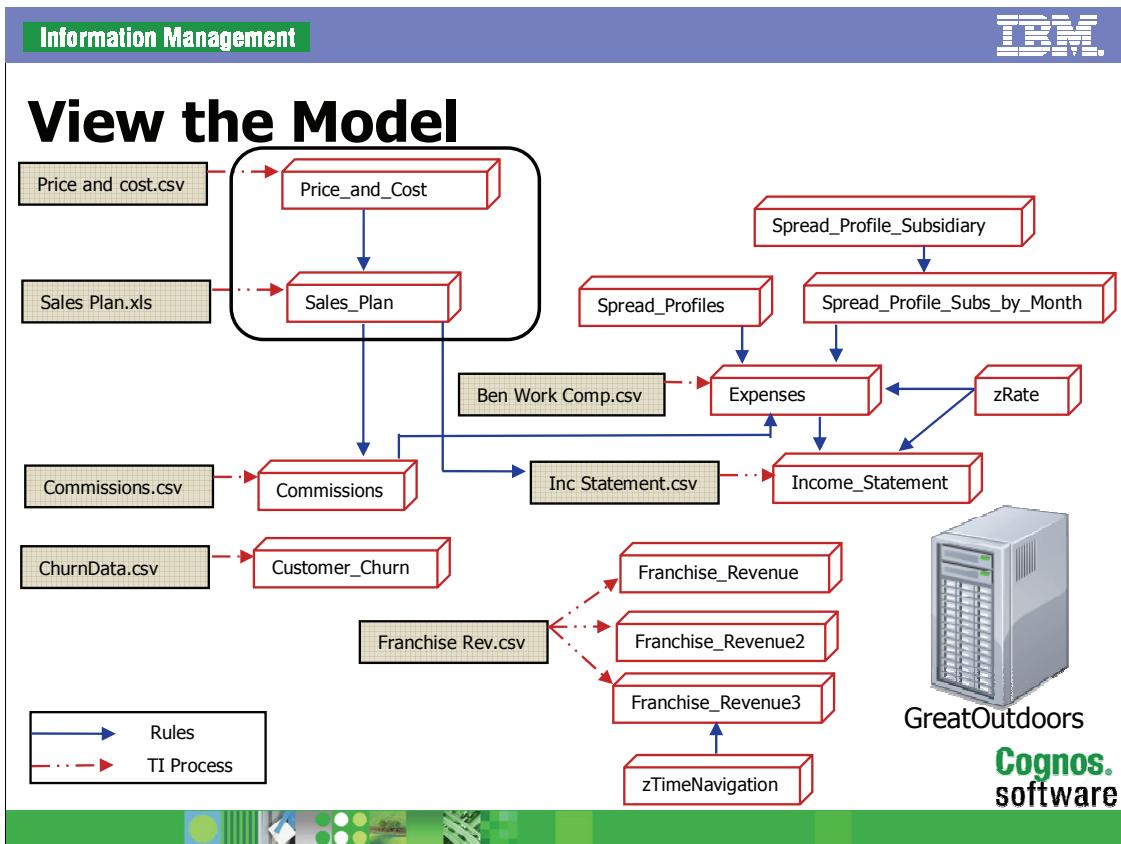
# Objectives

- At the end of this module, you should be able to:
  - discuss cubes and data points
  - create cubes
  - construct views of data in cubes
  - create and use a pick list



Cognos.  
software





In this module, you will create the following cubes:

- Price\_and\_Cost
- Sales\_Plan

## Store Data in Cubes

- Cubes store data for:
  - multi-dimensional analysis
  - collect data
  - enter data
  - calculate data



Cognos.  
software

You may have any number of cubes in your TM1 model.

Cubes store data for reports, analysis or staging data to be used in further calculations.

Data may come from a number of sources and be combined in a single cube or in multiple cubes using business logic.

---

TM1 can be its own 'data mart', pulling data from many sources into a cube for further analysis. HR has one system, GL has another; They share a common time and can be combined into a single cube or multiple cubes.

## Build a Cube

- Review data to load
- Determine the dimensions
  - dimensions will point to data points
- Name the cube

**Cognos.**  
software



Cubes are made up of two or more dimensions.

Dimensions are made up of one or more elements. Elements are consolidations, numeric, or string. Only numeric elements may be consolidated.

One element from each dimension identifies the location of a specific data point in a cube.

Establishing a naming convention for your model will make maintenance and expansion easier.

---

TM1 will support spaces and other characters in object names. However, it is better to use \_ instead of spaces and avoid using special characters. This will prevent conflicts with other applications and it will make things easier to read.

**INTERACTION - Whiteboard:** Make 3 columns on a whiteboard. Have people put examples of each type of element, consolidated, number and string.

## Consider Dimension Order

- Include a measures dimension last
- Order dimensions from:
  - smallest/sparse
  - largest/dense



Cognos.  
software

Every cube should have its own measures dimension, even if it only contains a single element (dollar, amount, or value).

The measures dimension should always be the last dimension specified in defining the cube.

Follow these recommendations to:

- make every cube extensible
- make cube data easier to locate
- ensure string measures will always work
- simplify rule and feeder statements

---

A sparse dimension has a low percentage of available data positions filled (e.g. Products, Markets, or Regions). A dense dimension has a high percentage of probability that one or more data points are occupied in every combination of dimensions (such as Version or Scenario).

Sparsity and density are usually estimated. For this reason this is a guideline and not a set rule. After loading data use the Cube Optimizer to improve calculation speed.

## Name the Cube

- Establish a naming convention
- Use a prefix indicating purpose:
  - lu\_ for look up cube
  - rp\_ for reporting cube
  - rf\_ for rates and factors



Cognos.  
software

Create a consistent naming standard for cubes.

Consider when naming cubes:

- Is it readable (use lowercase and underscores for spaces)?
- Should it contain prefix or suffix to identify their purpose?

---

The Server Explorer window sorts objects in alphabetical order. If you give a cube a prefix, this will allow similar cubes to be sorted together.

## Create a View

1. Arrange dimensions along:
  - title area
  - column area
  - row area
2. Recalculate the view
3. Save and name the view
  - public or private
  - default

 Cognos.  
software

A view saves the arrangement of dimensions for a cube.

Views of a cube are not calculated by default. After stacking or swapping dimensions click the Recalculate button (or press F9) to see the resulting data.

Only administrators may save public views for everyone to see.

An administrator may also publish a private view for public viewing.

---

Review functions such as Suppress Zeros, slice to Microsoft Excel, and Active Forms.

Review how to select and create subsets in dimensions.

The title area only allows the selection of a single element where the row and column areas allow for the display of multiple elements.

**INTERACTION - Whiteboard:** Put up a whiteboard and draw a grid. Ask participants to write in where the title, row and column areas are.

## Demo 1: Build a Cube

### Purpose:

The Great Outdoors Company sales plans rely on current price and cost information to forecast profit and margin. The price and cost information will be stored for all products for the two version of the budget. You need to create a cube to hold this data so it can be referenced by other cubes.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

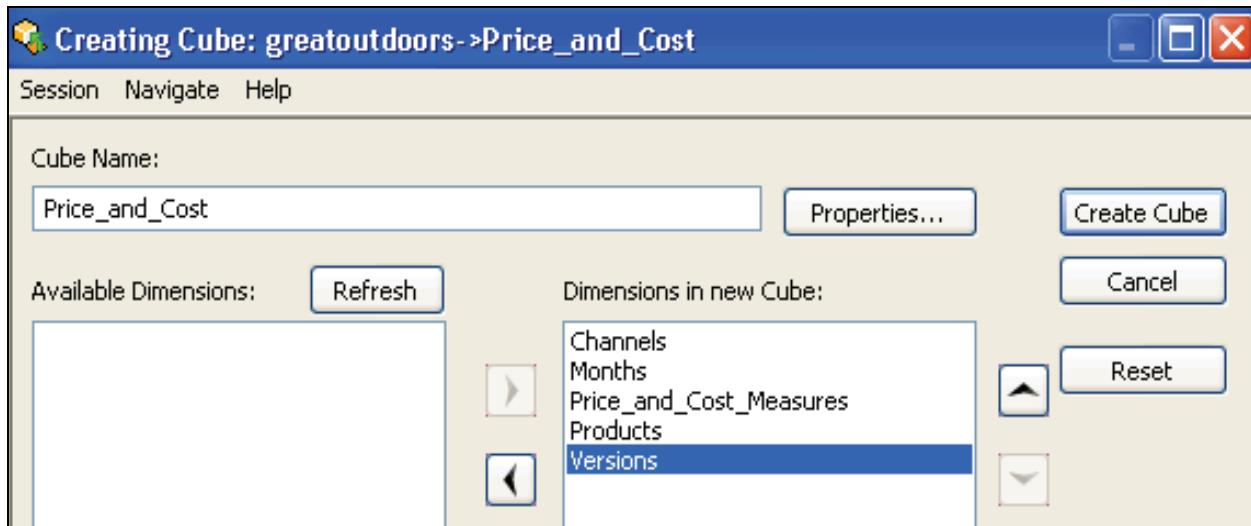
UserName: **admin**

Password: **<blank>**

### **Task 1. Create the Price\_and\_Cost cube.**

1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. Right-click **Cubes** and then click **Create new cube**.
4. In the **Cube Name** box, type **Price\_and\_Cost**.

5. In the **Available Dimensions** list, double-click all the dimensions to move them to the **Dimensions in new Cube** list.
- Channels
  - Months
  - Price\_and\_Cost\_Measures
  - Products
  - Versions



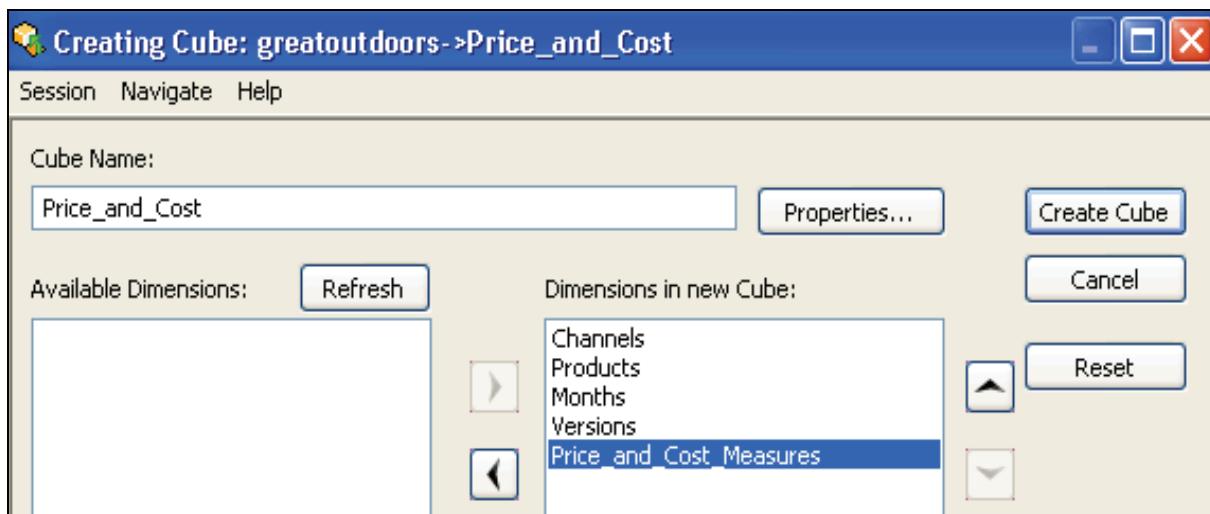
You can use the Right and Left arrows to select and deselect dimensions to appear in the cube.

You can also use the Up and Down arrows to change the order of the dimensions. You need to make the measures dimension last and we will make a guess at ordering the dimensions from smallest/sparse to largest/dense.

6. Click **Products** and then click the **Up** button twice to move **Products** below **Channels**.

7. Click **Price\_and\_Cost\_Measures** and then click the **Down** button to move **Price\_and\_Cost\_Measures** to the bottom.

The results appear as follows:



8. Click **Create Cube**.  
9. Under **Cubes**, expand **Price\_and\_Cost**, and then expand **Dimensions**.

## **Task 2. Arrange the dimensions in Price\_and\_Cost cube.**

1. Double-click the **Price\_and\_Cost** cube.

The cube viewer opens, displaying the dimensions in the title, row and column areas.

2. Drag **Price\_and\_Cost\_Measures** on top of the **Months** dimension ('Total Year') in the title area to swap them.

Price\_and\_Cost\_Measures should display the Unit Sale Price in the drop down list.

---

Remind students that this will be the order in which dimensions appear in the Server Explorer window. This order will not change unless the cube is deleted and recreated.

Once a cube has been created, you may NOT add, delete or rename dimensions. You must delete and recreate the cube if you wish to change the dimensions.

You can use the Cube Optimizer after loading data into a cube to help determine the most efficient dimension order. You right-click the cube name and click Optimize. The lowest percentage is the most efficient. This will reorganize the dimensions to TM1 but it will not physically reorder them (so functions and rules are not impacted).

3. Click the **Months** dimension to open the Subset Editor.

4. Click **All** .

5. Click **Filter by Level** , select **0**, and then click **OK**.

Now only the lowest level elements should appear.

6. From the **Subset** menu, click **Save As**.

7. In the **Select or Enter Subset Name** box, type **Months**, clear the **Private** check box, and then click **OK**.

This saves a subset named Months that can be used in other cubes and views.

8. Click **OK** to close the Subset Editor.

9. Click and drag **Versions** onto **Products (TOTAL PRODUCTS)** dimension to swap them.

10. Click **Recalculate** .

11. Click the Down arrow next to **ALL CHANNELS**, and then click **Golf Shop**.

### Task 3. Save the default view.

1. Click **Recalculate**.

2. Click **File** and then **Save As**.

3. Select **Default**.

This will disable the Select or Enter a Named View list because TM1 will name this view Default. The Default view is automatically opened when you double-click on the cube name.

---

You may create both a public and a private default view. Because we are the admin, the public, default view we create will be available to all users. They may also create their own private, default view that will override the public one.

4. Click **OK**.

The results appear as follows:

Products	Months											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
-- TOTAL PRODUCT	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
+ CAMPING EQL	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
+ GOLF EQUIPM	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
+ MOUNTAINEER	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
+ OUTDOOR PR...	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
+ PERSONAL AC...	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00

5. Close the Cube Viewer.

### Results:

**The Price\_and\_Cost cube is now available to store product price and cost information. It contains a default view for all users to see. You are now ready to load data from an external data source, and use it to forecast margin ratios.**

---

By default, the Cube Viewer displays a view with the last dimension in the column area, the second to last dimension in the row area and all others in the title area.

The Subset Editor will return either the selected element or all displayed elements (if none selected in the display). Be sure not to click on any one element in the display if you wish to return them all.

Data for this cube will be loaded in Module 4 "Load and Maintain Data".

You may also want to right-click the cube name and choose properties. You can specify which dimension represents the time and which dimension represents the measures dimensions. This has no impact on TM1 but if your cubes will be sourcing other applications (such as Framework Manager), they may require this setting.

## What is a Pick List?

- A pick list is a list of valid values for a specific element or cube cell.
- Values entered in cells containing a pick list are validated.
- Values entered in cells using the following methods are not validated:
  - Spread data
  - Turbo Integrator

 Cognos.  
software

When an administrator defines a pick list for an element or a cell, a drop-down menu containing the defined values is available for data entry into the specified cell when browsing a cube in any of the TM1 clients.

The data format of the pick list must match the data type of the element for which it is defined. If the list contains strings, it can only be used for a string element. If it is applied to a string element, it must reside in the last dimension of the cube.

Values in cells containing a pick list are validated meaning a user must select one of the predefined values for the cell. If a user attempts to enter a value that is not valid for the cell, an error appears indicating that only values from the pick list can be entered in the cell.

It is a good idea to apply pick lists to the measures dimension (especially if it is a string).

Please note, pick lists are available to end users when they browse a cube containing the dimension with the Picklist attribute.

Data entered via TI, spreading or after the pick list is defined, will not be 'validated' by the pick list. This means it is possible to see values in cells that do not match the ones in the pick list. This is different from pick lists in Planning

## Creating a Simple Pick List

- There are three types of pick lists:
  - Static
  - Subset
  - Dimension
- Create a text attribute named Picklist

Cognos  
software



There are two ways to create a pick list. A simple pick list is created by creating a new attribute called picklist. This is a simple picklist and it will appear in every cube containing that dimension. Another option is to create an advanced pick list using a rule. A rule may apply to specific elements or can use logic in the rule to determine when the pick list should appear.

A static pick list is composed of a colon-delimited list of values using the syntax, **static:value1:value2:value3:value4**.

A subset pick list contains values corresponding to all elements of a named subset. If the members of the subset change, the values available in the pick list change correspondingly.

A subset pick list is defined using the syntax, **subset:dimension\_name:subset\_name**. A dimension pick list contains values corresponding to all elements of a dimension. If the members of the dimension change, the values available in the pick list change correspondingly.

A dimension pick list is defined using the syntax **dimension:dimension\_name**.

It is good practice to have a blank item as the first item in your static pick list. This allows a user to clear a previous selection. This is also the default display for a subset or dimension pick list in TM1 web. In order to have a blank in the list display in MS Excel, you must define it as a space () .

---

MS Excel cannot support a blank in a pick list. When a blank is defined in the pick list, MS Excel removes the DBRW () function from the cell and no data is returned from the cube. Instead of having nothing in the pick list you need to define a space instead.

### **INTERACTION - Text Chat:**

Ask participants to type in some items that may be useful in a pick list. Some examples may include (dates, colors, etc. ). You can then discuss which would be better as static, subset or dimension based pick lists.

## Demo 2: Create a Pick List

### Purpose:

Create pick lists for the Price\_and\_Cost cube so that users can enter a comment. They have asked to be able to select some standard comments so you will create a pick list of options they can choose from.

TM1 Server: greatoutdoors

TM1 Client: Architect

User/Password: admin/<blank>

### Task 1. Create a string element.

1. In the Server Explorer window, click **Dimensions** to expand.
2. Right-click **Price\_and\_Cost\_Measures** and then click **Edit Dimension Structure**.
3. Click the last element in the list (**Margin%**).
4. Click **Edit** and then click **Insert Element**.
5. In the Insert Element Name box type, **Comment**.
6. In Element Type click **String**, click **Add** and then click **OK**.
7. Click **OK** and then click **Yes** to save changes to the dimension.

### Task 2. Create a static pick list.

1. Right-click **Price\_and\_Cost\_Measures** and then click **Edit Element Attributes**.
2. Click **Edit** and then click **Add New Attribute**.

3. In the **Name** box, type **Picklist**, ensure **Text** is selected and then click **OK**.

	Format (Text)	Picklist (Text)
Unit Sale P	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Unit Cost	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Margin	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Margin%	b:#,##0.00;(#,##0.00)¤CO 2 Y	
Comment		

4. In the cell next to **Comment**, under **Picklist** type the following:

**Static: :High:Medium:Low**

There is a space in before :High.

The syntax to define a subset pick list is:

**static: value1: value2:...valuen**

	Format (Text)	Picklist (Text)
Unit Sale P	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Unit Cost	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Margin	b:(\$#,##0.00;(\$#,###.00)¤C 2 Y \$	
Margin%	b:#,##0.00;(#,##0.00)¤CO 2 Y	
Comment		static: :High:Medium:Low

5. Click **OK**.

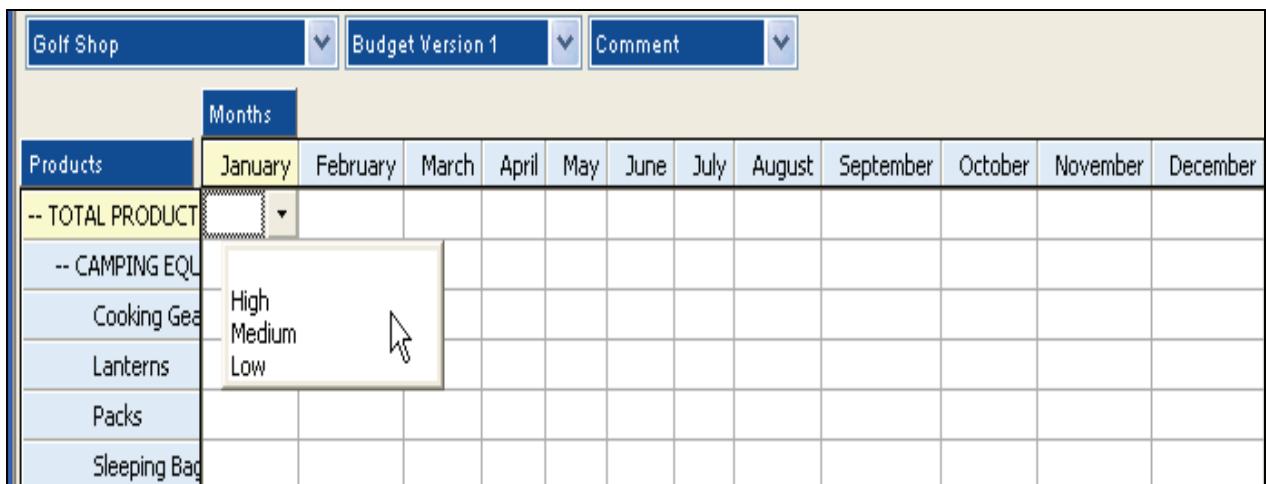
The pick list contains a list of text strings. You created a string element to hold the text once it is selected from the pick list.

String elements must reside in the last dimension in a cube. If the string element is not defined in the last dimension, it is ignored by the cube. If you expand the list of dimensions under the Price\_and\_Cost cube, the Price\_and\_Cost\_Measures dimension appears last.

### Task 3. Create a view to use the pick list.

1. Double-click the **Price\_and\_Cost** cube to open it in the Cube Viewer.
2. Double-click **Price\_and\_Cost\_Measures (Unit Sale Price)** to open in the Subset Editor.
3. Click **All** and then click **OK** to close the Subset Editor.
4. In the Price\_and\_Cost\_Measures drop down list, click **Comment** and then click **Recalculate**.
5. Click **Months**, click **Use Alias**  (be sure Select Alias displays Full Name) and then click **OK**.
6. Click **Recalculate**.
7. Click **File**, click **Save As** and name the **private** view **Price Comment**.
8. Click to expand **TOTAL PRODUCTS** and **CAMPING EQUIPMENT** and then click in the first cell under **January**.

The results appear as follows:

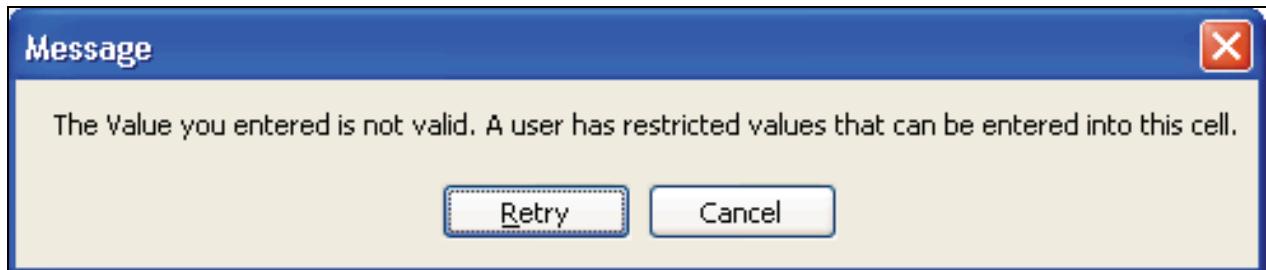


The screenshot shows the TM1 Cube Viewer interface. At the top, there are three dropdown menus: 'Golf Shop', 'Budget Version 1', and 'Comment'. Below these is a toolbar with a 'Months' button, which is currently selected and highlighted in blue. The main area is a grid table. The columns represent the months from January to December. The rows are categorized by product type: 'Products', 'TOTAL PRODUCT', 'CAMPING EQL', 'Cooking Gear', 'Lanterns', 'Packs', and 'Sleeping Bag'. The 'CAMPING EQL' row is expanded, showing three categories: 'Cooking Gear', 'Lanterns', and 'Packs'. The 'Lanterns' cell for January has a dropdown menu open, displaying three options: 'High', 'Medium', and 'Low'. A cursor arrow is pointing at the 'Medium' option. The rest of the grid is empty, with most cells being light gray.

A drop-down list appears in the cell. Click on any of the selections to populate the cell.

9. Click the cell again, type **Jan** and press **Enter**.

An error is generated because the value typed did not match any of the values from the pick list. The pick list validates items that are entered into a cell containing a pick list.



10. Click **Retry**.

11. Click the pick list again and then click **High**.

The results appear as follows:

Golf Shop	▼	Budget Version 1	▼	Comment	▼															
	Months																			
Products	January	February	March	April	May	June	July	August	September	October	November	December								
-- TOTAL PRODUCT	High																			
-- CAMPING EQL																				
Cooking Gear																				
Lanterns																				
Packs																				
Sleeping Bag																				
Tents																				
+ GOLF EQUIPM																				
+ MOUNTAINEER																				
+ OUTDOOR PR																				
+ PERSONAL AC																				

The pick list was defined for the Comment element in the Price\_and\_Cost\_Measures dimension. This is the only element that will generate the picklist.

12. Close the **Price Comment** view without saving changes.

### Results:

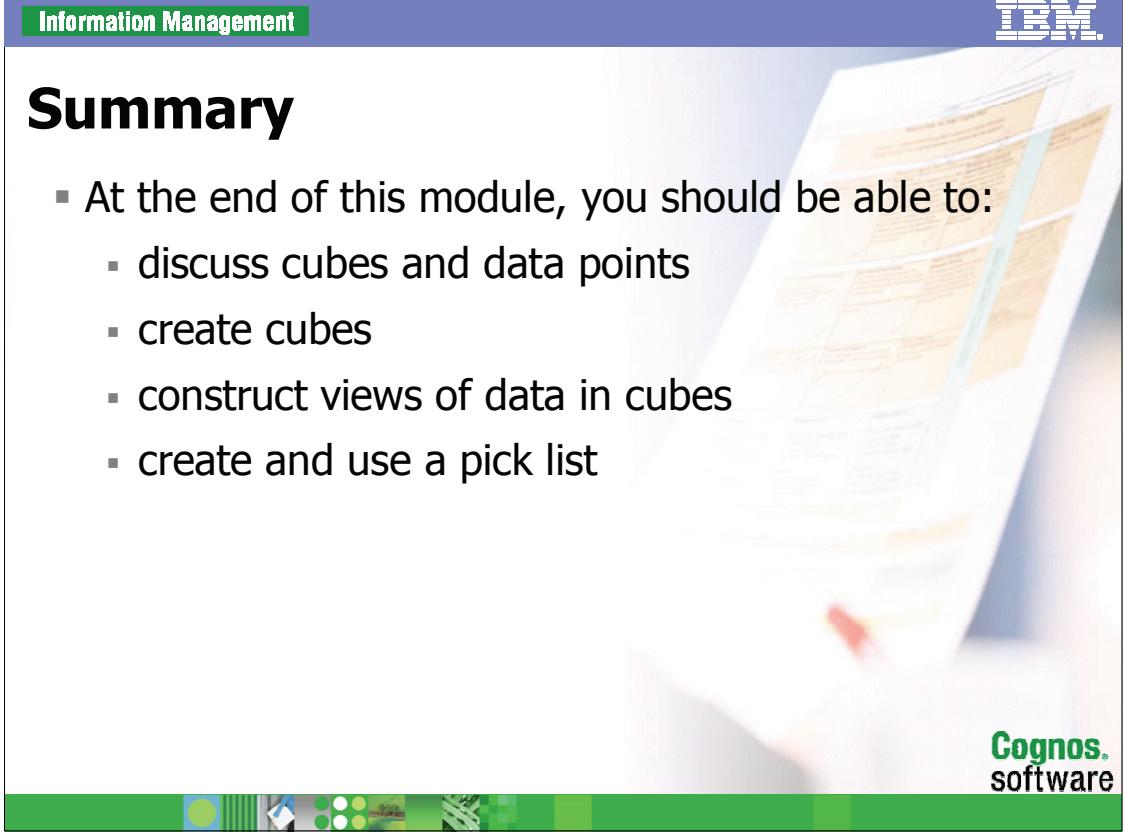
**You have created pick lists to enter the price level for different products. You also a string element in which to display the result of the pick list.**

---

You will create additional pick lists later in the course. This is simply an introduction.

## Summary

- At the end of this module, you should be able to:
  - discuss cubes and data points
  - create cubes
  - construct views of data in cubes
  - create and use a pick list



Cognos  
software



## Workshop 1:Build the Sales Plan Cube

To forecast future sales based on past performance, the Great Outdoors model requires a cube that will store sales data based on several dimensions. Currently, Great Outdoors products are sold through several channels, in several world wide subsidiaries. The cube must present sales over time for two separate versions of the budget, and should include details about unit cost, price, and quantity sold. Users would also like to add some information about the peak seasons for different products.

To accomplish this:

- Create two new dimensions, using the comma-delimited, text files provided.
  - One contains information on subsidiaries.
  - One contains sales metrics or measures such as Quantity and Unit Sale Price.
- Create the Sales\_Plan cube so it includes subsidiaries, channels, products, months, versions, and Sales\_Plan\_Measures.
- Re-orient the cube so that Sales\_Plan metrics are on rows, periods are on the columns, and the title shows data for lanterns, all channels, and all subsidiaries.
- Save a default view.
- Create a subset pick list for the Sales Plan Measures that identifies the peak season for sales using the Months subset for the Months dimension and format values.

For more detailed information outlined as tasks, see the Task Table section.

For the final query results, see the Workshop Results section that follows the Task Table section.

## Workshop 1: Task Table

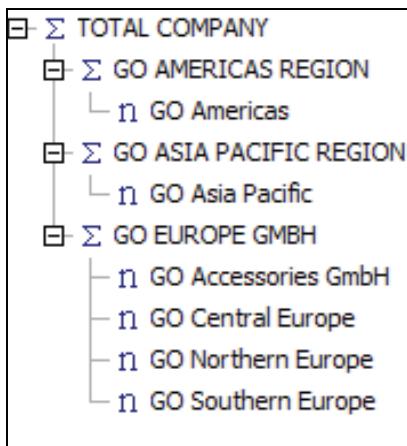
Task	Where to Work	Hints
1. Create the Subsidiaries dimension.	Turbo Integrator	<ul style="list-style-type: none"> <li>Import from Text file.</li> <li>C:\Edcognos\P6502\SourceFiles\Subsidiaries.csv.</li> </ul>
2. Create the Sales_Plan_Measures dimension.	Dimension Editor	<ul style="list-style-type: none"> <li>Copy and paste cells A1 - A13 from ASCII file.</li> <li>C:\Edcognos\P6502\SourceFiles\Sales Metrics.csv.</li> <li>Add a Seasons string element.</li> <li>Save as Sales_Plan_Measures.</li> </ul>
3. Create the Sales_Plan cube.	Cubes/New Cube	<ul style="list-style-type: none"> <li>Subsidiaries.</li> <li>Channels.</li> <li>Products.</li> <li>Months.</li> <li>Versions.</li> <li>Sales_Plan_Measures.</li> </ul>

Task	Where to Work	Hints
4. Create a default view of the Sales_Plan cube.	Cube Viewer	<ul style="list-style-type: none"> <li>● Swap Sales_Plan_Measures and Versions.</li> <li>● Swap Months and Versions.</li> <li>● Select Months subset.</li> <li>● Select data for lanterns.</li> <li>● Select Budget Version 1</li> <li>● Recalculate the view.</li> <li>● Save it as a default view.</li> </ul>
5. Create a subset pick list for the Sales Plan Measures.	Attribute Editor	<ul style="list-style-type: none"> <li>● Create the picklist attribute.</li> <li>● Subset: Months: Months</li> <li>● Format values</li> </ul>

If you need more information to complete a task, see the Step-by-Step instructions at the end of the Workshop.

## Workshop 1: Workshop Results

Task 1: The Subsidiaries dimension appears shown below:



Task 2: The Sales\_Plan\_Measures dimension appears shown below:

- Quantity
- Unit Sale Price
- GROSS SALES REVENUE
- Returns and Allowances
- Cash Discount
- Volume Discount
- TOTAL DISCOUNTS AND ALLOWANCES
- Freight
- NET SALES REVENUE
- Unit Cost
- COST OF SALES
- GROSS MARGIN
- GROSS MARGIN %
- Seasons

Task 4: The Sales\_Plan default view appears shown below:

TOTAL COMPANY		ALL CHANNELS		Lanterns		Budget Version 1								
								Months						
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
Quantity		0	0	0	0	0	0	0	0	0	0	0	0	
Unit Sale Price		0	0	0	0	0	0	0	0	0	0	0	0	
GROSS SALES REV		0	0	0	0	0	0	0	0	0	0	0	0	
Returns and Allow.		0	0	0	0	0	0	0	0	0	0	0	0	
Cash Discount		0	0	0	0	0	0	0	0	0	0	0	0	
Volume Discount		0	0	0	0	0	0	0	0	0	0	0	0	
TOTAL DISCOUNT		0	0	0	0	0	0	0	0	0	0	0	0	
Freight		0	0	0	0	0	0	0	0	0	0	0	0	
NET SALES REVENUE		0	0	0	0	0	0	0	0	0	0	0	0	
Unit Cost		0	0	0	0	0	0	0	0	0	0	0	0	
COST OF SALES		0	0	0	0	0	0	0	0	0	0	0	0	
GROSS MARGIN		0	0	0	0	0	0	0	0	0	0	0	0	
GROSS MARGIN %		0	0	0	0	0	0	0	0	0	0	0	0	
Seasons														

Task 5: The attributes for Sales\_Plan\_Measures are shown below:

	Picklist (Text)	Format (Text)
Quantity		b:#,###0.00;(#,##
Unit Sale Price		b:(\$#,###0.00;(\$
GROSS SALES REVENUE		b:(\$#,###0.00;(\$
Returns and Allowances		b:(\$#,###0.00;(\$
Cash Discount		b:(\$#,###0.00;(\$
Volume Discount		b:(\$#,###0.00;(\$
TOTAL DISCOUNTS AND		b:(\$#,###0.00;(\$
Freight		b:(\$#,###0.00;(\$
NET SALES REVENUE		b:(\$#,###0.00;(\$
Unit Cost		b:(\$#,###0.00;(\$
COST OF SALES		b:(\$#,###0.00;(\$
GROSS MARGIN		b:(\$#,###0.00;(\$
GROSS MARGIN %		[REDACTED]
Seasons	Subset: Months:Months	

# Workshop 1: Step-by-Step Instructions

## Task 1. Create the Subsidiaries dimension.

1. In the Server Explorer window, right-click **Processes**, click **Create New Process**, and then click **Text**.
2. Click **Browse** and navigate to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles** and then double-click **Subsidiaries.csv** to select it.
3. Click **OK**, click **Preview**, and then click the **Variables** tab.
4. Click **V1**, type **vSubsidiary**, and then under **Contents**, click **Element**.
5. Click **V2**, type **vSubRegion**, and then under **Contents**, click **Consolidation**.
6. Click **V3**, type **vTotal**, and then under **Contents**, click **Consolidation**.
7. Click the **Maps** tab and then click the **Dimensions** tab.
8. In the **Dimension** box, type **Subsidiaries**.
9. Click **By Input**, select **Automatic, Hierarchy**, and then click **OK**.
10. Click the **Consolidations** tab, in the **vSubRegion** row, click the arrow under **Child Variable**.
11. Click **vSubsidiary**, and then click **OK**.
12. In the **vTotal** row, click the arrow under **Child Variable**, click **vSubRegion**, and then click **OK**.
13. For each variable, click **By Input**, select **Automatic, Name** and then click **OK**.
14. Click the **Advanced** tab, and then click the **Prolog, Metadata, Data** and **Epilog** tabs.
15. Click **Save**, in the name box, type **CreateSubsidiariesDimCSV** and then click **OK**.
16. Click **Run**, click **OK**, and then close Turbo Integrator.

## Task 2. Create the Sales\_Plan\_Measures dimension.

1. In the Server Explorer, right-click **Dimensions** then select **Create New Dimension**.

Since this dimension is flat, it is easier to create it manually.
2. In **Microsoft Excel**, click **File** and then click **Open**.
3. Navigate to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles** and then double-click **Sales Metrics.csv** to open it.
4. Select cells **A1-A13**, and then press **Ctrl+C** to copy.
5. In the **Dimension Editor**, click **Edit**, and then click **Paste**.
6. Click **GROSS MARGIN %**, click **Edit** and then click **Insert Element**.
7. Name the element **Seasons**, click **String**, click **Add**, and then click **OK**.
8. From the **Dimension** menu, click **Save As**.
9. In the **Name** box, type **Sales\_Plan\_Measures**, and then click **OK**.
10. Click **OK** again to close the **Dimension Editor**.

Sales\_Plan\_Measures should appear under the Dimensions icon.

## Task 3. Create the Sales\_Plan cube.

1. In the **Server Explorer** window, right-click **Cubes** and then click **Create new cube**.
2. In the **Cube Name** box, type **Sales\_Plan**.
3. Double-click each of the following dimensions in the **Available Dimensions** list:
  - **Subsidiaries, Channels, Products, Months, Versions, Sales\_Plan\_Measures**

Use the Up and Down arrows to arrange the dimensions in this order if needed.

4. Click **Create Cube**.

The Sales\_Plan cube will appear below the Cubes icon.

## Task 4. Create a default view of the Sales\_Plan cube.

1. In Server Explorer, double-click the **Sales\_Plan** cube.
2. Click **Sales\_Plan\_Measures** (in column area) and drag onto **Versions** (in row area) to swap dimensions.
3. Click **Versions** (in the column area) and drag on top of the **Months** dimension (in the title area) to swap them.
4. Click **Months** (in the column area) to open the Subset Editor.
5. Click the Down arrow beside the Select Subset list and then click **Months**.  
You should now see a list of months.
6. Click **OK** to close the Subset Editor.



7. Click **Recalculate**.

The months should appear across the columns in the view. If not, repeat steps 4-7 and do not click on any single month before click OK in step 6.

8. Double-click **TOTAL PRODUCTS** (in the title area) to open the Subset Editor.
9. Expand **CAMPING EQUIPMENT**, click **Lanterns**, and then click **OK**.
10. Click **Recalculate**.
11. From the **File** menu, click **Save As**.
12. Select **Default** and then click **OK**.

Leave the default view open.

## Task 5. Create a subset pick list for the Sales\_Plan\_Measures.

1. In Server Explorer, right-click the **Sales\_Plan\_Measures** dimension and then click **Edit Element Attributes**.
2. Click **Edit** and then click **Add New Attribute**.
3. In the Name box type, **Picklist** and then click **OK** (text should be selected by default).
4. In the Picklist column, next to Seasons type:  
**Subset: Months:Months**
5. Under the **Format** column, click **Unit Sale Price** and drag down to **Gross Margin**.
6. Click **Format**.
7. Click **Currency** and then click **OK**.
8. Under the **Format** column, click **Quantity**, click **Format**, click **Comma** and then click **OK**.
9. Click **OK** to close the editor.
10. Click the **default** view of the Sales\_Plan cube and then click **Recalculate**.
11. Right-click next to **Seasons** to see the pick list of months.
12. Close the cube viewer, saving changes.
13. Close TM1 Architect, close MS Excel (without saving changes) and then close the GreatOutdoors server (click Yes to save changes).

**Information Management**



## Load and Maintain Data

IBM Cognos TM1 9.5



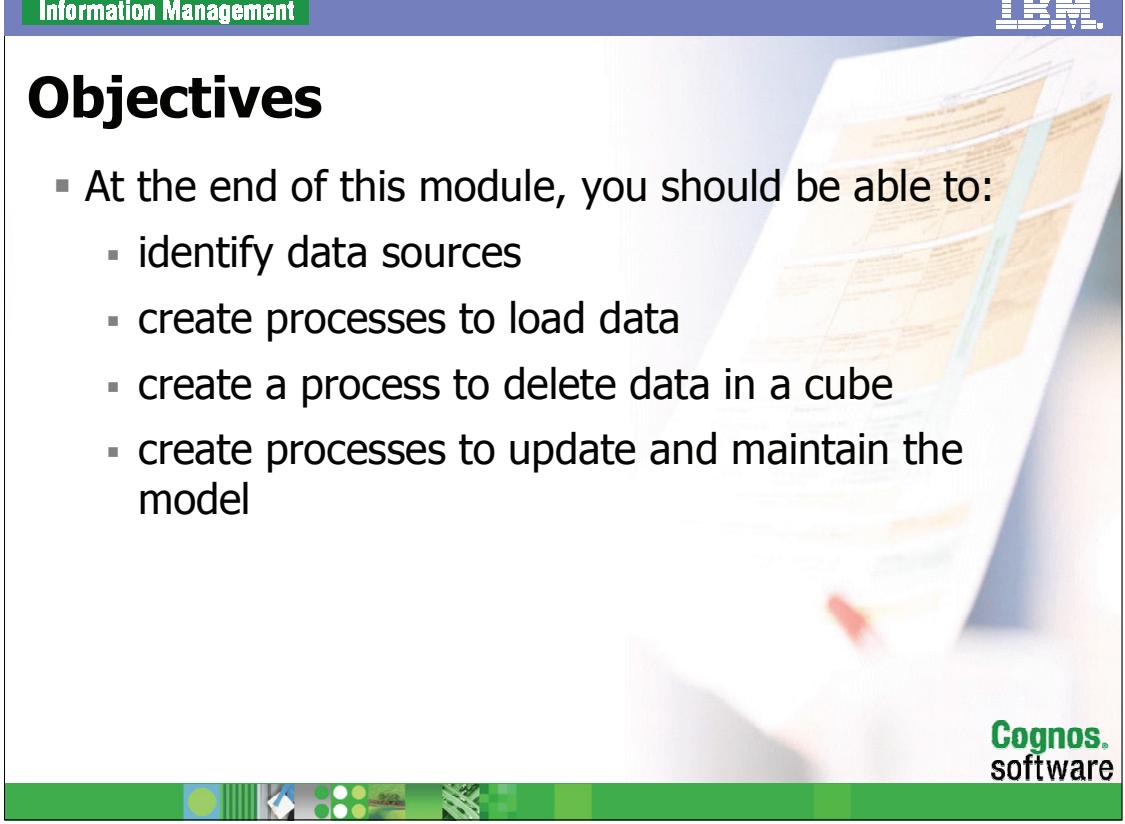
**Cognos.**  
software

© 2009 IBM Corporation



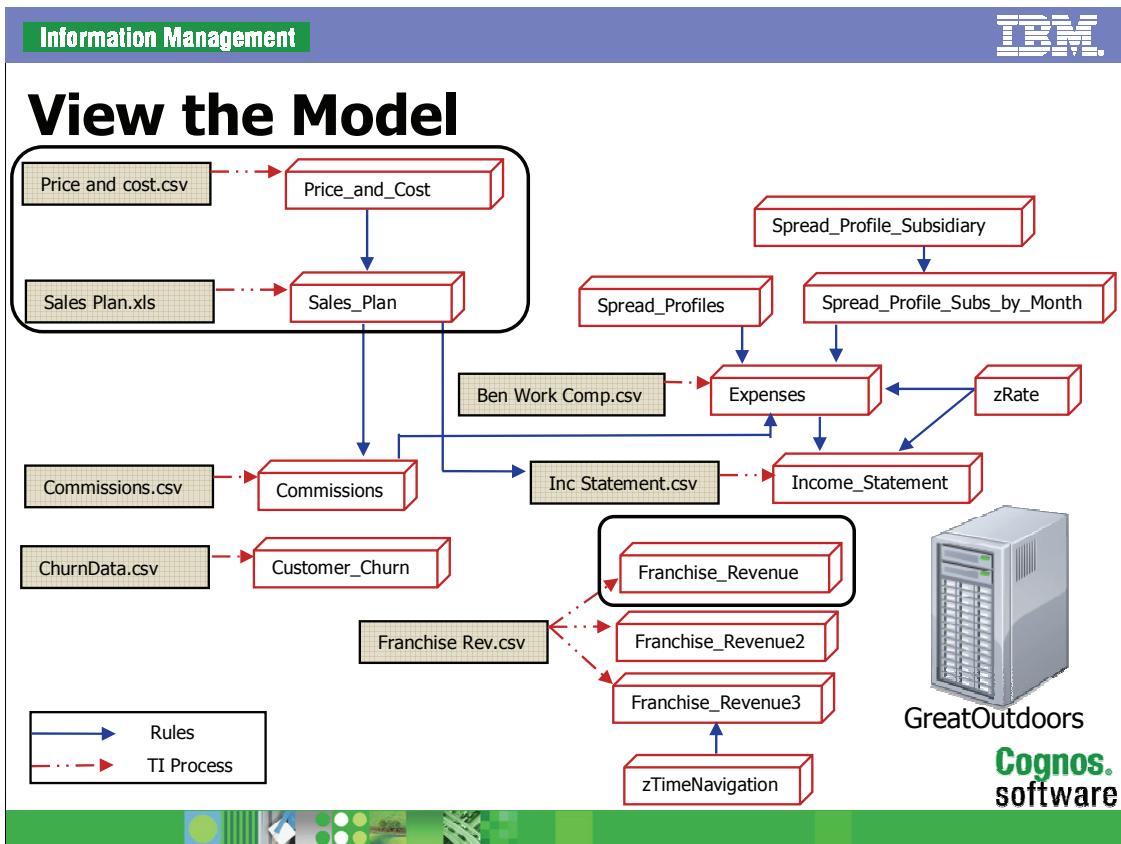
# Objectives

- At the end of this module, you should be able to:
  - identify data sources
  - create processes to load data
  - create a process to delete data in a cube
  - create processes to update and maintain the model



Cognos.  
software





In this module we will use Turbo Integrator to load data to the following cubes:

- Price\_and\_Cost
- Sales\_Plan
- Franchise\_Revenue

## Determine the Data Source

- Turbo Integrator (TI) can import data from:
  - ODBC sources
  - Text files
  - TM1 Views and Dimensions
  - ODBO (used with Microsoft Analysis Services)
  - SAP via RFC
- Data may be entered manually:
  - Type in cells
  - Spreading
- Processing Worksheets

 Cognos.  
software

You must establish connections to data sources such as:

- ODBC
- ODBO
- SAP

---

You should contact your database administrator for connections to other databases. You may need additional software or passwords in order to access these sources.

A processing worksheet is a specially formatted Microsoft Excel spreadsheet that may contain data to load to a cube. For more information on processing worksheets, please refer to the TM1 Developer Guide and section 5 "Importing Data with Processing Worksheets".

## Review the Data Structure

- Compare source data to desired data
- Data is loaded at the leaf (element) level cells
- An element must be specified for every dimension
- Create variables for:
  - calculations
  - functions
  - clean data
  - format data



Cognos.  
software

Data is loaded by specifying the leaf elements name or alias.

**INTERACTION - Whiteboard:** Put up a whiteboard and ask people to choose a color text and list some of the data source they expect to be loading data from. Discuss some sources and any issues that may arise.

## Load Data with Turbo Integrator

- To transfer data into a cube, perform the following steps:
  1. Connect to the source and preview data.
  2. Create and map variable contents.
  3. Identify cube, dimensions and measures.
  4. Save and execute process.



Cognos.  
software

TurboIntegrator is neither case sensitive nor space sensitive.

---

Variable names are the only places where spaces may not be used.

## Demo 1: Import Data from an ASCII File

### Purpose:

You have just received a file that contains the price and cost information for each of your products. You will need to populate the Price\_and\_Cost cube with this data so that it can be used to determine Gross Sales Revenue in another cube.

### Task 1. Specify the data source.

TM1 Server: **greatoutdoors**

TM1 - Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Start the **greatoutdoors TM1 server** and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. In the Server Explorer window, right-click **Processes** and click **Create New Process**, and then click **Text**.
4. Beside the Data Source Name box, click **Browse**, navigate to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles**, double-click **Price and Cost.csv** to open the file, and then click **OK** to dismiss the warning.

If you were working on a remote server, you would use a Universal Naming Convention (UNC) path for files.

5. In the **Number of title records** box, type **1**.

This indicates that the first row in the Text file contains descriptive information about the data.

6. Click **Preview**.

You can see the first 10 rows of data in your source.

## Task 2. Create and map variables.

1. Click the **Variables** tab.
2. Double-click **V1**, type **vProduct**, and then in the **Contents** column, click **Element**.  
If the column name is a reserved word to TI or contains a space, it will be replaced by Vx.
3. Double-click **V4**, type **vMonth**, in the **Contents** column, click **Element**.
4. Double-click **V5**, type **vCost**, in the **Contents** column, click **Data**.
5. Double-click **V6**, type **vPrice**, in the **Contents** column, click **Data**.
6. Next to **Version**, under **Content**, click **Element**.
7. Next to **Channel**, under **Content**, click **Element**.

This cube contains five dimensions including the measures dimension. You should have an element from each dimension in your variable list. In this case, Data is a type of element, so you do not need to map an element to the measures dimension. You may need to map a separate element when you only have one measure element and then you won't have to map it on a later tab.

## Task 3. Identify Cube, Dimensions and Measures.

1. Click the **Maps** tab.
2. Select **Update Cube**.

Cube Action determines what TI will do to a cube. It can:

- Create a cube
- Recreate a cube (destroy it and then create it)
- Update the cube
- Take no action

The Price\_and\_Cost cube already exists so we are going to update it.

It is good practice to place a v next to custom variable names to identify them as such in scripts. It is also recommended to use names that will help identify the data contents. This will help when mapping the variables to dimensions and elements.

3. In the list next to CubeName, select **Price\_and\_Cost**.
4. Click the **Dimensions** tab.
5. In the **Dimension** column, click the Dimension that corresponds to each variable in the list:
  - **vProduct: Products**
  - **Version: Versions**
  - **Channel: Channels**
  - **vMonth: click Months**
  - **(Data Variables): Price\_and\_Cost\_Measures**

Only change the Element Type if you are loading String (s) data into the cube, otherwise leave it as Numeric (n).

You are not updating or creating any dimensions so you leave the Action column set to AsIs.

6. Click the **Data** tab.
7. Next to **vCost**, under Element, click the arrow to open the Subset Editor.
8. Select **Unit Cost** and then click **OK** to close.
9. Next to **vPrice**, under Element, click the arrow to open the Subset Editor.
10. Select **Unit Sale Price** and then click **OK** to close.

## Task 4. Save and Execute Process.

1. Click the **Advanced** tab.
2. Click the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.

When you click on each of the tabs, you regenerate the scripts that are associated with the tab.

You will also notice there are statements regarding logging changes on the Prolog and Epilog tabs. When a cube is created by default any changes made to it are written to a log file. However, when data is loaded to a cube, there may be a lot of records going into a single cube. TM1 will write each record to the cube (one or more times depending on the script) but if logging is on, it will also write each record to the log file. This may cause the data load to take a very long time and it could also use a lot of space on your server (especially if the log files are not removed).

For these reasons, the default when loading data into a cube is to turn off logging. It can be turned back on (usually for troubleshooting purposes) by checking the Log Changes box on the Cubes tab.

The script on the Prolog tab looks at the current Log changes setting, records it and then turns logging off. On the Epilog tab, the Log changes setting is set back to the state it was in prior to the Prolog changes.

3. Click **Save**.
4. In the **Name** box, type **LoadPriceAndCostCSV** and then click **OK**.
5. Click **Run**.
6. Click **OK** when the process has completed, and then close Turbo Integrator.

7. In the Server Explorer window, double-click the **Price\_and\_Cost** cube.

The cube appears in the Cube Viewer as shown below:

Golf Shop		Budget Version 1	Unit Sale Price				
Products	Months	Jan	Feb	Mar	Apr	May	Jun
-- TOTAL PRODUCT	\$3,497.35	\$3,497.35	\$3,497.35	\$3,497.35	\$3,497.35	\$3,497.35	\$3,497.35
+ CAMPING EQL	\$802.04	\$802.04	\$802.04	\$802.04	\$802.04	\$802.04	\$802.04
+ GOLF EQUIPM	\$1,724.56	\$1,724.56	\$1,724.56	\$1,724.56	\$1,724.56	\$1,724.56	\$1,724.56
+ MOUNTAINEER	\$454.08	\$454.08	\$454.08	\$454.08	\$454.08	\$454.08	\$454.08
+ OUTDOOR PR	\$22.29	\$22.29	\$22.29	\$22.29	\$22.29	\$22.29	\$22.29
+ PERSONAL AC	\$494.38	\$494.38	\$494.38	\$494.38	\$494.38	\$494.38	\$494.38

8. Close the Cube Viewer.

### Results:

You created the process to load data from a comma separated value file. In it you specified which columns contained dimension elements and which contained data elements. You then saved and executed the process.

# Using Formulas in Variables

- Variables contain formulas for:
  - converting data types
  - placeholders for data not represented in the source
  - combine elements
  - extract only portions of data for an element



Cognos.  
software

Many functions may be used in both Turbo Integrator and rules.

---

Open the TM1 Function Reference in Help and show the list of TI functions

**INTERACTION - Whiteboard:** Have participants open the Function Reference guide and pick out some functions they think they may need and list them on the Whiteboard.

## Demo 2: Import Data from an ODBC Source

### Purpose:

You want to import data from a Microsoft Excel spreadsheet used as an ODBC source. However, there is no Month column in the spreadsheet, but there is a Date column.

You will create an ODBC data source name that identifies the spreadsheet. Then, you will create an alias in the Months dimension that refers to each month by number. Next, you will create a process that uses a function to convert the Date into the month number, as well as loads the data from the spreadsheet. Finally, you will run the process and confirm that the data has been loaded successfully.

### Task 1. View data and decide how to proceed.

TM1 Server: **greatoutdoors**

TM1 - Architect: Server Explorer

UserName: **admin**

Password: **<blank>**

1. Open Microsoft Excel.
2. Click **File**, click **Open** and browse to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles**, and then double-click **Sales plan2.xls**.

Notice that the first row contains descriptive information about the data. There are 11 columns in the Excel file.

Notice the format of the Order Date column. It displays the data as numbers. You will need to load the dates using the month number. However, currently, the month elements are all based on the month name. You need to add an alias to the Months dimension to number each month from 1 to 12.

Data may be loaded if it matches either the element name or an alias.

3. In the **Name** box, beside the Function bar, click the down arrow, and then click **Sales**.

The named range Sales, refers to the contents of the entire spreadsheet. You will use it as the database name in Turbo Integrator.

4. Close **Sales plan2.xls**.

If you do not close the CSV file you will get an error when you try to preview the data in Turbo Integrator, later in this demo.

## **Task 2. Create aliases for month numbers.**

1. In **Server Explorer**, right-click the **Months** dimension and then select **Edit Element Attributes**.

You will create an attribute for the number of each month. This will allow you to use a TI function to extract the month number from the Order Date and then map it to the Alias instead of the element name. TM1 allows you to use either the Element Name or the Element Alias to load data.

2. From the **Edit** menu, click **Add New Attribute**.
3. In the **Name** box, type **MonthNumber**, type click **Alias** and then click **OK**.  
A warning appears.
4. Click **OK**.

5. Type the numbers from **1** to **12**, below the **MonthNumber (Alias)** column.  
 The result appears as shown below:

	MonthNumber (Alias)
Total Year	Total Year
Q1	Q1
Jan	1
Feb	2
Mar	3
Q2	Q2
Apr	4
May	5
Jun	6
Q3	Q3
Jul	7
Aug	8
Sep	9
Q4	Q4
Oct	10
Nov	11
Dec	12

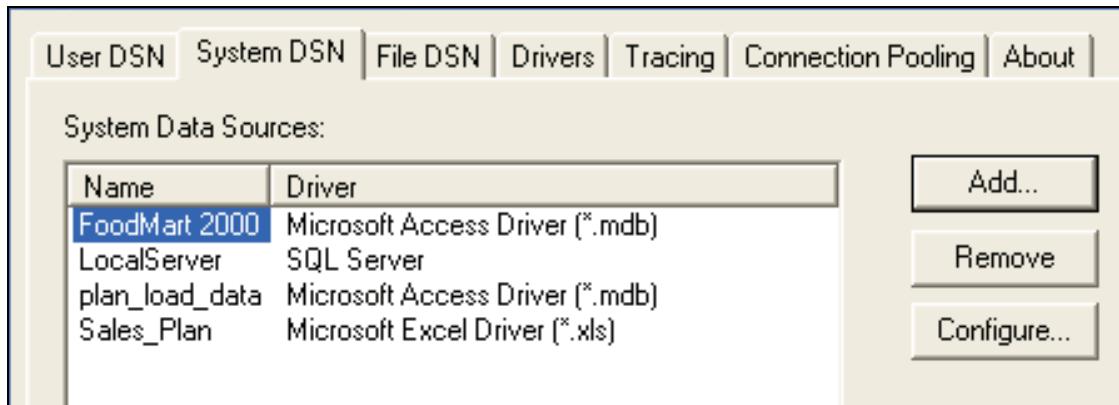
6. Click **OK**.

### Task 3. Add a connection to an ODBC driver.

1. From the **Start** menu, click **Control Panel**, and then click **Administrative Tools**.
2. Double-click **Data Sources (ODBC)**, and then click the **System DSN** tab.
3. Click **Add**, and then double-click **Microsoft Excel Driver (\*.xls)**.
4. On the ODBC Microsoft Excel Setup page, click **Select Workbook**, and then browse to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles**, click **Sales plan2.xls**, and then click **OK**.

5. In the **Data Source Name** box, type **Sales\_Plan**, and then click **OK**.

The Sales\_Plan ODBC connection appears in the System Data Sources window, as shown below:



6. Click **OK**, to close the ODBC Data Source Administrator, and then close the Administrative Tools window.

## Task 4. Specify the ODBC data source.

1. In the Server Explorer window, right-click **Processes** and click **Create New Process**.
2. Click **ODBC**.
3. Beside the **Data Source Name** box, click **Browse**, and then double-click **Sales\_Plan**.

This is the data source name that you created using the ODBC Administrator.

4. In the Query box type **select \* from sales**.

The query is not case sensitive, so you can type the select statement in lower case. 'Sales' refers to the named range in Microsoft Excel.

## 5. Click Preview.

The result appears as shown below:

Subsidiaries	Channel	Product Type	Version	Sales Staff	Order Date	Quantity	Cash Discount	Volume
GO Americas	Department Store	Binoculars	Budget version	Barros, Viviam	2008-01-07 00:00	1969.000000	7706.610000	5024.12
GO Americas	Department Store	Binoculars	Budget version	Barros, Viviam	2008-10-07 00:00	4623.000000	18097.700000	11798.1
GO Americas	Department Store	Binoculars	Budget version	Barros, Viviam	2008-09-07 00:00	2522.000000	9871.490000	6435.37
GO Americas	Department Store	Binoculars	Budget version	Bretton, Frank	2008-12-07 00:00	509.000000	1991.570000	1298.33
GO Americas	Department Store	Binoculars	Budget version	Chow, Donald	2008-10-07 00:00	1792.000000	7013.920000	4572.54
GO Americas	Department Store	Binoculars	Budget version	Orozco, Ana	2008-06-07 00:00	553.000000	2164.770000	1411.26
GO Americas	Department Store	Binoculars	Budget version	Orozco, Ana	2008-10-07 00:00	310.000000	1212.290000	790.350
GO Americas	Department Store	Binoculars	Budget version	Pierce, Samantha	2008-07-07 00:00	575.000000	2251.370000	1467.66
GO Americas	Department Store	Binoculars	Budget version	Rojas, Gabriela	2008-04-07 00:00	221.000000	865.890000	564.500
GO Americas	Department Store	Binoculars	Budget version	Rojas, Gabriela	2008-11-07 00:00	575.000000	2251.370000	1467.66

## Task 5. Create variables.

- Click the **Variables** tab.

You need to name all variables with default names that do not identify the contents of the variable. You also need to identify how the content will be used.

- Rename **V3** as **vProducts**, and then in the **Contents** column click **Element**.

3. Repeat step 2, with the following information:

Old Variable Name	New Variable Name	Contents
Subsidiaries	unchanged	Element
Channel	unchanged	Element
Version	unchanged	Element
V5	vStaff	Ignore
V6	vDate	Other
Quantity	unchanged	Data
V8	vCashDiscount	Data
V9	vVolumeDiscount	Data
Freight	unchanged	Data
V11	vReturnsAndAllowances	Data

A variable whose Contents is set to Other is available to a TI process but not required to be mapped to an Element, Consolidation or Attribute. Variables set to Ignore are unavailable to the process at all.

Next you want to create a new variable called vMonth.

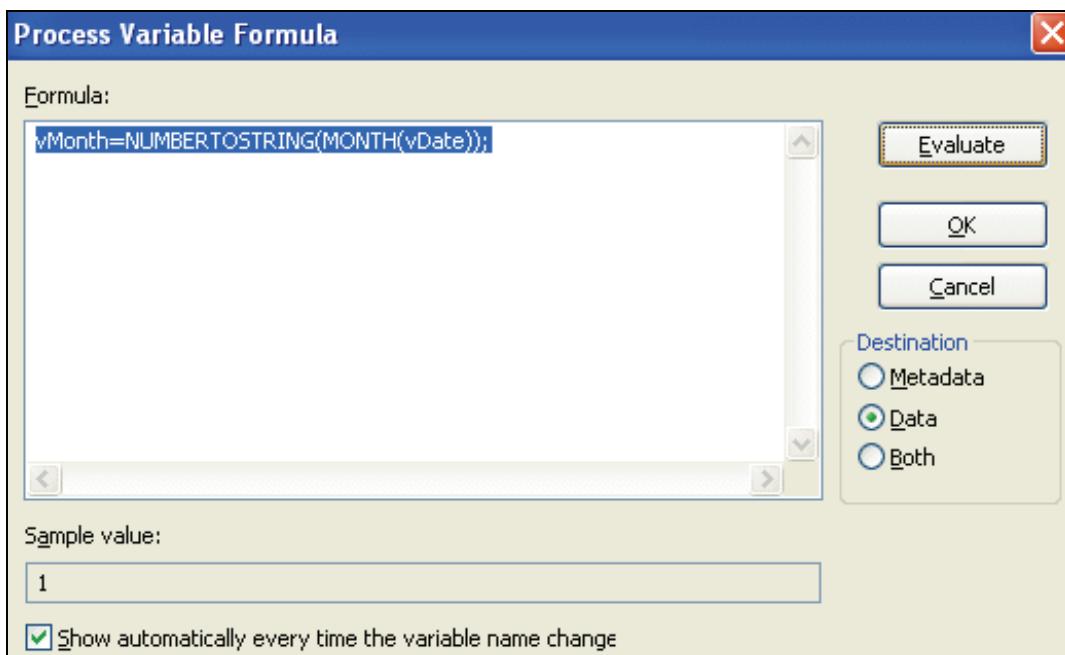
4. Click **New Variable**, and rename it **vMonth**

You want to convert the vMonth variable with the number that corresponds to the MonthNumber alias. You will use the MONTH function to retrieve the month from the variable and the NUMBERTOSTRING function to convert the number to text (again, to match the MonthNumber alias).

#### Syntax:

MONTH (date); - where date is in the format YY-MM-DD or YYYY-MM-DD  
 NUMBERTOSTRING(value); - where value is a real value

5. Click **Formula**, and in the **Formula** box, type:  
**vMonth=NUMBERTOSTRING(MONTH(vDate));**  
 Remember the ; (semicolon) at the end.
6. Below Destination, click **Data**, and then click **Evaluate**.  
 The result appears as shown below:



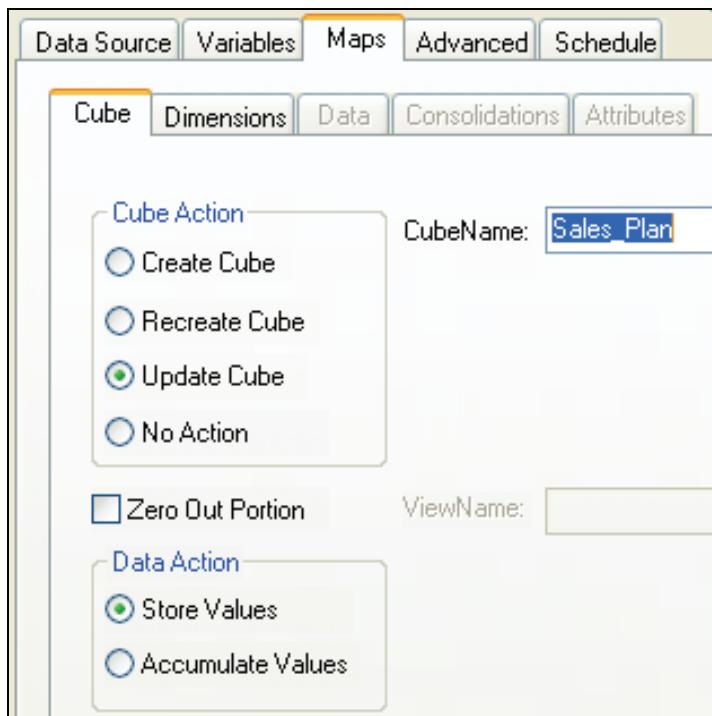
7. Click **OK**.
8. For **vMonth**, in the **Variable Type** column, click **String**, and then in the **Contents** column click **Element**.

Variables that are mapped to Element or Consolidation must be of type string. Next you will map the variables to TM1 objects like cubes, elements, consolidations and attributes. You will also determine what will be done with objects (for example, creating, updating, or recreating).

## Task 6. Map cube, dimension and data information.

1. Click the **Maps** tab, and then on the **Cube** tab, click **Update Cube**.
2. Click the down arrow beside **CubeName**, and then select **Sales\_Plan**.
3. Below **Data Action**, click **Store Values**.

The result appears as shown below:



You chose Update Cube, because the cube was created previously.

4. Click the **Dimensions** tab.

On this tab, you map the data variable to the dimension in TM1. Since the dimensions have already been created, the Action column remains "AsIs".

---

It is recommended that you use capital letters for functions. This way you immediately know the MONTH is a function and not a variable or data.

5. Beside each Element Variable, click the appropriate Dimension.

Element Variable	Dimension
Subsidiaries	<b>Subsidiaries</b>
Channel	<b>Channels</b>
vProducts	<b>Products</b>
Version	<b>Versions</b>
vMonth	<b>Months</b>
(Data Variables)	<b>Sales_Plan_Measures</b>

6. Click the **Data** tab.

7. Beside each Data Variable, click the appropriate Element.

Data Variable	Element
Quantity	<b>Quantity</b>
vCashDiscount	<b>Cash Discount</b>
vVolumeDiscount	<b>Volume Discount</b>
Freight	<b>Freight</b>
vReturnsAndAllowances	<b>Returns and Allowances</b>

8. Click the **Advanced** tab, and then click on the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.
9. Save the process as **LoadSalesPlanODBC** and then run it.
10. Click **OK**, and then close **Turbo Integrator**.

11. Double-click the **Sales\_Plan** cube.

The result appears as shown below:

TOTAL COMPANY		ALL CHANNELS			
Lanterns		Budget Version 1			
Months					
Sales_Plan_Measu	Jan	Feb	Mar	Apr	May
Quantity	140,097.00	100,044.00	289,448.00	230,644.00	143,491.00
Unit Sale Price	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS SALES REV	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Returns and Allow	\$186,924.02	\$133,484.44	\$386,195.79	\$307,734.24	\$191,451.76
Cash Discount	\$138,277.36	\$98,746.33	\$285,689.13	\$227,647.90	\$141,625.32
Volume Discount	\$90,146.22	\$64,374.52	\$186,246.47	\$148,406.72	\$92,328.53
TOTAL DISCOUNT	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Freight	\$429,925.23	\$307,014.22	\$888,250.35	\$707,788.77	\$440,339.03
NET SALES REVEN	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Unit Cost	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
COST OF SALES	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN %	0.00	0.00	0.00	0.00	0.00
Seasons					

The default view was generated in the Workshop of Module 3. If your default view does not look like the above, arrange the dimensions and save the default view now.

12. Close Cube Viewer.

### Results:

You created an ODBC data source name that identifies the spreadsheet. You then created an alias in the Months dimension that refers to each month by number. Next, you created a process that uses a function to convert the Date into the month number, as well as loads the data from the spreadsheet. Finally, you ran the process and confirmed that the data was loaded successfully.

## Use Turbo Integrator to Maintain the Model

- Not just for loading data
- Clear data
- Move data from cube to cube
- Automatic saving
- Chores

**Cognos**  
software



Chores allow for the creation of batch files that may be scheduled to run off hours.

When scheduling a TI process, TI automatically creates a chore that will be scheduled. Either a chore or a process can be run on-demand from the menu. Users may run TI processes (if granted security access). However, only admin group members may edit the processes.

In order to prevent Chores from trying to execute while being edited, it must be deactivated before you can open and edit the Chore. After editing a Chore, it must be reactivated in order to be run on its scheduled time.

**INTERACTION - Text Chat:** Ask participants what other tasks they have seen (or would like to see) TI perform. An alternative is to use the Microphone instead of Text.

## Demo 3: Clear Data from a Cube using a Process

### Purpose:

You want to load data into the Sales\_Plan cube from another data source. Before you do that, you want all the data from the cube to be removed, so all the data that you view later is from the second data source. You will create a process that clears all the data in a cube.

### Task 1. Identify the view to be zeroed.

1. In the Server Explorer window, right-click **Processes** and click **Create New Process**.
2. On the **Data Source** tab, click **TM1**.  
Cube View is enabled automatically.
3. Click **Browse**.  
All the cubes that are available on your server appear.
4. Click the **greatoutdoors: Sales\_Plan** cube, and then click **Create View**.
5. In the **View Extract** window, click **OK**.  
You do not need to make any selections as the default is to select All elements.
6. In the **Enter or Select Named View** box, type **zDeleteAll** and then click **OK**.  
This view should only be used in a TI process. It will try to display all the cells in the cube which may result in a very large view that may be too large in memory. TM1 displays views in alphabetical order so placing a 'z' at the beginning will make it more difficult to select this view.  
Private is disabled because TI processes may only use Public views.
7. In the **Select View** column, click **zDeleteAll**, and then click **OK**.

**8. Click Preview.**

The result appears as shown below:

The screenshot shows the 'Data Source' configuration window for TM1. The 'Datasource Type' section is expanded, showing options for ODBC, Text, ODBO, TM1, SAP, and None. The TM1 option is selected, with 'Cube View' chosen from its dropdown. The 'Data Source Name' field contains 'greatoutdoors:Sales\_Plan->zDeleteAll'. A 'Browse...' button is available to the right of the name field. Below the configuration area is a preview table with the following data:

	Subsidiaries	Channels	Products	Months	Versions	Sales_Plan_Meas
1	TOTAL COMPANY	ALL CHANNELS	TOTAL PRODUCTS	Total Year	Budget Version 1	Quantity

A 'Preview' button is located at the bottom right of the preview area.

**9. Save the process as **SalesPlanDeleteData**.**

## Task 2. Add function to the Prolog tab.

1. Click the **Variables** tab.
2. Click the **Advanced** tab, and then click the **Prolog** tab.

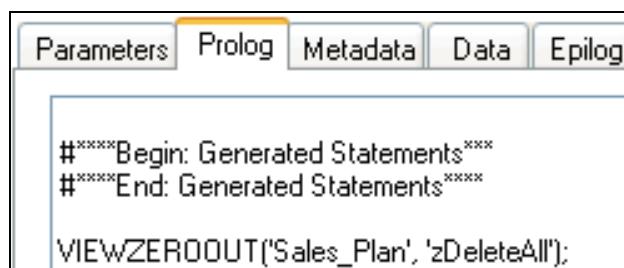
You want to add a function that sets all data values to zero in the Sales\_Plan cube.

### Syntax:

```
VIEWZEROOUT('Cube', 'ViewName');
```

3. Below the two lines already there, type  
**ViewZeroOut('Sales\_Plan','zDeleteAll');**

The results appear as follows:



A screenshot of the Turbo Integrator interface showing the Prolog tab selected. The tab bar at the top includes Parameters, Prolog (which is highlighted in yellow), Metadata, Data, and Epilog. Below the tabs is a text area containing the following code:

```
#****Begin: Generated Statements***  
#****End: Generated Statements****  
  
VIEWZEROOUT('Sales_Plan', 'zDeleteAll');
```

4. Save the process, and run it.
5. When the process completes successfully, click **OK**.
6. Close Turbo Integrator.

## Task 3. View the cube.

- Double-click the **Sales\_Plan** cube.

The result appears as shown below:

TOTAL COMPANY		ALL CHANNELS											
Lanterns		Budget Version 1											
		Months											
Sales_Plan_Measu		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Quantity		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Unit Sale Price	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS SALES REV	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Returns and Allow.	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Cash Discount	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Volume Discount	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
TOTAL DISCOUNT	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Freight	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
NET SALES REVEN	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Unit Cost	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
COST OF SALES	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN %	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Seasons													

- Close the Cube Viewer, saving the view if prompted.

### Results:

You created a process that clears all the data in a cube.

## Demo 4: Accumulate Data

### Purpose:

It has been brought to your attention that since you ignored the data for staff, and you did not accumulate data going into the Sales\_Plan Cube, the cube only contains the last data for each targeted cell.

Your cube is empty again after running the delete process so now you can reload data from the ODBC source and accumulate the data.

You will edit the process to load the data from an ODBC source, accumulate the data, and then compare the results with the results from Demo 2.

### Task 1. Specify the data source.

1. In the Server Explorer window, expand **Processes**, and then double-click **LoadSalesPlanODBC**.
2. Click the **Maps** tab.
3. In the Data Action box, click **Accumulate Values**.

In some cases, such as this one, you may have a number of records in your data source that need to be summarized and put into a cell. This is very common when you have more columns in the data source than you have dimensions in your cube.

In our example, we do not have a staff dimension, so we ignored the names of the staff members. TI will need to ignore the staff column and keep adding data into the appropriate cells. If the data was not accumulated, then each record would overwrite the previous one and the data would be incorrect (too low).

## Task 2. Create variables.

1. Click the **Advanced** tab, and then click on the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.

Because we made a change to the process, we need to be sure the underlying scripts reflect this change.

2. Click **Save**, and then click **Run**.
3. Click **OK**, and then close **Turbo Integrator**.
4. Double-click the **Sales\_Plan** cube.

The result appears as shown below:

TOTAL COMPANY		ALL CHANNELS		Lanterns	
Budget Version 1					
		Months			
Sales_Plan_Measu		Jan	Feb	Mar	Apr
Quantity		429,208.00	252,272.00	752,814.00	1,015,598.00
Unit Sale Price		\$0.00	\$0.00	\$0.00	\$0.00
GROSS SALES REV		\$0.00	\$0.00	\$0.00	\$0.00
Returns and Allow.		\$554,201.70	\$325,925.68	\$959,297.64	\$1,270,925.76
Cash Discount		\$409,972.30	\$241,104.80	\$709,642.15	\$940,171.92
Volume Discount		\$267,268.39	\$157,180.75	\$462,628.60	\$612,913.78
TOTAL DISCOUNT		\$0.00	\$0.00	\$0.00	\$0.00
Freight		\$1,274,663.86	\$749,629.12	\$2,206,384.57	\$2,923,129.32
NET SALES REVEN		\$0.00	\$0.00	\$0.00	\$0.00
Unit Cost		\$0.00	\$0.00	\$0.00	\$0.00
COST OF SALES		\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN		\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN %		0.00	0.00	0.00	0.00
Seasons					

5. Compare your results with the results in the screen capture found in Demo 2, Task 6, Step 11 of this module.

In that view, the Quantity for Jan was only 140097. Now you can see it is 429208. The records are now being accumulated where they were previously being over-written.

As expected, the values are significantly higher than earlier, because data is now aggregated.

6. Close the Cube Viewer.

**Results:**

**You edited a process to load the data from an ODBC file, accumulated the data, and then compared the results with the results from Demo 2.**

## Demo 5: Save Data

### Purpose:

All objects and data in TM1 are in memory, until you save them to the server. You are normally prompted to do this whenever you shut down the server, but you want to ensure that data is saved periodically during the day. You will create a process that saves all your changes, test it, and then schedule it to run every two hours.

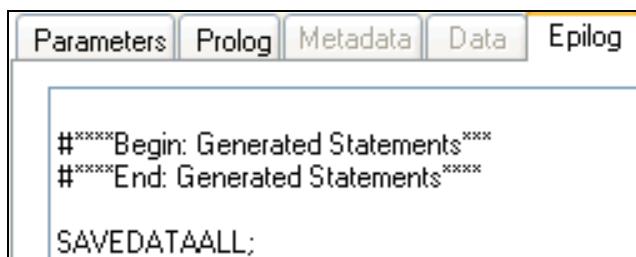
### Task 1. Create a process.

1. In the Server Explorer window, right-click **Processes**, and then click **Create New Process**.

You want to run a TurboIntegrator function that affects all data, and not just a specific cube or view. You do not need to make any changes to the Data Source or Variables tab, and can start working on the Advanced tab.

2. Click the **Advanced** tab.
3. In **Turbo Integrator**, click the **Epilog** tab.
4. After End: Generated Statements, type **SAVEDATAALL;**

The result appears as shown below:



## Task 2. Schedule the process to run every two hours.

- Click the **Schedule** tab, select the **Schedule this Process as a Chore Named** checkbox, and then beside it, type **SaveAllData**.
- Below **Chore Frequency**, beside **Hours**, type **2**.

The result appears as shown below:

Date:	July. 2008					
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	<b>28</b>	29	30	31	1	2
3	4	5	6	7	8	9

Local Time: 10:32:56 AM

GMT time: 2008/07/28 14:32:56

Chore Frequency:

Run Every: 0 Days  
2 Hours  
0 Minutes  
0 Seconds

- Click **Save**, and save the process as **SaveAllData**.

This chore will run every two hours starting immediately.

- Close **TurboIntegrator**.

Below Processes, the new process SaveAllData appears.

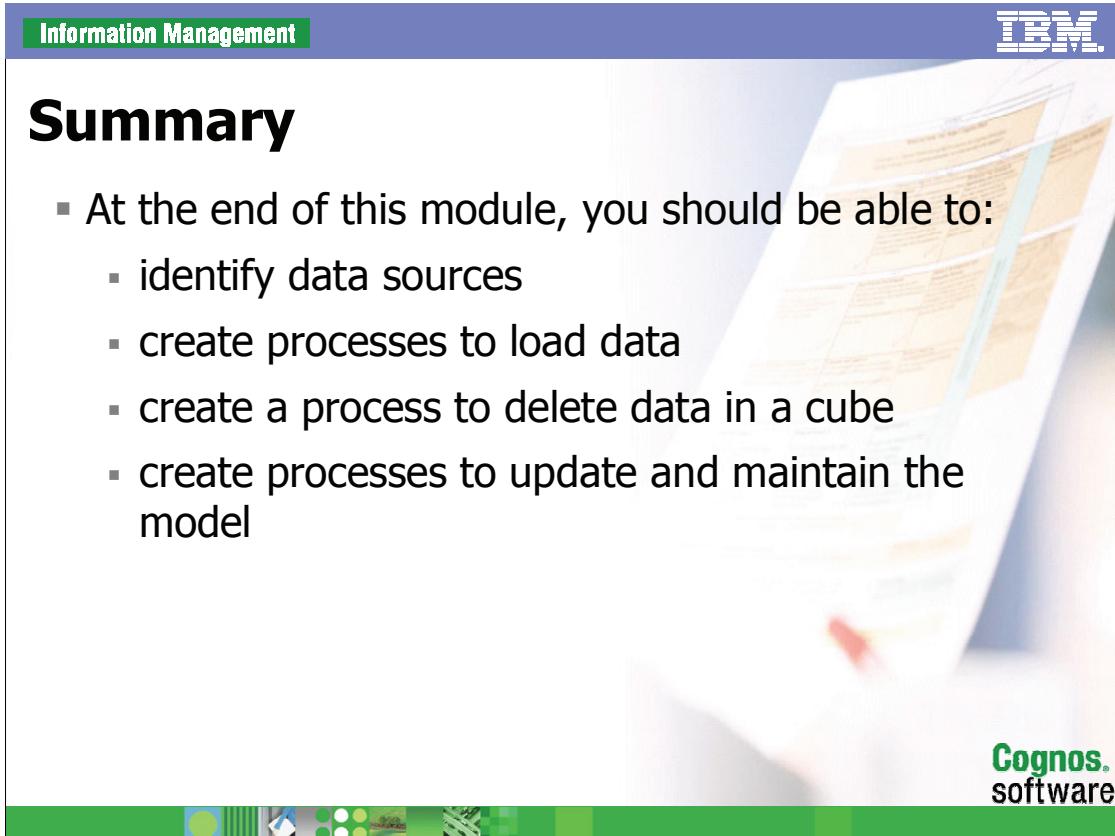
- Expand **Chores**.

Below Chores, the new chore SaveAllData appears.

You can run either the Chore or the Process by right-clicking them and then clicking Run.

### Results:

You created a new process to save all data.



The slide template features a blue header bar with the text "Information Management" on the left and the "IBM" logo on the right. Below the header is a large white area containing a list of objectives. At the bottom is a green decorative bar with various icons.

## Summary

- At the end of this module, you should be able to:
  - identify data sources
  - create processes to load data
  - create a process to delete data in a cube
  - create processes to update and maintain the model

**Cognos.**  
software

## Workshop 1: Create the Franchise\_Revenue Cube

The budget data for franchise revenue is now available. The data is currently in a comma separated values file called Franchise Rev.csv, and needs to go into a cube that holds franchise revenue information for all Subsidiaries and months for both budget versions.

To do this you must:

- Create the Franchise\_Revenue cube containing the following dimensions:
  - Subsidiaries
  - Months
  - Versions
- Create a TI process to import the data into the Franchise\_Revenue cube.
  - Use the SUBST () function to extract specific characters from a column of data in order to match the format of the dimension.
  - This cube will also NOT contain a specified measures dimension. In a later example we will see the difficulty this presents.
- Run the process to load the data
- Browse the cube to verify the data loaded correctly.

For more detailed information outlined as tasks, see the Task Table section.

For the final results, see the Workshop Results section that follows the task table.

### Instructor Notes

It is by design that we are creating this cube without a recommended measures dimension. In a later example we will illustrate why it is best to construct a specific measures dimension.

## Workshop 1: Task Table

Task	Where to Work	Hints
1. Create the Franchise_Revenue cube.	Server Explorer / Create New Cube	<ul style="list-style-type: none"> <li>• Name the cube Franchise_Revenue.</li> <li>• Subsidiaries, Months, Versions</li> <li>• Create a default view</li> </ul>
2. Create a process to load data from a csv file.	Turbo Integrator / Data Source	<ul style="list-style-type: none"> <li>• C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Franchise rev.csv</li> <li>• First row is title row.</li> </ul>
3. Define variables.	Turbo Integrator / Variables	<ul style="list-style-type: none"> <li>• Subsidiaries - Element, vMonth - Other, vBudget1 - Data, vBudget2 - Data, vDate - Element, String</li> <li>• vDate=SUBST(vMonth, 4, 3);</li> <li>• SUBST(string, begin_character, number_of_characters)</li> </ul>
4. Map variables to Franchise_Revenue cube.	Turbo Integrator / Maps	<ul style="list-style-type: none"> <li>• Cube - Update Cube - Franchise_Revenue</li> <li>• Map dimensions</li> <li>• Map Data to Budget Version 1 &amp; Budget Version 2.</li> </ul>

Task	Where to Work	Hints
5. Save and run the process.	Turbo Integrator	<ul style="list-style-type: none"> <li>● Click the Advanced tabs.</li> <li>● Save as LoadFranchiseRevenueCSV, and run process.</li> </ul>
6. Browse the Franchise_Revenue cube.	Cube Viewer	<ul style="list-style-type: none"> <li>● Versions - title, Months - columns, Subsidiaries - rows.</li> </ul>

## Workshop 1: Workshop Results

After Task 1, the cube appears below:

		Months											
Subsidiaries		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
-- TOTAL COMPANY		0	0	0	0	0	0	0	0	0	0	0	0
-- GO AMERICAS		0	0	0	0	0	0	0	0	0	0	0	0
GO Americas		0	0	0	0	0	0	0	0	0	0	0	0
-- GO ASIA PACI		0	0	0	0	0	0	0	0	0	0	0	0
GO Asia Pac		0	0	0	0	0	0	0	0	0	0	0	0
-- GO EUROPE G		0	0	0	0	0	0	0	0	0	0	0	0
GO Accessor		0	0	0	0	0	0	0	0	0	0	0	0
GO Central E		0	0	0	0	0	0	0	0	0	0	0	0
GO Northern		0	0	0	0	0	0	0	0	0	0	0	0
GO Southern		0	0	0	0	0	0	0	0	0	0	0	0

After completing the Workshop, the Data Source tab appears below:

Datasource Type

ODBC

Text

ODBO

Cube

TM1

Cube View

SAP

InfoCube

Data Source Name: C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Franchise rev.csv

Data Source Name On Server: C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Franchise rev.cs

Delimiter Type

Delimited

Fixed Width

Delimiter

Tab  Space  Comma

Semicolon  Other

Quote Char:

Number of title records:

Number Delimiters

Decimal Separator:

Thousands separator:

After completing the Workshop, the variables tab appears below:

Variable Name	Variable Type	Sample Value	Contents
Subsidiaries	String	GO Americas	Element
vMonth	String	07-Jan	Other
vBudget1	Numeric	42265205.04	Data
vBudget2	Numeric	44378465.29	Data
vDate	String	Jan	Element

After completing the Workshop, the Maps tab appears below:

Cube Dimensions Data Consolidations Attributes

Cube Action

- Create Cube
- Recreate Cube
- Update Cube
- No Action

Zero Out Portion

ViewName:

Data Action

- Store Values
- Accumulate Values

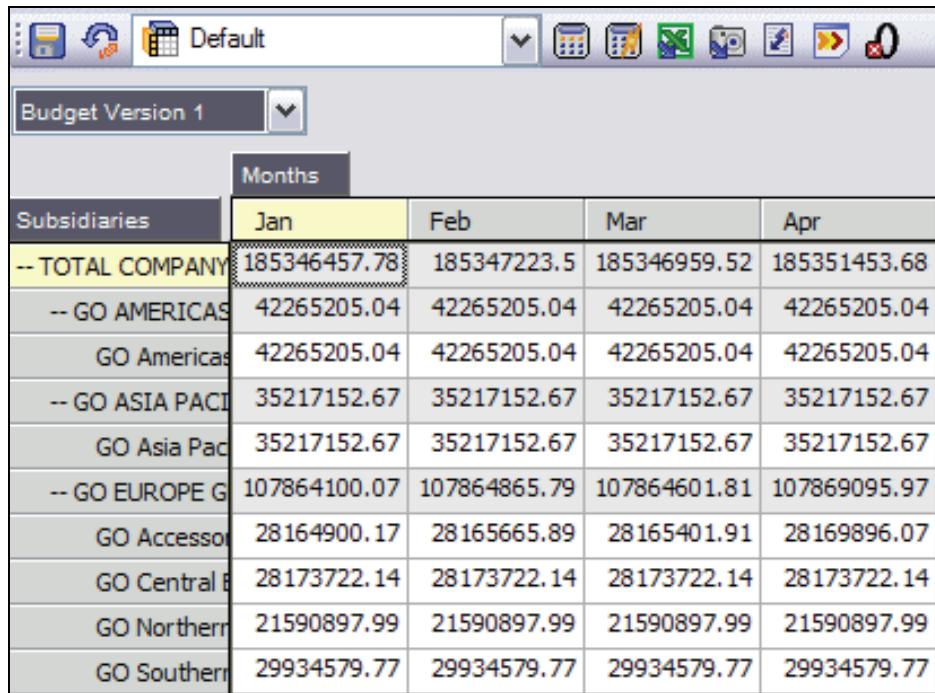
After completing the Workshop, the Dimensions tab appears below:

Element Variable	Sample Value	Dimension
Subsidiaries	GO Americas	Subsidiarie <input type="button" value="▼"/>
vDate	Jan	Months <input type="button" value="▼"/>
(Data Variables)		Versions <input type="button" value="▼"/>

After completing the Workshop, the Data tab appears below:

Data Variable	Element	Element Type
vBudget1	Budget Version 1 ►	Numeric ▾
vBudget2	Budget Version 2 ►	Numeric ▾

After completing the Workshop, the Default view of the Franchise\_Revenue cube appears below:



The screenshot shows the IBM Cognos TM1 Workshop interface. The title bar says "Default". Below it is a toolbar with various icons. The main area is titled "Budget Version 1". A dropdown menu next to the title shows "Months". The table has "Subsidiaries" in the first column and "Jan", "Feb", "Mar", "Apr" in the subsequent columns. The data rows include:
 

Subsidiaries	Jan	Feb	Mar	Apr
-- TOTAL COMPANY	185346457.78	185347223.5	185346959.52	185351453.68
-- GO AMERICAS	42265205.04	42265205.04	42265205.04	42265205.04
GO Americas	42265205.04	42265205.04	42265205.04	42265205.04
-- GO ASIA PACI	35217152.67	35217152.67	35217152.67	35217152.67
GO Asia Pac	35217152.67	35217152.67	35217152.67	35217152.67
-- GO EUROPE G	107864100.07	107864865.79	107864601.81	107869095.97
GO Accessor	28164900.17	28165665.89	28165401.91	28169896.07
GO Central E	28173722.14	28173722.14	28173722.14	28173722.14
GO Northern	21590897.99	21590897.99	21590897.99	21590897.99
GO Southern	29934579.77	29934579.77	29934579.77	29934579.77

## Workshop 1: Step-by-Step Solution

### Task 1. Create the Franchise\_Revenue cube.

1. In Server Explorer, right-click **Cubes** and then click **Create new cube**.
2. Name the cube **Franchise\_Revenue**.
3. Select the **Subsidiaries**, **Months**, and **Versions** dimensions.
4. Click **Create Cube**.
5. In Server Explorer, double-click **Franchise\_Revenue**.
6. Put **Subsidiaries** in the rows, **Months** in the columns and **Versions** in the Title area and then click **Recalculate**.
7. Select the **Months** subset from the Subset Editor and then Recalculate.
8. Drill-down on the **Subsidiaries**.
9. Save the view as **default**.

### Task 2. Create a process to load data from a csv file.

1. In Server Explorer, right-click **Processes** and then click **Create New Process**.
2. On the Data Source tab, click **Text**.
3. Browse to **C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Franchise rev.csv** and then click **OK**.
4. In Number of Title Records, type **1**.
5. Click **Preview**.

## Task 3. Define variables.

1. Click the **Variables** tab.
2. Name and map the variables as follows:

Name	Variable Type	Contents
Subsidiaries	String	<b>Element</b>
vMonth	String	<b>Other</b>
vBudget1	Numeric	<b>Data</b>
vBudget2	Numeric	<b>Data</b>

3. Click **New Variable**, and then rename the **V5** variable to **vDate**.
4. In the formula box type, **vDate=SUBST(vMonth, 4, 3);**  
This will return just the 3 characters in the vMonth variable starting with character number 4 (inclusive).
5. Click **Evaluate**, and then click **OK**.
6. For **vDate**, under **Variable Type** click **String**, and under **Contents**, click **Element**.
7. Click the **Maps** tab.

## Task 4. Map variables to Franchise\_Revenue cube.

1. Click **Update Cube** and then, in the **CubeName** list, select **Franchise\_Revenue**.
2. Click the **Dimensions** tab.
3. Next to **Subsidiaries**, in the **Dimension** column, click **Subsidiaries**.
4. Next to **vDate**, in the **Dimension** column, click **Months**.

5. Next to **(Data Variables)**, in the **Dimension** column, click **Versions**.
6. Click the **Data** tab.
7. Map vBudget1 to **Budget Version 1** under Element.
8. Map vBudget2 to **Budget Version 2** under Element.

## Task 5. Save and run the process.

1. Click the **Advanced** tab and all tabs below.
2. Save the process as **LoadFranchiseRevenueCSV**.
3. Run the process, click **OK** to close the message, and then close Turbo Integrator.

## Task 6. Browse the Franchise\_Revenue cube.

1. Double-click the **Franchise\_Revenue** cube.
2. Recalculate the view.
3. Click **Save** to save as the default view.
4. Close all open windows and then close the greatoutdoors server (saving changes).



**Information Management**



## Add Business Rules

IBM Cognos TM1 9.5



**Cognos.**  
software

© 2009 IBM Corporation



# Objectives

- At the end of this module, you should be able to:
  - discuss a rule
  - construct rules for elements or consolidations
  - use functions in rules
  - populate a pick list using a rule
  - discuss rules in a Rule Worksheet



Cognos.  
software

## Add Business Logic

- Use rules to:
  - ensure consistency of calculations
  - apply custom calculations
  - reformat data
  - override hierarchical calculations
  - share data between cubes

Cognos  
software



Business logic can be created and stored with the cube on a server by using rules, ensuring all users have the same calculations in their analysis.

Rules are stored in text files with the extension, \*.rux. In this module you will use the TM1 Advanced Rule Editor to create and modify rules for our cubes.

---

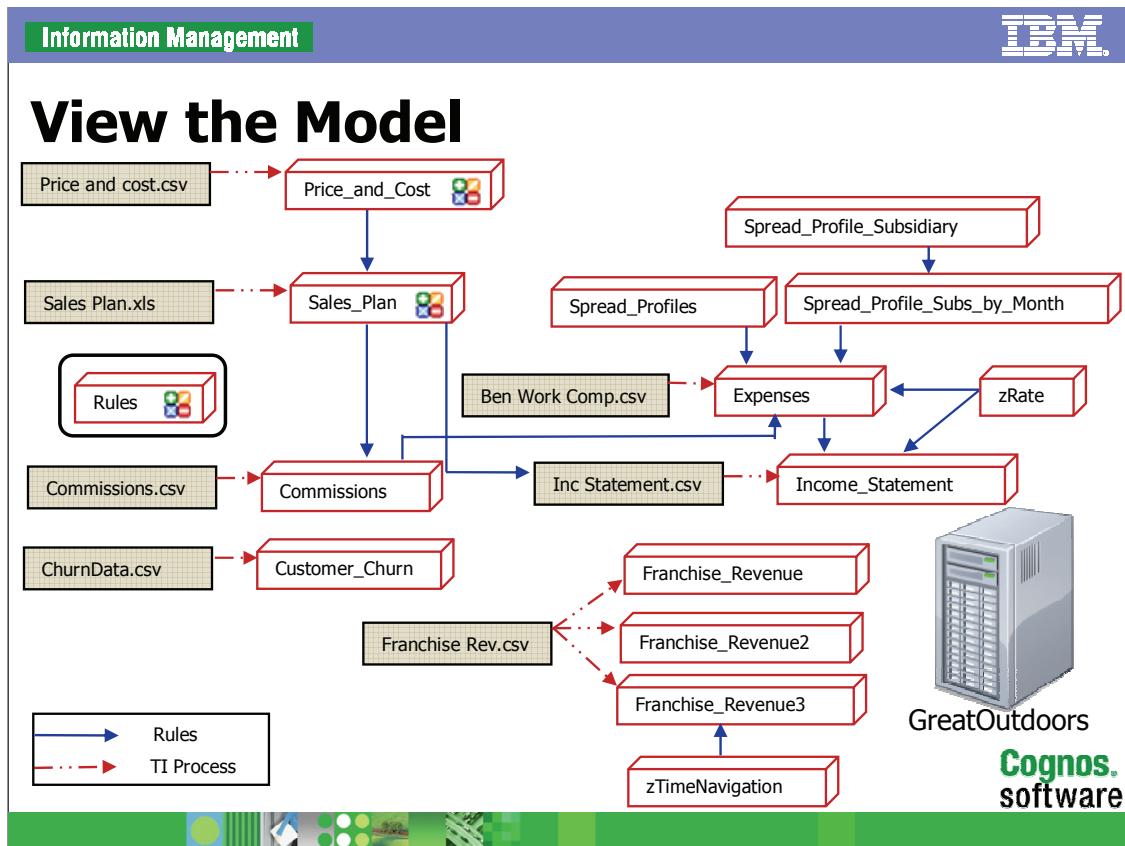
For additional information on rules, please refer to the Reference Material found in the online help or the TM1 Users Guide. The TM1 Rules Guide is also a source of more in depth information.

TM1 has 2 types of Rules Editor. The Advanced Rules Editor or a simpler version (this is the default). In order to change the editor, use the following in the TM1p.ini file under c:\Documents and Settings\Student\Applix\TM1p.ini:

AdvancedRulesEditor=F (for the simple editor or T for the advanced editor)

**NOTE:** If you try to invoke the Advanced Rules Editor in TM1 Perspectives, no editor will appear. This is an MS Office problem with DotNet and the fix cannot be installed properly on our image. Please use TM1 Architect for the demos & workshops or change the setting in c:\Documents and Settings\Student\Applix\TM1p.ini to AdvancedRulesEditor=T.

**INTERACTION - Star Sticker:** Place a sticker next to each bullet point as it is discussed.



In this module you will create a new cube called Rules. It will be used to practice techniques and syntax for business rules attached to cubes.

You will add rules to the following cubes:

- Rules
- Price\_and\_Cost
- Sales\_Plan

---

The demos and workshops in this module were done using the Advanced Rules Editor in TM1 Architect. To date, there is a problem invoking the Advanced Rules Editor in TM1 Perspectives from our images and no editor will appear.

If you do use the original Rules Editor, many of the tools are different and the step-by-step instructions in Demos and Workshops will be different. However, all rules will work in either editor.

You can name the Rules cube something else in order to avoid any confusion, Demo or Practice, for example.

## Create a Rule

- Create an element to store the result
- Select a cube to apply the rule
- Rule structure:
  - [AREA]=FORMULA;
  - AREA is the target cells to which the rule applies
  - FORMULA is the actual calculation

 Cognos  
software

The area definition tells TM1 what values should be calculated by this rule. It restricts the calculation so that it only applies to certain areas of the cube. The calculation can be restricted so that it applies to just one cell in the cube, but more usually we will only restrict the calculation to members in one or two of the dimensions. TM1 will then automatically apply the calculation to all the members of the other dimensions.

Using areas we can specify different calculation for different parts of the cube. Specify elements using the following guides:

- A list: `[{'January', 'February', 'March'}]`
- All elements: `[]`
- A specific cell: `['May', 'London', 'Price']`

---

While area definitions are powerful, they do not allow you to identify a subset of members from a dimension. If you want a rule to apply to four types of fish, you will need to write four separate rules.

### INTERACTION - Text Chat:

Ask for other examples of n level, consolidation, and string elements.

## Use a Modifier in a Rule

- TM1 makes a distinction between base level cells and consolidated level cells.
- Rules can be applied to the following types of cells:
  - consolidated level (C:)
  - base level (N:)
  - string (S:)



Cognos  
software

If the N: modifier is used, the rule will only be applied at the base level. The values of cells at higher levels will be calculated according to the consolidation hierarchies defined in the cube's dimensions. In most cases, if a calculation can be performed just at the base level, it is more efficient to allow TM1 to simply consolidate the results, rather than performing the calculation at every level of the hierarchies.

If the C: modifier is used, the calculation will be applied to consolidated cells only. This can be useful to override the normal consolidation that the dimension hierarchies provide. However, this should be used with care to avoid over complicating the model. Suitable uses of the C: modifier might be to prevent the price being consolidated in our example:

```
[Price]=C:0;
```

The lack of a modifier indicates that the calculation is to be applied at all levels. This is necessary for calculations such as ratios, percentages and averages, which usually cannot be consolidated and must be performed at each level of the dimensions' hierarchies.

There is another modifier, S:, which is used when the rule is being used to 'calculate' string (text) value instead of a numeric value.

## Review Rule Processing

- When data is requested the TM1 engine will look to the left side of a rule for a match.
- If found, it evaluates the formula and returns the result.
- If there are references to members that are calculated by rules, they are calculated before evaluating the result of the current rule.

The first rule, with an area definition that matches the requested data, will be used. Any subsequent rules with area definitions that also match the requested data will not be evaluated. You should take advantage of this, by placing rules with more specific area definitions ahead of rules with more general area definitions. In this way you can use a more specific rule to handle an exceptional case, while the other cases drop through to the more general rule.

For example, if Gross Profits for New York were calculated differently from the rest, we would need to put the New York rule first, and the general rule for Gross profit last.

`['New York','Sales'] = ['Units Sold']*['New York','Price'];`

`['Sales'] = ['Units Sold']*['Price'];`

## Demo 1: Create Simple Rules

### Purpose:

You recognize that the rules are performed in the order they are written, and that a cell has a rule applied to it only once. To see the impact of rules on a cube, you will create a cube called Rules. The cube will contain rules for leaf level elements, consolidations, and string elements.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

Password: <blank>

### Task 1. Create a cube to contain the rules.

1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. In Server Explorer, right-click **Cubes** and then click **Create new cube**.
4. In the **Cube Name** box, type **Rules**.
5. Under Available Dimensions, double-click **Months** and **Price\_and\_Cost\_Measures**.

Typically, you would create a new measures dimension for the new cube. However, this cube is only being used for demonstrations and is not part of the overall model.

6. Click **Create Cube**.
7. Double-click **Rules** to open in the **Cube Viewer**.

---

This cube does not have to be named rules. You may wish to name it Rules\_Demo or just Demo to avoid any confusion.

8. Click **Recalculate**.

The result appears as follows:

	Price_and_Cost_Measures				
Months	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
+ Total Year	\$0.00	\$0.00	\$0.00	0.00	

It is often helpful to open a view of the cube so you may immediately see the result of the rule. You can arrange the windows to display the view and the rule at the same time.

## Task 2. Create rule to set unit sale price to 1, for all values including consolidations.

1. Expand Months to display Jan, Feb, March, Q1 and Total Year.

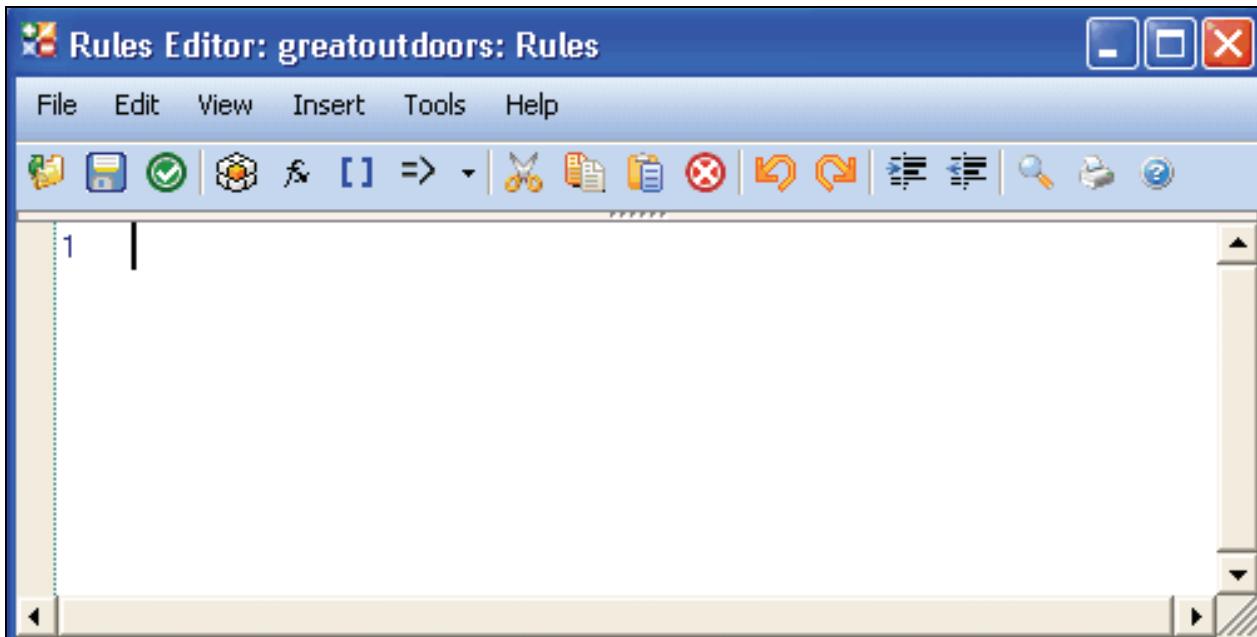
**Cube Viewer: greatoutdoors->Rules->(Unnamed)**

The screenshot shows the Cube Viewer interface with the title bar "Cube Viewer: greatoutdoors->Rules->(Unnamed)". The menu bar includes File, Edit, View, Options, and Help. Below the menu is a toolbar with various icons. The main area displays a data grid with the following columns: Months, Unit Sale Price, Unit Cost, Margin, Margin%, and Comment. The data rows are as follows:

Months	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$0.00	\$0.00	\$0.00	0.00	
-- Q1	\$0.00	\$0.00	\$0.00	0.00	
Jan	\$0.00	\$0.00	\$0.00	0.00	
Feb	\$0.00	\$0.00	\$0.00	0.00	
Mar	\$0.00	\$0.00	\$0.00	0.00	
+ Q2	\$0.00	\$0.00	\$0.00	0.00	
+ Q3	\$0.00	\$0.00	\$0.00	0.00	
+ Q4	\$0.00	\$0.00	\$0.00	0.00	

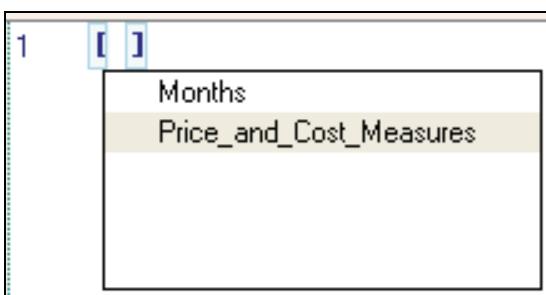
2. In the Server Explorer window, right-click **Rules** and then click **Create Rule**.

The Rules Editor should now be open. The cursor should be blinking on the first row of the editor next to the number **1**. This is where you will create your rules statements.



In the Advanced Editor line numbers are displayed for each row. You may use to indent lines for better organization. You may also Undo and Redo using . Other tools will be reviewed as they are needed.

3. Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.



- Click **Unit Sale Price** and then click **OK**.

The result appears as shown below:

```
1 ['Unit Sale Price' ]
```

This is the area of the cube to populate with our rule.

- At the end of line 1, type **= 1;**

All rules statements must end with a semicolon (;).

```
[1 ['Unit Sale Price' ] =1;]
```

Unit Sale Price is blue indicating that it is a string. Strings must be enclosed in single quotes (''). In the Advanced Editor, numbers appear in red.

- Click **Check Syntax** .

Check Syntax will check your rule for any syntax errors and indicate the row number of an error. It is common to forget to place a semicolon after a rule, this would generate a syntax error.

- Delete the semicolon (;) at the end of line 1.

- Click **Check Syntax** .



This is an example of a Syntax Error message generated in the Rules Editor.

9. Click **OK** to dismiss the error message.
10. Click at the end of line 1 and then type a semicolon.
11. Click **Check Syntax** and then click **Save**.
12. Click the Cube Viewer and click **Recalculate**.

The result appears as shown below:

The screenshot shows two windows side-by-side. The top window is titled "Cube Viewer: greatoutdoors->Rules->(Unnamed)". It displays a grid titled "Price\_and\_Cost\_Measures" with columns: Months, Unit Sale Price, Unit Cost, Margin, Margin%, and Comment. The data rows are: -- Total Year (\$1.00), -- Q1 (\$1.00), Jan (\$1.00), Feb (\$1.00), and Mar (\$1.00). The "Unit Sale Price" column is highlighted. The bottom window is titled "Rules Editor: greatoutdoors: Rules". It shows a toolbar with various icons and a text editor containing the rule: 1 ['Unit Sale Price' ] =1;

All cells in the Unit Sale Price column contain \$1.00. This rule applies to all levels of the Unit Sale Price overriding any consolidations.

## Task 3. Apply the rule to only the leaf level cells.

1. Click the Rules Editor.
  2. Double-click the equal sign and then click Delete  to remove the equal sign.
  3. Click **Insert Qualifier or Operator** and then click =N:.
- 1 ['Unit Sale Price'] = N: 1;**
4. Click **Check Syntax** and then click **Save**.
  5. Click the **Cube Viewer** and then click **Recalculate**.

The result appears as shown below:

Months	Price_and_Cost_Measures				Comment
	Unit Sale Price	Unit Cost	Margin	Margin%	
-- Total Year	\$12.00	\$0.00	\$0.00	0.00	
-- Q1	\$3.00	\$0.00	\$0.00	0.00	
Jan	\$1.00	\$0.00	\$0.00	0.00	
Feb	\$1.00	\$0.00	\$0.00	0.00	
Mar	\$1.00	\$0.00	\$0.00	0.00	
+ Q2	\$3.00	\$0.00	\$0.00	0.00	
+ Q3	\$3.00	\$0.00	\$0.00	0.00	
+ Q4	\$3.00	\$0.00	\$0.00	0.00	

Now you see that \$1.00 appears only in the leaf (**n**) level cells and the consolidations are now performed in the hierarchy by the aggregation engine. By using the N: after the equal sign, you have told TM1 to apply the rule only to the lowest level in the hierarchy and to use the consolidation engine for the consolidations.

Also notice the calculated cells are gray, indicating data may not be entered into these cells. Rules driven cells will not accept data entry.

## Task 4. Apply a rule to both the leaf and consolidation.

1. In the Rules Editor, click after the semicolon and type **C: 20;**

The result appears as shown below:

```
1 ['Unit Sale Price'] = N: 1; C: 20;
```

2. Click **Save**, and then recalculate the view.

The result appears as shown below:

Months	Price_and_Cost_Measures				
	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	
-- Q1	\$20.00	\$0.00	\$0.00	0.00	
Jan	\$1.00	\$0.00	\$0.00	0.00	
Feb	\$1.00	\$0.00	\$0.00	0.00	
Mar	\$1.00	\$0.00	\$0.00	0.00	
+ Q2	\$20.00	\$0.00	\$0.00	0.00	
+ Q3	\$20.00	\$0.00	\$0.00	0.00	
+ Q4	\$20.00	\$0.00	\$0.00	0.00	

All consolidated values are \$20.00 and the leaf level cells are \$1.00. The rule overrides the consolidation engine.

3. In the Rules Editor, on the second line, type the following rule:

```
[Q1, 'Unit Sale Price'] = 7;
```

The result appears as follows:

```
1 ['Unit Sale Price'] = N: 1; C: 20;
2 [Q1, 'Unit Sale Price'] = 7;
```

This rule is more specific because it identifies only the Q1 cells for Unit Sale Price.

4. Click **Check Syntax** and the click **Save** in the Rules Editor.

- Click **Recalculate** in the Cube Viewer.

The result appears as follows:

Months	Price_and_Cost_Measures				
	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	
-- Q1	\$20.00	\$0.00	\$0.00	0.00	
Jan	\$1.00	\$0.00	\$0.00	0.00	
Feb	\$1.00	\$0.00	\$0.00	0.00	
Mar	\$1.00	\$0.00	\$0.00	0.00	
+ Q2	\$20.00	\$0.00	\$0.00	0.00	
+ Q3	\$20.00	\$0.00	\$0.00	0.00	
+ Q4	\$20.00	\$0.00	\$0.00	0.00	

The result did not change because of the order in which rules are calculated. Once a rule applies to a cell, TM1 does not look for any other calculation for that cell, instead it calculates other cells. This allows cells to be calculated quicker. It also means you should put more specific rules before generic rules in the editor.

- In the **Rules Editor**, insert an extra row at the top.
- Select the **'Q1'. 'Unit Sale Price'** line, click **Edit** and then click **Cut**.
- Click on the first row, click **Edit** and then click **Paste**.

The result appears as shown below:

```
1  ['Q1','Unit Sale Price'] =7;
2  ['Unit Sale Price'] = N: 1; C: 20;
```

- Save the rule, and then recalculate the view.

The result appears as shown below:

Months	Price_and_Cost_Measures				
	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	
-- Q1	\$7.00	\$0.00	\$0.00	0.00	
Jan	\$1.00	\$0.00	\$0.00	0.00	
Feb	\$1.00	\$0.00	\$0.00	0.00	
Mar	\$1.00	\$0.00	\$0.00	0.00	
+ Q2	\$20.00	\$0.00	\$0.00	0.00	
+ Q3	\$20.00	\$0.00	\$0.00	0.00	
+ Q4	\$20.00	\$0.00	\$0.00	0.00	

By changing the order of the rules, the more specific rule is calculated and then the more generic rule is applied to all the 'other' cells. Since Q1 for Unit Sale Price was calculated first, it is not recalculated as a part of the second rule. This allows TM1 to be more efficient in the calculations and saves resources.

- Save the view as **Default**.

## Task 5. Populate a string element with a rule.

In Module 4 you populated the Comment element using a Pick List attribute on the Products dimension. The Rules cube does not contain the Products dimension so that pick list is not available in this cube. Instead you will create a rule to populate the Comment.

- Expand the **Rules** cube and double-click the **Rules** icon  to open the Rules Editor (if not already open).
- Press **Enter** after the last rule to create a new line.
- On line 3, click **Brackets**, and then double-click **Price\_and\_Cost\_Measures**.
- Click **Comment**, and then click **OK**.

5. Click outside the brackets on line 3, click **Insert Qualifier or Operator** and then click **= S:**.
6. Type '**Winter**'; (including quotes).

The result appears as follows:

```
[Q1,'Unit Sale Price'] =7;
[Unit Sale Price] = M: 1; C: 20;
[Comment] = S: 'Winter';
```

7. Click **Check Syntax** and if there are no errors, click **Save** and then recalculate the view.

The result appears as follows:

	Price_and_Cost_Measures				
Months	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	Winter
-- Q1	\$7.00	\$0.00	\$0.00	0.00	Winter
Jan	\$1.00	\$0.00	\$0.00	0.00	Winter
Feb	\$1.00	\$0.00	\$0.00	0.00	Winter
Mar	\$1.00	\$0.00	\$0.00	0.00	Winter
+ Q2	\$20.00	\$0.00	\$0.00	0.00	Winter
+ Q3	\$20.00	\$0.00	\$0.00	0.00	Winter
+ Q4	\$20.00	\$0.00	\$0.00	0.00	Winter

All cells now show a Comment of Winter. When the result of a rule is a string, it must be a string element. You must also specify a string rule using S:.

Remember, a string element must reside in the last dimension in a cube in order to be stored properly. If you make your measures dimension the last dimension, you will always know where to put the string elements.

---

If you are comparing two string elements ('A'='B'), you must use @= instead of =.

## Task 6. Add logic to a rule.

1. In the Rules Editor, click the end of the string rule and delete 'Winter';

You will add logic to this rule that will evaluate the Unit Sale Price and display 'Summer' when the price is over 5 and 'Winter' under 5.

2. After the S: type, IF ( ['Unit Sale Price'] > 5, 'Summer', 'Winter');

Syntax for IF: IF (test, true\_condition, false\_condition)

The result appears as follows:

```
[Q1,'Unit Sale Price'] =7;
[Unit Sale Price] = N: 1; C: 20;
[Comment] = S: IF (['Unit Sale Price']>5, 'Summer', 'Winter');
```

3. Save the rule and recalculate the view.

The result appears as follows:

Months	Price_and_Cost_Measures				
	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	Summer
-- Q1	\$7.00	\$0.00	\$0.00	0.00	Summer
Jan	\$1.00	\$0.00	\$0.00	0.00	Winter
Feb	\$1.00	\$0.00	\$0.00	0.00	Winter
Mar	\$1.00	\$0.00	\$0.00	0.00	Winter
+ Q2	\$20.00	\$0.00	\$0.00	0.00	Summer
+ Q3	\$20.00	\$0.00	\$0.00	0.00	Summer
+ Q4	\$20.00	\$0.00	\$0.00	0.00	Summer

You will now exclude the Q1 cell from this rule. To exclude an area from a rule, use the STET function immediately before the rule.

Syntax for STET is: **STET**; (it has no parameters or brackets).

4. Click the beginning of line 3 and press **Enter** to insert a blank line.
5. On the blank line type, **[Q1] = STET;**

The result appears as follows:

```
1  ['Q1', 'Unit Sale Price' ] =7;
2  ['Unit Sale Price' ] = N: 1; C: 20;
3  ['Q1' ] = STET;
4  ['Comment' ] = S: IF ([Unit Sale Price] >5, 'Summer', 'Winter');
```

6. Save the rule and recalculate the view.

The result appears as follows:

Months	Price_and_Cost_Measures				
	Unit Sale Price	Unit Cost	Margin	Margin%	Comment
-- Total Year	\$20.00	\$0.00	\$0.00	0.00	Summer
-- Q1	\$7.00	\$0.00	\$0.00	0.00	
Jan	\$1.00	\$0.00	\$0.00	0.00	Winter
Feb	\$1.00	\$0.00	\$0.00	0.00	Winter
Mar	\$1.00	\$0.00	\$0.00	0.00	Winter
+ Q2	\$20.00	\$0.00	\$0.00	0.00	Summer
+ Q3	\$20.00	\$0.00	\$0.00	0.00	Summer
+ Q4	\$20.00	\$0.00	\$0.00	0.00	Summer

No rule was applied to the Q1 cell. String cells cannot be aggregated so no value is displayed.

7. Close the Rules Editor, and then close the Cube Viewer, clicking **No** if asked to save changes to the view.

### Result:

To see the impact of rules on a cube, you created a cube called Rules. The cube contained rules for leaf level elements, consolidations, and string elements. Next you will use cubes that make up the GreatOutdoors model to display the results of our business rules.

## Use a Rule to Override Aggregation

- Rules override stored data or default aggregation.
- Aggregation is performed at the consolidated level
  - It is the summation of the children at the lowest level or N: level in the hierarchy.
- Use C: after the = in a rule to override the default aggregation.

Cognos  
software



When a dimension is created, TM1 automatically aggregates the lowest levels according to the dimension hierarchy. Sometimes, it is necessary to override the default aggregation with custom calculations or no calculations.

One example is a Price of a Product should not be summed along the dimension but instead display a zero or an average calculation.

---

All rules can be found in c:\edcognos\P6502\TextFiles. Rules are stored in \*.rux files. Open the file in a text editor (Notepad or Wordpad) and copy the rule or feeder. There is one RUX file per cube per module. It contains all the rules up through the module for which it is named. TXT files contain TI scripts or SQL Queries.

## Demo 2: Create a Rule to Suppress a Consolidation

### Purpose:

You do not want unit price values to be summed or calculated at the consolidation level.

You will add elements to contain the values for calculating average price at the leaf level. Next, at the consolidation level, you will create the average calculation using your leaf level calculations.

### Task 1. Create elements to hold calculations.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Open **Price\_and\_Cost\_Measures** in the Dimension Editor.

The calculation for average price at the consolidated level = (total price for elements)/(total number of prices). You have the total price for elements in the consolidated elements for Unit Sale Price so we need to retain that information in a holder element. You will override the consolidations for Unit Sale price with the result of the rule.

You do not currently have an element that tells you how many elements have prices so you will create a count element and assign the value of 1 where a price is found.

You will create two new elements to hold data for our calculation.

2. Click **Margin %**, click **Edit** and then click **Insert Element**.

3. Add two new simple elements after **Margin%**:
  - **UnitPrice\_hold**
  - **UnitPrice\_count**
4. Close the editor, saving the changes when prompted.

The results appear as follows:



## Task 2. Create new elements.

1. Open the **Default** view for the **Price\_and\_Cost** cube.
2. Swap **Months** and **Price\_and\_Cost\_Measures** (displaying Unit Sale Price).
3. Click **Price\_and\_Cost\_Measures** (in the column area) to open the Subset Editor.
4. In the **Subset Editor**, click **All**, Ctrl+click **Unit Sale Price**, **UnitPrice\_count** and **UnitPrice\_hold**, click **Keep**, and then click **OK**.
5. Click **Recalculate**.
6. Click **CAMPING EQUIPMENT** to expand.

7. Ensure that **Budget Version 1** is selected, and then click **Recalculate**.

The result appears as shown below:

Price_and_Cost_Measures			
Products	Unit Sale Price	UnitPrice_hold	UnitPrice_count
-- TOTAL PRODUCT	\$3,497.35	0	0
-- CAMPING EQL	\$802.04	0	0
Cooking Gear	\$40.31	0	0
Lanterns	\$29.12	0	0
Packs	\$175.37	0	0
Sleeping Bag	\$103.83	0	0
Tents	\$453.41	0	0
+ GOLF EQUIPM	\$1,724.56	0	0
+ MOUNTAINEER	\$454.08	0	0

8. Save the view as **AverageRule**.
  9. In the **Server Explorer** window, right-click the **Price\_and\_Cost** cube and then click **Create Rule**.
  10. With the cursor on line 1 type, # hold the Unit Sale Price for the average price.
- The # indicates the rest of the line is a comment and will be ignored even if it contains a calculation. This is useful for documenting the following lines or to temporarily disable a rule.
11. Press **Enter** to add another line and click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
  12. Click **UnitPrice\_hold**, and then click **OK** to return to the Rules Editor.
  13. Click to the right of the ], click the **Insert Qualifier or Operator** icon and then click **=N:**.
  14. Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
  15. Click **Unit Sale Price** and then click **OK**.

16. Click to the right of the last bracket and then type a semicolon.

```
1 #hold the Unit Sale Price for the Average Price
2 ['UnitPrice_hold'] = N: ['Unit Sale Price'] ;
```

17. Click **Check Syntax** and correct any syntax errors.

18. Click **Save** and recalculate the view.

The result appears as follows:

Golf Shop				Budget Version 1	Jan
Price_and_Cost_Measures					
Products	Unit Sale Price	UnitPrice_hold	UnitPrice_count		
-- TOTAL PRODUCT	\$3,497.35	3497.35	0		
-- CAMPING EQL	\$802.04	802.04	0		
Cooking Gear	\$40.31	40.31	0		
Lanterns	\$29.12	29.12	0		
Packs	\$175.37	175.37	0		
Sleeping Bag	\$103.83	103.83	0		
Tents	\$453.41	453.41	0		

The UnitPrice\_hold column has the same values as the Unit Sale Price column. Next you want to create a counter that indicates the number of items that are being counted in the Unit Sale Price.

### Task 3. Create a counter.

1. On line 3 of the Rules Editor, click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
2. Click **UnitPrice\_count** and then click **OK**.
3. After the brackets type **=N:**

- Click the **Insert Function** icon .

You will create a function that evaluates the Unit Sale Price. The function determines there is a value, and enters the number 1 in the corresponding cell; if there is no value, it enters 0 in the cell. You will use this function in another calculation to summarize how many cells contain prices, and also use it in your average price calculation. To do this you will use an IF statement.

SYNTAX for IF: IF(condition, true\_condition, false\_condition)

- In the **Select a Category** list click **Logical**.
- In the **Select a Function** list click **IF** and then click **OK**.
- In the **expression** box type **['Unit Sale Price'] <> 0**.
- In the **true-value** box type, **1**.
- In the **false-value** box type, **0**.

This expression will look at the Unit Sale Price and if there is a value other than 0, it will return a 1 to the counter, if there is a 0, it will place a 0 in the counter.

The results appear as follows:

Function Arguments	
<b>IF</b>	
<b>expression</b>	<b>['Unit Sale Price'] &lt;&gt; 0</b>
<b>true-value</b>	<b>1</b>
<b>false-value</b>	<b>0</b>

- Click **OK**.
- Type a semicolon, click **Check Syntax**, and then click **Save**.

## 12. Recalculate the view.

The result appears as follows:

The screenshot shows three windows from the TM1 Architect interface:

- TM1 Architect: Server Explorer**: Shows the project structure under TM1, including the greatoutdoors cube, its cubes (Franchise\_Revenue, Price\_and\_Cost), dimensions, and rules.
- Cube Viewer: greatoutdoors->Price\_and\_Cost->Ave...**: Displays a grid titled "Price\_and\_Cost\_Measures" with columns: Products, Unit Sale Price, UnitPrice\_hold, and UnitPrice\_count. The data includes items like TOTAL PRODUCT, CAMPING EQUIPMENT, Cooking Gear, Lanterns, Packs, Sleeping Bag, Tents, and GOLF EQUIPMENT.
- Rules Editor: greatoutdoors: Price\_and\_Cost**: Shows the rule code for calculating average price:
 

```

1 #hold the Unit Sale Price for the Average Price
2 ['UnitPrice_hold'] = M: ['Unit Sale Price'] ;
3 ['UnitPrice_count'] = M: IF(['Unit Sale Price'] <> 0, 1, 0);
4
      
```

This confirms that for the leaf-level elements, the Unit Sale Price is really the price for one unit. For consolidated items, you want to look at the average unit price.

## Task 4. Create the average unit price.

1. On line 4 of the Rules Editor, click the **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
2. Click **Unit Sale Price** and then click **OK**.
3. Click outside the brackets and type = C:.
4. Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
5. Click **UnitPrice\_hold** and then click **OK**.
6. Click to the right of the last bracket and then type \.

In TM1, \ indicates division, where a number divided by zero is replaced by 0. If you use / (the normal division sign), when you divide by zero you get 'NA'.

7. Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
8. Click **UnitPrice\_count**, click **OK**, and then add a ; to the end of the line.

The result appears as follows:

```

1  #hold the Unit Sale Price for the Average Price
2  ['UnitPrice_hold'] = M: ['Unit Sale Price'] ;
3  ['UnitPrice_count'] = M: IF(['Unit Sale Price'] <> 0, 1, 0);
4  ['Unit Sale Price'] = C: ['UnitPrice_hold'] \ ['UnitPrice_count'] ;

```

- Save the rule and recalculate the view.

The result appears as follows:

Price_and_Cost_Measures			
Products	Unit Sale Price	UnitPrice_hold	UnitPrice_count
-- TOTAL PRODUCT	\$166.54	3497.35	21
-- CAMPING EQUIPMENT	\$160.41	802.04	5
Cooking Gear	\$40.31	40.31	1
Lanterns	\$29.12	29.12	1
Packs	\$175.37	175.37	1
Sleeping Bag	\$103.83	103.83	1
Tents	\$453.41	453.41	1
+ GOLF EQUIPMENT	\$431.14	1724.56	4
+ MOUNTAINEERING	\$113.52	454.08	4
+ OUTDOOR PRODUCTS	\$7.43	22.29	3
+ PERSONAL ACCESSORIES	\$98.88	494.38	5

In Task 3, Step 12 it shows the Unit Sale Price for CAMPING EQUIPMENT as \$802.04. Now you have added a rule to use the average price of the products for the consolidated elements which reflects the average Unit Sale Price for CAMPING EQUIPMENT is \$160.41.

There is no need to format UnitPrice\_hold or UnitPrice\_count because these will probably not be used in reports or analysis, only in the calculation.

- Close the Rules Editor and the Price\_and\_Cost cube, click **Yes** to save changes to the 'Average Rule' view if prompted.

### Results:

You added elements to contain the values for calculating average price at the leaf level. Next, at the consolidation level, you created the average calculation using your leaf level calculations.

## Use a Function in a Rule

- Use functions to:
  - reformat data
  - combine data
  - logical operations
  - share data
- Identify an element using long form

A cell in the current cube may be referenced by the element name in brackets. You may also reference cells in the current cube or across cubes using the DB function:

- Syntax: DB('cube\_name', 'dimension1', 'dimension2', ... 'dimensionn');
- An element\_name or !dimension must be referenced for each 'dimension' in the source cube in the order listed in the Server Explorer tree.
- DB('Price\_and\_Cost', !Channels, 'TOTAL PRODUCTS', !Months, !Versions, 'Unit Sale Price')
- Enclose the cube name or any element name in single quotes.
- Use an exclamation mark (!) preceding a dimension name to reference the current element.

It is helpful to open the On Line help and show the Rules Functions and Turbo Integrator functions in the Reference Guide.

In the example displayed, !Channels means the function will look to the target cell for a match to the Channels dimension in the source cube and TOTAL PRODUCTS is an element from the Products dimension and 'Unit Sale Price' is in the Measures dimension in the source.

## Demo 3: Create a Rule to Share Data Between Cubes

### Purpose:

In the Sales\_Plan cube, the Quantity has been loaded but you need the Unit Sale Price in order to calculate Gross Revenue. The Unit Sale Price exists in the Price\_and\_Cost cube so you will use a rule to pull that value into this cube. This way, if the price in the Price\_and\_Cost cube is updated, the change will automatically be reflected in the Sales\_Plan cube.

### Task 1. Create a rule to bring the unit price to the Sales\_Plan cube.

1. In the Server Explorer window, open the Default view of the Sales\_Plan cube.
2. Pick a leaf element from each dimension:
  - Subsidiaries: **GO Americas**
  - Channels: **Department Store**
  - Products: **Cooking Gear**
  - Versions: **Budget Version 1**

- Click **Recalculate** and save the view as **Revenue**.

The result appears as follows:



The screenshot shows a Data Grid window with the following settings at the top:

- Filter 1: GO Americas
- Filter 2: Department Store
- Filter 3: Cooking Gear
- Filter 4: Budget Version 1

The grid displays monthly sales data for four months (Jan, Feb, Mar, Apr). The columns are labeled "Sales\_Plan\_Measur" (which is partially cut off), "Jan", "Feb", "Mar", and "Apr". The data rows include:

Sales_Plan_Measur	Jan	Feb	Mar	Apr
Quantity	47,742.00	11,935.00	43,090.00	17,024.00
Unit Sale Price	\$0.00	\$0.00	\$0.00	\$0.00
GROSS SALES REV	\$0.00	\$0.00	\$0.00	\$0.00
Returns and Allow.	\$88,190.61	\$22,047.61	\$79,598.51	\$31,450.34
Cash Discount	\$65,239.03	\$16,310.20	\$58,883.46	\$23,265.18
Volume Discount	\$42,530.70	\$10,632.99	\$38,387.10	\$15,167.13
TOTAL DISCOUNT	\$0.00	\$0.00	\$0.00	\$0.00
Freight	\$202,838.37	\$50,709.49	\$183,076.58	\$72,335.76
NET SALES REVENUE	\$0.00	\$0.00	\$0.00	\$0.00
Unit Cost	\$0.00	\$0.00	\$0.00	\$0.00
COST OF SALES	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN	\$0.00	\$0.00	\$0.00	\$0.00
GROSS MARGIN %	0.00	0.00	0.00	0.00
Seasons				

- In the Server Explorer window, right-click the **Sales\_Plan** cube and then click **Create Rule**.
- Click **Brackets** and then double-click **Sales\_Plan\_Measures**.
- Click **Unit Sale Price** and then click **OK**.
- Click to the right of the last bracket, click the **Insert Qualifier or Operator** icon, and then click **=N:**

8. Click the **Insert Function** icon, scroll down in the Select a Function list, and click **DB** and then click **OK**.

The DB() uses the long form name for any element. It can be used to refer to elements in the current cube or across cubes. In this example, you will use it to pull the Unit Sale Price from the Price\_and\_Cost cube into the Sales\_Plan cube. This ensures that when a price is added or updated in the Price\_and\_Cost cube, the change will be reflected in the Sales\_Plan.

9. In the **Cube** list click **Price\_and\_Cost**.

Cube:		
<input checked="" type="checkbox"/> Price_and_Cost		
<input type="checkbox"/> Include Control Cubes		
Dimensions: 5		<b>Reset All</b>
	Name	Reference
► 0	Channels	!Channels
1	Products	!Products
2	Months	!Months
3	Versions	!Versions
4	Price_and_Cost_Measures	!Price_and_Cost_Measures

All the dimensions in the Price\_and\_Cost cube are listed. The reference to be used in the DB () function is displayed in the Reference column. You can select a specific element from a dimension by clicking the Subset icon.

The ! in front of the Reference dimensions indicates that TM1 will use the current value for that dimension will be used in the function instead of using a specific element in the function. This allows the function to be more dynamic.

10. In the **Price\_and\_Cost\_Measures** row, click the **Subset** icon .
11. Click **Unit Sale Price**, and then click **OK**.

Dimensions: 5

Reset All

	Name	Reference	
0	Channels	!Channels	
1	Products	!Products	
2	Months	!Months	
3	Versions	!Versions	
▶ 4	Price_and_Cost_Measures	Unit Sale Price	

You used the element name for Unit Sale Price because you only want that value brought into the Sales\_Plan cube and no other elements.

12. Click **OK**.
13. Add a ; to the end of the line.
14. Press **Enter** and then type the following:

**C:['GROSS SALES REVENUE']\['Quantity'];**

The result appears as follows:

```
1  ['Unit Sale Price'] = N: DB('Price_and_Cost', !Channels,
2                               !Products, !Months, !Versions, 'Unit Sale Price');
3  C: ['GROSS SALES REVENUE'] \ ['Quantity'] ;
```

The first rule pulls the Unit Sale Price from Price\_and\_Cost at the N: level, the consolidated levels instead will compute the Unit Sale price based on the Gross Sales Revenue divided by Quantity so it will not be aggregated at the consolidated levels.

Spaces and returns may be inserted into rules to make them more readable.

15. Click **Save** in the Rules Editor, and then recalculate the view.

The result appears as follows:

Sales_Plan_Measu	Jan	Feb	Mar	Apr	May
Quantity	47,742.00	11,935.00	43,090.00	17,024.00	4,739.00
Unit Sale Price	\$42.33	\$42.33	\$42.33	\$42.33	\$42.33

16. Save the view as **Price**.

17. Close the Rules Editor and then close the Cube Viewer.

Leave TM1 Architect open for the next demo.

### Results:

You shared the Unit Sale Price in the Price\_and\_Cost cube with the Sales\_Plan cube using a DB() function in a rule.

## Populate a Pick List Using a Rule

- Pick lists can be populated from a rule on a TM1 Control Cube.
- A control cube is used internally by TM1 to store information about the model.
  - Security, Attributes, PickList, etc.
- You can create pick lists using rules on control cubes.
  - reduces manual maintenance of objects



Cognos  
software

Control cubes and dimensions are hidden by default. They are preceded with a right-curly bracket (}).

This gives you greater control over which cube cells should contain pick lists and allows greater flexibility in defining pick lists for individual cells.

You can also create rules for the pick list control cube, which allows you to define pick lists for any section of a cube, from a single cell to the entire cube.

A pick list control cube is composed of the same dimensions as the regular cube it is associated with, along with an additional dimension named }Picklist.

The }Picklist dimension contains a single string element, named Value.

## Creating a Pick List Control Cube

- Right-click the cube to apply the pick list
  - Click Create Pick List Cube
  - This creates }PickList\_cubename
    - where cubename is the name of the original cube
  - Type the list in the cells of the control cube
  - Write a rule against the new control cube

Cognos  
software



When typing into the cell of the control cube, use the same syntax as the attribute:

- Static: value1: value2...valueN
- Subset:Dimension\_name:Subset\_name
- Dimension:Dimension\_name

Rules that define pick lists follow the same conventions as all other TM1 rules.

- Pick lists are strings so the string qualifier must be used (=S:)

---

It is possible to have a numeric pick list, but not probable. You would only do this if you want the pick list to aggregate.

## Demo 4: Create a Pick List Using a Rule

### Purpose:

You need to add more pick lists to the Comment element in the Price\_and\_Cost cube. You would like to use rules to populate the lists so you will not have to type them all manually in the attributes.

TM1 Server: greatoutdoors

User/Password: admin/<blank>

TM1 Architect: Server Explorer

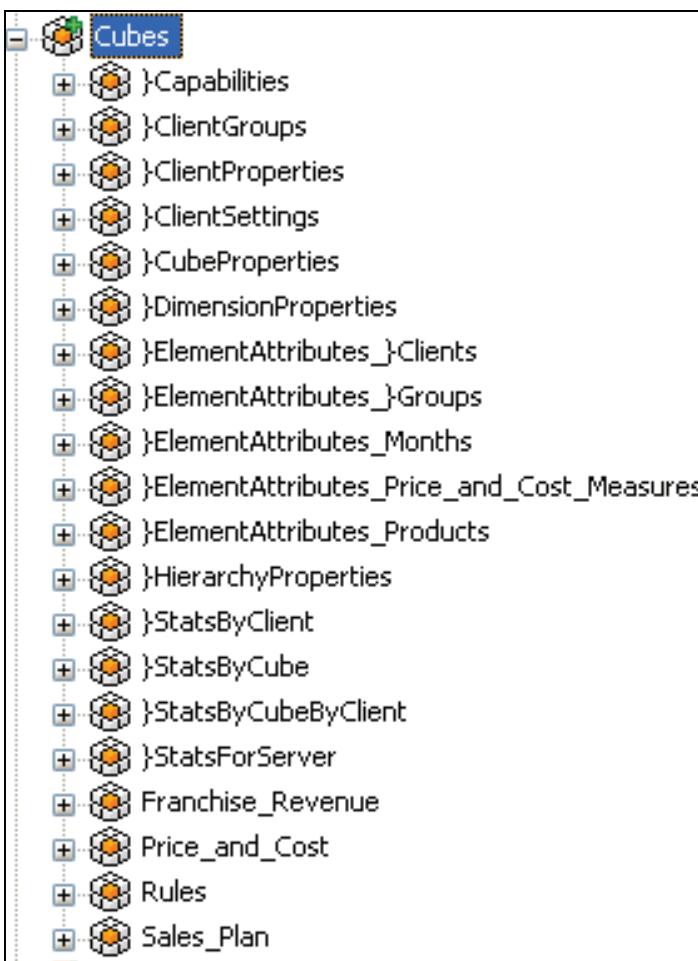
### Task 1. Display control cubes.

1. Open the **Price Comment** view of the Price\_and\_Cost cube and then click to expand **OUTDOOR PROTECTION**.
2. If necessary, select **Comment** from Price\_and\_Cost\_Measures, and then collapse **Camping Equipment**.

Golf Shop		Budget Version 1		Comment	
		Months			
Products		January	February	March	April
-- TOTAL PRODUCT					
+ CAMPING EQL					
+ GOLF EQUIPM					
+ MOUNTAINEER					
-- OUTDOOR PR					
First Aid					
Insect Repe					
Sunscreen					

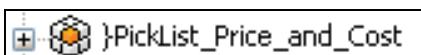
Move this view to the side so that you can see the Server Explorer tree.

3. In the Server Explorer window, click **View** and then click **Display Control Objects**.



Control objects (cubes and dimensions) are identified with the curly-bracket ({}).

4. Right-click the **Price\_and\_Cost** cube and then click **Create Pick List Cube**.



A control cube is added and named **}PickList\_cubename**.

This cube contains the same dimensions as the **Price\_and\_Cost** cube with one more called **}PickList**. You can apply rules to this cube just as you can apply rules to a data cube in TM1. This allows for greater flexibility and easier maintenance of your pick lists.

## Task 2. Create and use a static pick list.

1. Double-click the }PickList\_Price\_and\_Cost control cube, and then click Recalculate.

The screenshot shows a control cube interface with the following dimensions:

- ALL CHANNELS
- TOTAL PRODUCTS
- Total Year
- Budget Version 1

Below these dimensions is a table titled }PickList:

Price_and_Cost_Me	Value
Unit Sale Price	
Unit Cost	
Margin	
Margin%	
Comment	
UnitPrice_hold	
UnitPrice_count	

The cube contains all the dimensions of the Price\_and\_Cost cube and another dimension named }PickList.

2. Arrange the }Picklist cube to look like following:
  - Title elements: **Golf Shop, Budget Version 1, Jan, Comment**
  - Columns: **}Picklist**
  - Rows: **Products** and drill down on **OUTDOOR PROTECTION**.

3. Click **Recalculate**.

Products	Value
-- TOTAL PRODUCT	
+ CAMPING EQL	
+ GOLF EQUIPM	
+ MOUNTAINEER	
-- OUTDOOR PR	
First Aid	
Insect Repellent	
Sunscreen	
+ PERSONAL AC	

4. Click the **Value** for **Insect Repellents** and type the following:  
**static: :Jun:Jul:Aug:Sep**

5. Click **Recalculate**.

Sports Store	Comment	Jan
Budget Version 1		
)PickList		
Products	Value	
-- TOTAL PRODUCT		
-- CAMPING EQL		
Cooking Gear		
Lanterns		
Packs		
Sleeping Bag		
Tents		
+ GOLF EQUIPM		
+ MOUNTAINEER		
-- OUTDOOR PR		
First Aid		
Insect Repellent	static: :Jun:Jul:Aug:Sep	
Sunscreen		
+ PERSONAL AC		

6. Click (or Open) the **Price Comment** view of the Price\_and\_Cost cube.  
 7. If necessary, click **OUTDOOR PROTECTION** to expand.

8. Click **Comment** for Insect Repellents in Jan.

Golf Shop	Budget Version 1	Comment					
Products	Months						
	January	February	March	April	May	June	July
-- TOTAL PRODUCT							
+ CAMPING EQL							
+ GOLF EQUIPM							
+ MOUNTAINEER							
-- OUTDOOR PR							
First Aid							
Insect Repe							
Sunscreen	Jun	Jul	Aug	Sep			
+ PERSONAL AC							

- Click **Aug** to select.

The results appear as follows:

Golf Shop		Budget Version 1	Comment					
		Months						
Products		January	February	March	April	May	June	July
-- TOTAL PRODUCT								
+ CAMPING EQL								
+ GOLF EQUIPM								
+ MOUNTAINEER								
-- OUTDOOR PR								
First Aid								
Insect Repellent		Aug						
Sunscreen								
+ PERSONAL AC								

This pick list was defined only for the intersection of Golf Shop, Budget Version 1, Comment, January, and Insect Repellents. When you created a Pick List using the attribute, it was for all elements in the dimension. Defining a pick list in the control cube allows the definition to be much more specific.

You can leave this view open for the next task.

Click in the other cells and you will see the Picklist attribute is still in effect. However, using a rule to populate a pick list will override the attribute.

## Task 3. Create a pick list using a rule.

In the previous example, if we needed to expand the range for the pick list, we could do it manually. However, it is much easier to maintain if we write a rule to define and populate the pick list.

1. Right-click the **{PickList\_Price\_and\_Cost** cube and then click **Create Rule**.
2. On line 1, click the **Brackets** and then double-click **Products**.
3. Ctrl-click the following elements and click **OK**:
  - **CAMPING EQUIPMENT**
  - **GOLF EQUIPMENT**
  - **MOUNTAINEERING EQUIPMENT**
  - **OUTDOOR PROTECTION**
  - **PERSONAL ACCESSORIES**

```

1  [ 'CAMPING EQUIPMENT', 'GOLF EQUIPMENT',
2  'MOUNTAINEERING EQUIPMENT', 'OUTDOOR PROTECTION',
3  'PERSONAL ACCESSORIES' ] |
```

Note: You can insert returns to make the line easier to read.

4. Click to the right of the bracket, click **Insert Qualifier or Operator** and then click **=S**:
5. Type the following:  
**('subset:Months:Months');**

The first Months reference is the dimension name and the second one is the name of a subset in that dimension.

6. Click **Check Syntax**, fix any errors and then click **Save**.

```
1  [ {'CAMPING EQUIPMENT', 'GOLF EQUIPMENT',
2  'MOUNTAINEERING EQUIPMENT', 'OUTDOOR PROTECTION',
3  'PERSONAL ACCESSORIES'} ] = $: ('subset:Months:Months');
```

7. Recalculate the **Price Comment** view of the Price\_and\_Cost cube.  
 8. Click **Comment** for any of the consolidations and you should now see a drop down list of the 12 months in the Months subset.

The results appear as follows:

		Months							
Products		January	February	March	April	May	June	July	August
-- TOTAL PRODUCT									
+ CAMPING EQL									
+ GOLF EQUIPM									
+ MOUNTAINEER									
-- OUTDOOR PR									
First Aid		Jan							
Insect Repellent	A	Feb							
Sunscreen		Mar							
+ PERSONAL AC		Apr							
		May							
		Jun							
		Jul							
		Aug							
		Sep							
		Oct							
		Nov							
		Dec							

9. Close the **Price Comment** view and save changes.
10. Close the rules for the **{PickList\_Price\_and\_Cost}** cube.
11. Close the **{PickList\_Price\_and\_Cost}** cube and do not save changes.
12. In Server Explorer, click **View** and then click **Display Control Objects**.  
This will turn off the display of control objects.
13. Close TM1 Architect and any open windows.

**Results:**

**You added more pick lists to the Comment elements in the Price\_and\_Cost cube using control cubes and rules. You added both a static and a subset pick list.**

# Using Rule Worksheets

- Use worksheets to:
  - keep versions of rules together.
  - make rules easier to read with Microsoft Excel formatting options.
  - enhance rules with Microsoft Excel functions.



Cognos  
software

Do not use both the rule worksheets and the Rules Editor for the same cube as this may cause data to be lost. Write rules with either the Rules Editor or a Rule Worksheet.

When saving rules in a worksheet, you must use Save Rule from the TM1 menu to save the file. Using the TM1 menu saves the \*.xru file and generates the \*.rux file from column A. The rules editor only reads/writes to the \*.rux (compiled) file.

Using File - Save from the MS Excel menu will save only the \*.xru file without compiling the \*.rux file. If the rules are then opened in the Rules Editor, changes in the \*.xru file will not be reflected and business logic may be lost.

## Demo 5: Review a Rule Worksheet (Instructor Only)

### Purpose:

You want to look at rules in rule worksheets. You will open the rules for the Rules cube and review the format in Excel.

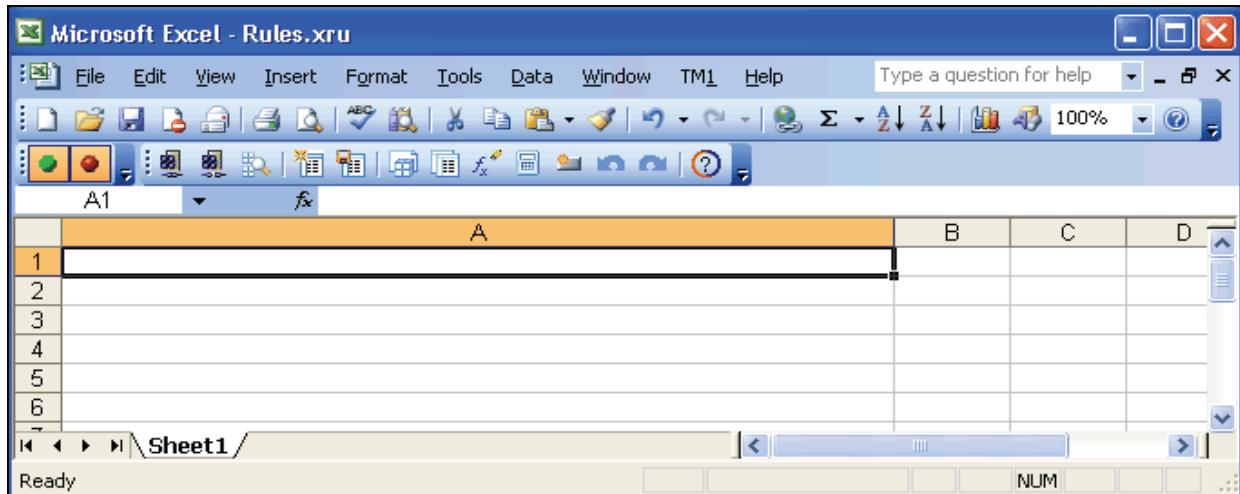
### Task 1. Open a rule worksheet.

1. Start the **greatoutdoors** server if it is not already running.
2. Click **Start > All Programs > IBM Cognos > TM1** and then click **Perspectives for MS Excel**.
3. Click **Enable Macros**, if prompted.
4. In **Microsoft Excel**, from the **TM1** menu, point to **Network**, and then click **Connect**.
5. Enter the following:
  - Server Id: **greatoutdoors**
  - Client Id: **admin**
  - Password: <none>
6. Click **OK** to close the Connect to TM1 Server window, and then click **OK** to close the confirmation window.
7. From the **TM1** menu, point to **Rule WorkSheets**, and then click **New**.

8. Click **greatoutdoors:Rules**, and then click **OK**.

Rules.xru opens. Notice that column A is very wide.

Only statements placed in column A are compiled. You may place comments or older versions of rules in other columns to document them.



Rules created in the Rules Editor will not appear. Rule Worksheets read in and write to the worksheet (\*.xru) files. This is why you should only create or edit rules using either the Rules Editor or Rules Worksheets for a cube. You should never use both editors for a single cube.

You write rules in Column A using the same format as with the Rules Editor.

Since only column A is compiled, it can be useful to use the additional columns for comments or versioning of your rules.

9. Close the rule worksheet and do not save it.

10. Close MS Excel.

**Result:**

**You reviewed steps to create a rule worksheet for the Rules cube.**

## Determine Location of Logic

Location	Use for:
Hierarchies & Weighting	Core calculations
Rules	Formal modeling with comments
TI	ETL, reports & KPIs, performance optimization
Spreading	Spontaneous modeling
Excel	Ad-hoc, specialized
VBA	Usability (not business logic)
VBA and API	Specialized solutions

 Cognos software

You have looked at a number of options for calculating data. Understanding these options will allow you to decide where to add business logic to best suit the needs of your users and your data. The calculations may be performed within the dimension hierarchy, when data is pulled into a cube via TI, or in TM1 rules.

In order to optimize performance of the model, what can be processed on the server instead of in spreadsheets and local files, should be processed there.

- Dimension Hierarchies, Cube Rule, TI process
- Define complex SQL queries and views in the RDBMS

When looking at the model, ask does this really need to be computed dynamically or can it be stored? It is also a good idea to review the model periodically asking this question.

### Instructor Notes

Dimension hierarchy can perform calculations using weighting. To have a consolidation do subtraction you can assign an element a weight of -1. This will be multiplied by the value associated with this element (giving it a negative value) upon calculating the consolidation.

## Demo 6: Calculate Gross Margin

### Purpose:

You need to calculate Gross Margin and Margin % in the Sales\_Plan cube. You know you can use rules to do this but the calculation is simple addition and subtraction. Instead of writing a rule, you determined the best place to calculate Gross Margin is within the dimension hierarchy (using weighting).

### Task 1. Edit the dimension structure.

TM1 Server: **greatoutdoors**

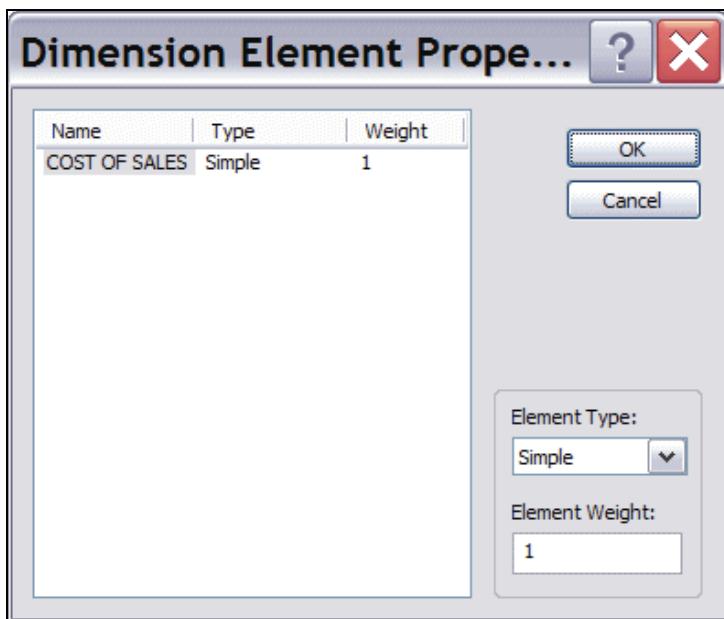
TM1 Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Launch the TM1 Architect: Server Explorer.
2. Expand **TM1**, double-click **greatoutdoors**, in the username type **admin**, and then click **OK**.
3. Right-click the **Sales\_Plan\_Measures** dimension and click **Edit Dimension Structure**.
4. Click **NET SALES REVENUE**, and then from the **Edit** menu, click **Cut**.  
You could also copy the element if you wished to see the value both within the Gross Margin and as a stand alone element. Both would display the same value.
5. Click **OK**.
6. Click **GROSS MARGIN**, and then from the **Edit** menu, click **Paste as Child**.  
The sigma (summation) icon appears in front of GROSS MARGIN, and NET SALES REVENUE appears below it.
7. Repeat steps 4 to 6 with **COST OF SALES**, pasting it as a child of **GROSS MARGIN**.

8. Right-click **COST OF SALES**, and then click **Element Properties**.



Here you may change the type of element or its weight. Consolidations are summations by default. Giving a child element a negative weight is how subtraction may be achieved in the consolidation.

9. In the **Element Weight** box type, **-1**, and then click **OK**.

## 10. Click **GROSS MARGIN**.

The element list shows COST OF SALES with a weight of -1.

	Name	Security Owner	Type	Level	Weight
	n NET SALES REVENUE		Simple	0	1
	n COST OF SALES		Simple	0	-1

The screenshot shows the SAP BusinessObjects Dimension Editor. On the left, there's a tree view of elements under a node labeled 'Σ GROSS MARGIN'. The elements listed are: Quantity, Unit Sale Price, GROSS SALES REVENUE, Returns and Allowances, Cash Discount, Volume Discount, TOTAL DISCOUNTS AND ALLOWANCES, Freight, Unit Cost, and three leaf nodes: NET SALES REVENUE, COST OF SALES, and GROSS MARGIN %. The 'COST OF SALES' node has a weight of -1. To the right of the tree view is a table with columns: Name, Security Owner, Type, Level, and Weight. It contains two rows: one for 'NET SALES REVENUE' (Simple type, Level 0, Weight 1) and one for 'COST OF SALES' (Simple type, Level 0, Weight -1).

The consolidation Gross Margin is a summation of Net Sales Revenue - Cost of Sales, and is calculated without feeders.

## 11. Click **Save** and then close the Dimension Editor.

### Result:

**You calculated Gross Margin in the dimension hierarchy using weightings in the consolidation algorithm.**

## Summary

- At the end of this module, you should be able to:
  - discuss a rule
  - construct rules for elements or consolidations
  - use functions in rules
  - populate a picklist using a rule
  - discuss rules in a Rule Worksheet

Cognos.  
software



## Workshop 1: Calculate Average Unit Cost

Now that you have computed Average Unit Price for consolidations in the Price\_and\_Cost cube, you will need to compute the Average Unit Cost for consolidations. Create the rules and review the consolidated Unit Cost values to ensure they are the average of the base level elements.

To accomplish this:

- Create new elements in Price\_and\_Cost\_Measures.
  - UnitCost\_hold
  - UnitCost\_count
- Add a rule to hold the Unit Cost data.
- Add a rule to count the number of non-empty cells in Unit Cost.
- Add a rule to compute the average unit cost at the consolidated level.

For more detailed information outlined as tasks, see the Task Table section.

For the final results, see the Workshop Results section that follows the Task Table section.

## Workshop 1: Task Table

Task	Where to Work	Hints
1. Create new elements in Price_and_Cost_Measures.	Dimension Editor	<ul style="list-style-type: none"> <li>Insert new elements:           <ul style="list-style-type: none"> <li>UnitCost_hold</li> <li>UnitCost_count</li> </ul> </li> </ul>
2. Create a rule for UnitCost_hold.	Rules Editor	<ul style="list-style-type: none"> <li>[UnitCost_hold]=N:[Unit Cost];</li> </ul>
3. Create a rule for UnitCost_count.	Rules Editor	<ul style="list-style-type: none"> <li>[UnitCost_count]= N: IF ([Unit Cost] &lt;&gt;0,1,0);</li> </ul>
4. Create a rule for the Average Unit Cost for consolidated cells.	Rules Editor	<ul style="list-style-type: none"> <li>[Unit Cost] = C: [UnitCost_hold] \ [UnitCost_count];</li> </ul>
5. Review the default view.	Cube Viewer	<ul style="list-style-type: none"> <li>Price_and_Cost cube.</li> <li>Default view.</li> <li>Expand CAMPING EQUIPMENT.</li> </ul>

If you need more information to complete a task, see the Step-by-Step instructions at the end of the Workshop.

## Workshop 1: Workshop Results

At the end of Task 5, the rules editor will appear as shown below:

```

1  #hold the Unit Sale Price for the Average Price
2  ['UnitPrice_hold' ] = N: ['Unit Sale Price' ] ;
3  ['UnitPrice_count' ] = N: IF(['Unit Sale Price'] <> 0, 1, 0);
4  ['Unit Sale Price' ] = C: ['UnitPrice_hold' ] \ ['UnitPrice_count' ] ;
5  #Calculate Average Unit Cost for consolidated levels.
6  ['UnitCost_hold' ] = N: ['Unit Cost' ] ;
7  ['UnitCost_count' ] = N: IF(['Unit Cost' ] <> 0,1,0 );
8  ['Unit Cost' ] = C: ['UnitCost_hold' ] \ ['UnitCost_count' ] ;

```

At the end of the Workshop, the result will appears as shown below:

Golf Shop	Budget Version 1	Unit Cost					
Products	Months						
	Jan	Feb	Mar	Apr	May	Jun	Jul
-- TOTAL PRODUCT	\$109.79	\$109.79	\$109.79	\$109.79	\$109.79	\$109.79	\$109.79
-- CAMPING EQ	\$128.11	\$128.11	\$128.11	\$128.11	\$128.11	\$128.11	\$128.11
Cooking Gear	\$29.87	\$29.87	\$29.87	\$29.87	\$29.87	\$29.87	\$29.87
Lanterns	\$21.14	\$21.14	\$21.14	\$21.14	\$21.14	\$21.14	\$21.14
Packs	\$126.51	\$126.51	\$126.51	\$126.51	\$126.51	\$126.51	\$126.51
Sleeping Bag	\$78.01	\$78.01	\$78.01	\$78.01	\$78.01	\$78.01	\$78.01
Tents	\$385.04	\$385.04	\$385.04	\$385.04	\$385.04	\$385.04	\$385.04

## Workshop 1: Step-by-Step Instructions

### Task 1. Create new elements in Price\_and\_Cost\_Measures.

1. In Server Explorer, right-click the **Price\_and\_Cost\_Measures** dimension, click **Edit Dimension Structure**, and then click **All**.
2. Click **UnitPrice\_count**, and then from the **Edit** menu, click **Insert Element**.
3. In Insert Element Name, type **UnitCost\_hold** and then click **Add**.
4. In Insert Element Name, type **UnitCost\_count** and then click **Add**.
5. Click **OK**, click **OK** again and then click **Yes** to save the changes.

### Task 2. Create a rule for UnitCost\_hold.

1. Double-click the rules file for the **Price\_and\_Cost** cube to open the Rules Editor.
2. Place your cursor on line 5 and add the following comment:  
**#Calculate Average Unit Cost for consolidated levels.**
3. On line 6, type the following:  
**['UnitCost\_hold'] = N: ['Unit Cost'];**
4. Press **Enter**.

### Task 3. Create a rule for UnitCost\_count.

1. Type the following:  
**['UnitCost\_count'] = N: IF (['Unit Cost'] <>0,1,0);**
2. Click **Check Syntax**.
3. Press **Enter**.

## Task 4. Create a rule for the Average Unit Cost for consolidated cells.

1. Type the following:

```
['Unit Cost'] = C: ['UnitCost_hold'] \ ['UnitCost_count'] ;
```

2. Press **Enter**.
3. Click **Save**.
4. Close the rules editor.

## Task 5. Review the default view.

1. In Server Explorer, double-click the **Price\_and\_Cost** cube.
2. In the Price\_and\_Cost\_Measures drop down list, click **Unit Cost**.
3. Recalculate the view, expand **CAMPING EQUIPMENT**, and then review the results.
4. Close the Cube Viewer without saving if prompted.

## Workshop 2: Complete Calculations for Sales\_Plan\_Measures

The Sales\_Plan\_Measures are still not complete. Now that you have brought the Unit Sale Price and Unit Cost measures into the cube, you want to finish calculating the rest of the measures. You will take advantage of the different calculation options TM1 has to offer.

To accomplish this:

- Change the dimension structure to:
  - Net Sales Revenue will aggregate:
    - Gross Sales Revenue
    - Total Discounts and Allowances (weight of -1)
    - Freight (weight of -1)
  - Total Discounts and Allowances will aggregate:
    - Volume Discount
    - Cash Discount
    - Total Returns and Allowances
- Create a rule to bring Unit Cost to the Sales\_Plan cube.
- Create a rule to calculate Cost of Sales.
  - Quantity\*Unit Cost
- Calculate a view of the Sales\_Plan cube.

For more detailed information outlined as tasks, see the Task Table section.

For the final results, see the Workshop Results section that follows the Task Table section.

## Workshop 2: Task Table

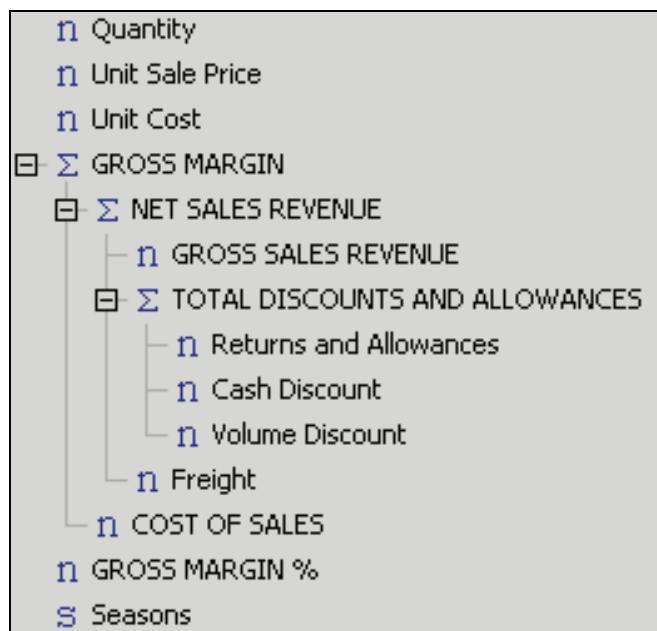
Task	Where to Work	Hints
1. Edit the Sales_Plan_Measures dimension.	Dimension Editor / Sales_Plan_Measures	<ul style="list-style-type: none"> <li>• Cut and Paste as children of NET SALES REVENUE:           <ul style="list-style-type: none"> <li>• GROSS SALES REVENUE</li> <li>• TOTAL DISCOUNTS AND ALLOWANCES (weight of -1)</li> <li>• Freight (weight of -1).</li> </ul> </li> <li>• Cut and Paste as children of TOTAL DISCOUNTS AND ALLOWANCES:           <ul style="list-style-type: none"> <li>• Volume Discount</li> <li>• Cash Discount</li> <li>• Returns and Allowances</li> </ul> </li> </ul>
2. Create a rule for Unit Cost.	Rules Editor / Sales_Plan	<ul style="list-style-type: none"> <li>• ['Unit Cost'] = N:DB('Price_and_Cost',!Channels,!Products,!Months,!Versions,'Unit Cost');</li> <li>• C:[COST OF SALES] \['Quantity'];</li> </ul>

3. Create rules for Cost of Sales and Gross Sales Revenue.	Rules Editor / Sales_Plan	<ul style="list-style-type: none"> <li>• ['GROSS SALES REVENUE']=N:[Quantity]*[Unit Sale Price];</li> <li>• ['COST OF SALES'] = N:[Quantity] * [Unit Cost];</li> </ul>
4. Calculate the Revenue view of the Sales_Plan cube.	Cube Viewer	<ul style="list-style-type: none"> <li>• Open the Revenue view</li> <li>• Cube: Sales_Plan</li> </ul>

If you need more information to complete a task, see the Step-by-Step instructions at the end of the Workshop.

## Workshop 2: Workshop Results

The Sales\_Plan\_Measures dimension appears as follows:



The results appear as follows:

```

['Unit Sale Price'] = N: DB('Price_and_Cost', !Channels,
    !Products, !Months, !Versions, 'Unit Sale Price');
    C: ['GROSS SALES REVENUE'] \ ['Quantity'];
['Unit Cost'] = N: DB('Price_and_Cost', !Channels, !Products,
    !Months, !Versions, 'Unit Cost');
    C: ['COST OF SALES'] \ ['Quantity'];
['GROSS SALES REVENUE'] = N: ['Quantity'] * ['Unit Sale Price'];
['COST OF SALES'] = N: ['Quantity'] * ['Unit Cost'];
  
```

The Sales\_Plan cube as follows:

		Months			
Sales_Plan_Measu		Jan	Feb	Mar	Apr
Quantity		47,742.00	11,935.00	43,090.00	17,024.00
Unit Sale Price		\$42.33	\$42.33	\$42.33	\$42.33
Unit Cost		\$31.36	\$31.36	\$31.36	\$31.36
-- GROSS MARGIN		\$124,549.09	\$31,131.18	\$112,406.93	\$44,398.68
-- NET SALES RE		\$1,621,905.31	\$405,454.55	\$1,463,860.14	\$578,330.90
GROSS SALE		\$2,020,704.02	\$505,154.84	\$1,823,805.80	\$720,549.31
-- TOTAL DIS		\$195,960.34	\$48,990.80	\$176,869.07	\$69,882.65
Returns		\$88,190.61	\$22,047.61	\$79,598.51	\$31,450.34
Cash Disc		\$65,239.03	\$16,310.20	\$58,883.46	\$23,265.18
Volume D		\$42,530.70	\$10,632.99	\$38,387.10	\$15,167.13
Freight		\$202,838.37	\$50,709.49	\$183,076.58	\$72,335.76
COST OF SALES		\$1,497,356.22	\$374,323.37	\$1,351,453.21	\$533,932.22
GROSS MARGIN %		0.00	0.00	0.00	0.00
Seasons					

## Workshop 2: Step-by-Step Instructions

### Task 1. Edit the Sales\_Plan\_Measures dimension.

1. In Server Explorer, right-click **Sales\_Plan\_Measures** and then click **Edit Dimension Structure**.
2. Ctrl+click **GROSS SALES REVENUE**, **TOTAL DISCOUNTS AND ALLOWANCES**, and **Freight** to select.
3. Ctrl+X to cut and then click **NET SALES REVENUE**.
4. From the **Edit** menu, click **Paste as Child**.
5. Right-click **TOTAL DISCOUNTS AND ALLOWANCES** (under **NET SALES REVENUE**) and then click **Element Properties**.
6. In the **Element Weight** box type **-1** and then click **OK**.
7. Repeat steps **5 - 6** for **Freight**.
8. Ctrl+click **Volume Discount**, **Cash Discount** and **Returns and Allowances** to select.
9. Ctrl+X to cut and then click **TOTAL DISCOUNTS AND ALLOWANCES** (under **NET SALES REVENUE**).
10. From the **Edit** menu, click **Paste as Child**.
11. Click **Set Dimension order**.
12. Click **OK** and then click **Yes** to save changes to the dimension.

### Task 2. Create a rule for Unit Cost.

1. Open the rules file for the **Sales\_Plan** cube.
2. Start a new line, and then type the following:  
**[Unit Cost] = N:**

3. Click **Insert Cube Reference**.
4. In the Cube list, click the **Price\_and\_Cost** cube.
5. Click the **Subset Editor** icon next to **Price\_and\_Cost\_Measures**, click **Unit Cost**, and then click **OK** twice.
6. Type a ; at the end of the line, and then press **Enter**.
7. Type the following:  
**C:[COST OF SALES] \ ['Quantity'];**
8. Press **Enter**.

### **Task 3. Create rules for Cost of Sales, and Gross Sales Revenue.**

1. Type the following rule:  
**[GROSS SALES REVENUE]=N:['Quantity']\*['Unit Sale Price'];**
2. Press **Enter**.
3. Type the following rule:  
**[COST OF SALES] = N:['Quantity']\*['Unit Cost'];**
4. Click **Save**.

### **Task 4. Calculate the Sales\_Plan cube.**

1. Open the **Sales\_Plan** cube.
2. In the **Select View** list, select **Revenue**.
3. Click **Sales\_Plan\_Measures**, click **All** and then click **OK**.
4. Click **Recalculate**.

Unit Cost should now contain values and the Sales\_Plan\_Measures should be reorganized.

5. Close all open windows, click **Yes** to save changes and overwrite the view and save changes when shutting down the greatoutdoors server.

**Information Management**



## **Optimize Rule Performance**

IBM Cognos TM1 9.5



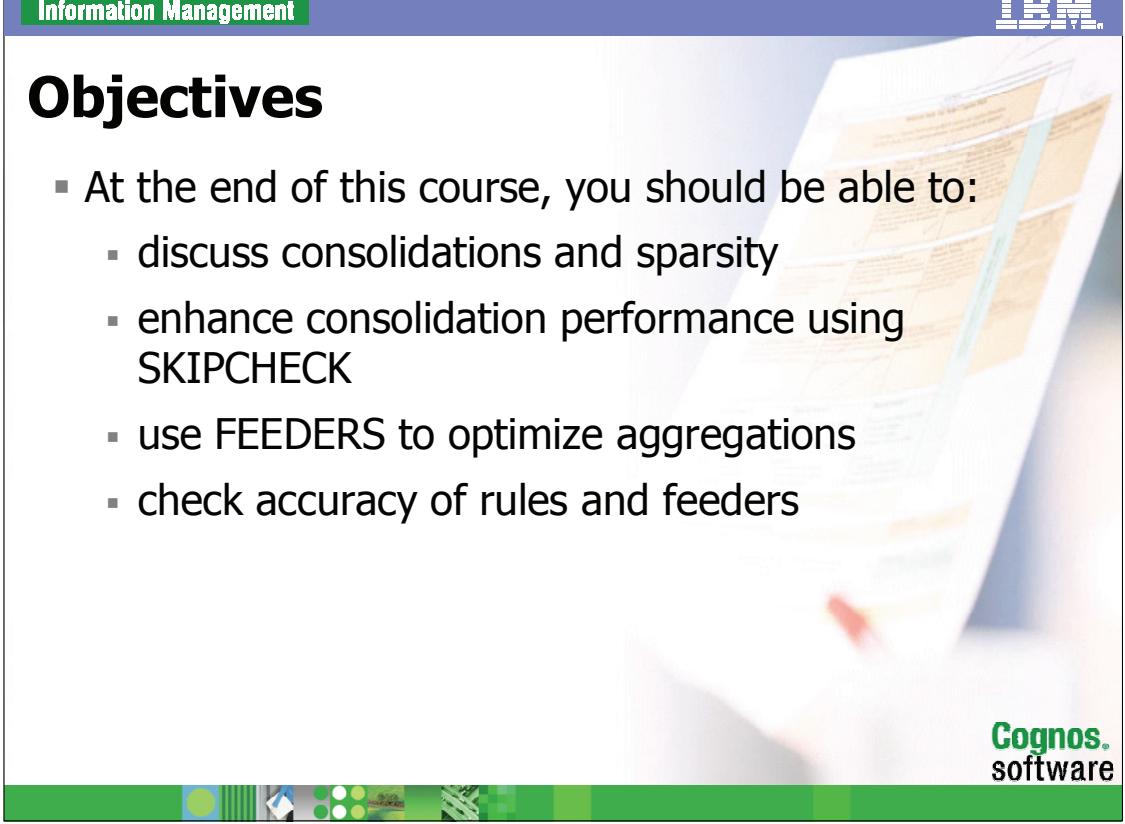
**Cognos.**  
software

© 2009 IBM Corporation



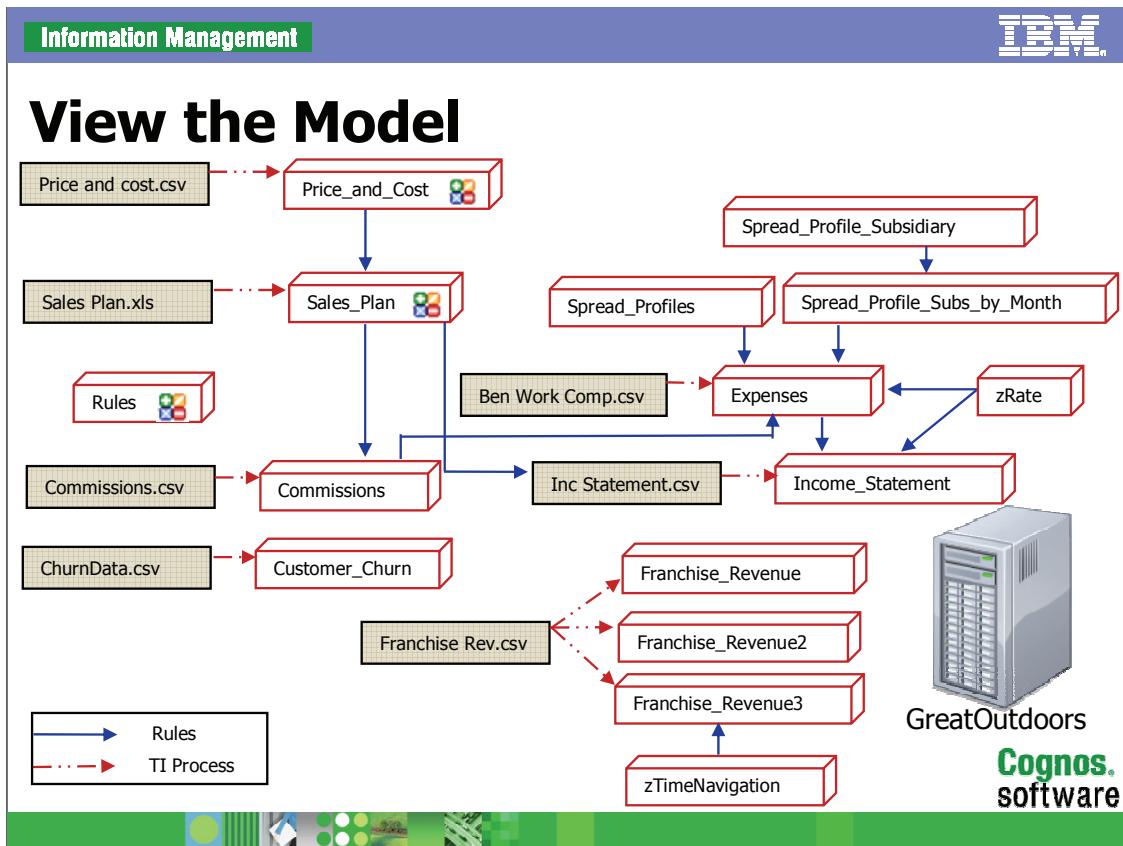
# Objectives

- At the end of this course, you should be able to:
  - discuss consolidations and sparsity
  - enhance consolidation performance using SKIPCHECK
  - use FEEDERS to optimize aggregations
  - check accuracy of rules and feeders



Cognos.  
software





In this course you will not create any new cubes; instead you will modify existing rules and cubes.

## Understanding Consolidations and Sparsity

- The consolidation engine skips empty cells or zeros.
- TM1 generates a zero when a stored or calculated value is not found.
- Example: not all products are sold all year:

	Total Year	Q1	Q2	Q3	Q4
Total Products	130	10	20	40	70
Product1	40	10	0	10	20
Product2	20	0	20	0	0
Product3	40	0	0	30	10
Product4	40	0	0	0	40

 Cognos  
software

In the above example there are four Quarters in the time dimension and four Products in the products dimension. There are a total of 16 values ( $4*4=16$ ) in the cube.

However, TM1 only uses 7 of the 16 cells when aggregating the consolidated values (like Total Year or Total Products).

If you have 100 products, 60 time periods, 20 markets and 5 measures you need to track, you would have ( $100*60*20*5=600,000$ ) 600,000 potential values in your cube. In a multidimensional model, this is still a small example and it is very likely that most of these will be zero. It is much faster to look at 1% of the values (6,000) instead of evaluating 600,000 values.

By avoiding the consolidation of zero values, TM1 also avoids the explosion in the size of the cube that can arise if zero values at the base level are consolidated to create even more zero values at consolidated levels.

**INTERACTION - Highlighter:** Ask for a volunteer to highlight the zeros in the table.

## Effect of Rules on Sparse Consolidation

- Cells populated by rules are seen as having no value by the consolidation engine.
- Once rules have been added to the cube:
  - TM1 needs to CHECK every cell to see if there is a Rule that applies to it.
  - TM1 needs to evaluate that Rule to see if it will supply a non-zero value.
  - TM1 will then need to consolidate it.



Cognos  
software

Rule Checking occurs as soon as a rule file is associated with a cube, even if the file contains no rules.

This will significantly increase the time it takes to aggregate data for consolidations.

You can turn off this cell CHECKing by using the SKIPCHECK statement in the rule file. This will also turn off the consolidation speed of TM1. In many models, less than 1 in a million cells are actually populated.

## Optimize your Rules using SKIPCHECK

- SKIPCHECK;
  - add it to the top of the Rule file.
  - turns off checking all cells.
  - consolidates in the normal, efficient manner.
- Values stored at the base level calculate correctly.
- Figures at the consolidated levels of the hierarchy will be zero if figures at the base are derived by a rule.



In this course we are working with a limited amount of data. In your model, you may have much more data to calculate and consolidate. This may cause the aggregation engine, which has performed very well, to slow significantly.

In a large cube the number of zero values will typically be very high. The ratio between the number of non-zero cells in a cube and the total number of cells in a cube is called its sparsity. Often times, there are more zero values in a cube than non-zero values.

Zeroes are a fact of life in multi-dimensional cubes. Managing those zeroes correctly is very important in multi-dimensional modeling tools. To avoid wasting space, TM1 never stores a zero value. TM1 simply generates a zero when someone requests a value for a combination of elements that does not have a figure. As TM1 does not store zero values, this reduces the storage requirements for large cubes.

## SKIPCHECK with Consolidations

- The last two columns contain calculations
- The Total Products consolidation ignores the values in them:

	Total Year	Q1	Q2	Q1+Q2	Q3+Q4
Total Products	30	10	20	0	0
Product1	10	10	0	10	20
Product2	20	0	20	20	0
Product3	0	0	0	0	10
Product4	0	0	0	0	40

Cognos  
software

To correct this problem where consolidations are displayed as zeros, you need to use FEEDERS.

Feeders are a way of telling TM1 which calculated cells will receive non-zero values via Rules, so TM1 knows where there are non-zero values to be consolidated. This improves the speed of consolidations while preserving accuracy.

Use the FEEDERS; statement in a rule file below all rules statements, and then follow it with statements indicating exactly which cells will contain values used in rules.

Assume Total Year consolidates Q1 and Q2 separately and not Q1+Q2 and Q3+Q4. FYI: Q1+Q2 not consolidated, but calculated by a rule.

**INTERACTION - Markup > My Arrow Tool: Use a markup tool to add arrows to the zeros in consolidations (or ask a participant to do it).**

## Discuss a Feeder statement

- Example rule:
  - `['Revenue']=N:['Quantity']*['Unit Price'];`
- Example feeder:
  - FEEDERS;
  - `['Quantity']=>['Revenue'];`
- This feeder tells TM1 to place an internal marker in the Revenue cell if there is a value in Quantity.
- The marker is seen by the consolidation engine when consolidating values for the Revenue rule.

 Cognos  
software

Feeders are activated when the cube is first loaded into memory, and thereafter, whenever a value is changed in the cube. This is the opposite of Rules, which are only activated when a value is requested by a user. You can think of Rules as pulling values in, while Feeders push markers out. Once a feeder is fired, it does not fire again until the server is restarted or the cube is reloaded.

When writing Feeders you need to ask the question “Which element or combination of elements on the right hand side of the Rule determines whether there will be a value on the left hand side of the Rule”.

Writing Feeders for rules is known as "Feeding" cells.

---

In this example, you would feed 'Revenue' based on 'Quantity' because the Quantity would only contain a value if there were actually sales made. Unit Price is more likely to contain values even if there were no quantities sold. Quantity is more likely to contain zeros and allow TM1 to look at fewer cells so it is a better choice.

## Using Feeder Statements

- Feed every **simple** cell that is calculated by a rule
  - feed this:  $[X]=N:[Y]*[Z]$ ;
  - not this:  $[X]=C:[Y]*[Z]$ ;
- Do not feed rules that apply to consolidated cells
  - unless using Suppress Zeros
- Do not feed more cells than you need
  - Results in overfeeding

**Cognos**  
software



We need to be careful about over-feeding, for example feeding more cells than is necessary, in other words feeding cells whose result will be zero anyway.

Remember that, when you feed a cell, a small marker is stored in that cell, which takes up a small amount of memory. Feeding a cell also means that TM1 will evaluate any rule associated with this cell when it is consolidating, even if that rule is going to return a zero value.

Feeders take up processor time, particularly when the cube is first loaded, since at that time the processor has to evaluate all the feeders. Thereafter, the burden is lower, since TM1 only needs to evaluate feeders when a value changes, and even then, only those feeders related to the value that was changed.

Chronic over-feeding, will cause performance problems since it effectively returns us to the state that we were in, before we added the SKIPCHECK statement.

## Using Feeder Statements (cont'd)

- In multiplication or division, feed using the least data
  - $[Revenue]=[Quantity]*[Price];$
  - $[Quantity]=>[Revenue];$
- In addition, subtraction, or logic, feed using each section
  - $[HalfYear]=[Q1]+[Q2];$
  - $[Q1]=>[HalfYear]; [Q2]=>[HalfYear];$

A little over-feeding is not normally a problem. When in doubt it is best to over-feed. Under-feeding, i.e. failing to feed cells for calculated results that will be non-zero, will result in incorrect consolidation of results.

You may need to feed consolidations as well as base level cells when users turn on Suppress Zeros for a view. Suppress hides all values displaying only zeros in the entire row or column. If both the base level and consolidated level calculation is not fed, suppress zeros may hide the columns or rows containing calculations.

---

Suppress Zeros is displayed later in the module.

**INTERACTION - Line > Arrow:** Use an arrow to draw a line from Quantity in the Rule to Quantity in the Feeder. Have a participant do it for Q1 & Q2 above.

## Inter-cube Feeders

Feeder Cube (Source)				
	Jan	Feb	Mar	Apr
P1	1 ➔	2 ➔	8 ➔	1
P2	2 ➔	4 ➔	6 ➔	3
P3	3 ➔	6 ➔	4 ➔	5
P4	4 ➔	8 ➔	2 ➔	7

Rules Cube				
	Q1	Jan	Feb	Mar
P1	11	1 DB()	2 DB()	8 DB()
P2	12	2 DB()	4 DB()	6 DB()
P3	13	3 DB()	6 DB()	4 DB()
P4	14	4 DB()	8 DB()	2 DB()

- Rules can 'pull' data into cells from other cubes.
- Feeders put a marker in the source cells and 'push' it to the calculated cell in the target cube.

Cognos  
software



While rules reside in the target cube, feeders generally reside in the source cube.

The rule is attached to the cube that will display the result.

Use the DB () function to reference the cell(s) containing the data and pull in the data.

This is because of the "push" nature of feeders. The feeder statement tells the system to feed cells when a value changes in the source cells. If the source cells are in another cube, the feeder statement needs to be there too.

The feeder statements are attached to the cube that contains the source data and push the marker to the target calculation.

**INTERACTION - Line > Line: Use a line and draw a line from a cell in the Source to the Target**

## Demo 1: Add SKIPCHECK and FEEDERS

### Purpose:

You want to improve the performance of calculations in the Price\_and\_Cost cube. To do this you will add SKIPCHECK and FEEDERS to your rules in this cube.

### Task 1. Determine which rules must be fed.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. In the Server Explorer window, open the **Default** view of the **Price\_and\_Cost** cube.

#### 4. Drill down on CAMPING EQUIPMENT.

The result appears as shown below:

		Budget Version 1	Unit Sale Price					
		Months						
Products		Jan	Feb	Mar	Apr	May	Jun	Jul
-- TOTAL PRODUCT		\$166.54	\$166.54	\$166.54	\$166.54	\$166.54	\$166.54	\$166.54
-- CAMPING EQL		\$160.41	\$160.41	\$160.41	\$160.41	\$160.41	\$160.41	\$160.41
Cooking Gear		\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31
Lanterns		\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12
Packs		\$175.37	\$175.37	\$175.37	\$175.37	\$175.37	\$175.37	\$175.37
Sleeping Bag		\$103.83	\$103.83	\$103.83	\$103.83	\$103.83	\$103.83	\$103.83
Tents		\$453.41	\$453.41	\$453.41	\$453.41	\$453.41	\$453.41	\$453.41
+ GOLF EQUIPM		\$431.14	\$431.14	\$431.14	\$431.14	\$431.14	\$431.14	\$431.14

#### 5. Open the Rules Editor for Price\_and\_Cost.

#### 6. Insert a new line at the beginning of row 1 and type SKIPCHECK;

```
SKIPCHECK;
# hold the Unit Sale Price for the average price
['UnitPrice_hold' ] = N:[ 'Unit Sale Price' ];
['UnitPrice_count' ] = N:IF([ 'Unit Sale Price' ] <>0, 1, 0);
['Unit Sale Price' ] = C: ['UnitPrice_hold' ] \ ['UnitPrice_count' ];
# Calculate Average Unit Cost for consolidated levels
['UnitCost_hold' ] = N: [ 'Unit Cost' ];
['UnitCost_count' ] = N: IF([ 'Unit Cost' ] <>0, 1,0);
['Unit Cost' ] =C: ['UnitCost_hold' ] \ ['UnitCost_count' ];
```

- Save the rule and recalculate the view.

The screenshot shows a TM1 interface with three dropdown menus at the top: 'Golf Shop', 'Budget Version 1', and 'Unit Sale Price'. Below these is a table titled 'Months' with columns for Jan through Jul. The rows include 'Products' (Cooking Gear, Lanterns), a summary row for '-- TOTAL PRODUCT', and a row for '-- CAMPING EQL'. The data values are as follows:

Products	Jan	Feb	Mar	Apr	May	Jun	Jul
-- TOTAL PRODUCT	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
-- CAMPING EQL	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Cooking Gear	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31
Lanterns	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12

The consolidation no longer calculates the average because TM1 does not see the calculated values in the N: level elements UnitPrice\_hold and UnitPrice\_count. They are calculations, stored only in memory, TM1 does not know they contain values to compute and are seen as 0 when aggregating consolidated values.

You will need to feed those expressions so the values for these cells can be used for other calculations (like the average Unit Sale Price).

You will feed the N: level calculations for UnitPrice\_hold and UnitPrice\_count using Unit Sale Price.

## Task 2. Add the feeder statement.

- Create a blank row at the bottom of the rules on line 11 type **FEEDERS**; and press **Enter**.
- Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
- Click **Unit Sale Price** and then click **OK** to return to the rules.
- Click to the right of the bracket, click **Insert Qualifier or Operator** and then click **=>**.
- Click **Brackets** and then double-click **Price\_and\_Cost\_Measures**.
- Click **UnitPrice\_hold** and then click **OK** to return to the rules.

7. Click to the right of the bracket, type a , (comma), click **Brackets**, double-click **Price\_and\_Cost\_Measures** and then select **UnitPrice\_count**.
8. Click **OK** to return to the rules, click to the right of the bracket, and then add a ; to the end of the line.

In this example you would feed both of the rules with the same element so you can reference them both in a single line by separating them with commas.

The result appears as follows:

```

1  SKIPCHECK;
2  #hold the Unit Sale Price for the Average Price
3  ['UnitPrice_hold'] = N: ['Unit Sale Price'] ;
4  ['UnitPrice_count'] = N: IF(['Unit Sale Price'] <> 0, 1, 0);
5  ['Unit Sale Price'] = C: ['UnitPrice_hold'] \ ['UnitPrice_count'] ;
6  #Calculate Average Unit Cost for consolidated levels.
7  ['UnitCost_hold'] = N: ['Unit Cost'] ;
8  ['UnitCost_count'] = N: IF(['Unit Cost'] <> 0,1,0 );
9  ['Unit Cost'] = C: ['UnitCost_hold'] \ ['UnitCost_count'] ;
10
11 FEEDERS;
12 ['Unit Sale Price'] => ['UnitPrice_hold'], ['UnitPrice_count'];

```

- Click **Save**, and then recalculate the view.

The view should now consolidate properly.

The result appears as shown below:

Golf Shop	Budget Version 1	Unit Sale Price					
Months							
Products	Jan	Feb	Mar	Apr	May	Jun	Jul
-- TOTAL PRODUCT	\$166.54	\$166.54	\$166.54	\$166.54	\$166.54	\$166.54	\$166.54
-- CAMPING EQL	\$160.41	\$160.41	\$160.41	\$160.41	\$160.41	\$160.41	\$160.41
Cooking Gear	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31	\$40.31
Lanterns	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12	\$29.12

As you add more data to your cube, it will calculate more efficiently with the feeders that you added, than if you did not add the feeder statements.

- Close the Rules Editor and the Cube Viewer without saving changes.

## Results:

You improved the performance of calculations in the Price\_and\_Cost cube by adding SKIPCHECK and FEEDERS to your rules in the cube.

## Instructor Notes

Our sample data is very small so we did not see a large performance impact without proper feeders. However, it is much more common to see a large increase in calculation time when opening views of a cube that have just added rules. This has caused many people to avoid using rules. Rules are a very powerful addition to a cube. It is recommended that you consider feeders while you write your business rules.

# Troubleshooting Rules and Feeders

- Use the Rules Tracer to troubleshoot rules and feeders in TM1.
  - Trace Calculations
  - Trace Feeders
  - Check Feeders
- Rules Tracer has two panes:
  - Top (current cell definition or calculation)
  - Bottom (components of calculation or feeder)

 Cognos  
software

Right-click any cell and choose Trace Calculation to trace it to the base or leaf level elements.

Trace Feeders checks the way the selected cell feeds other cells. Trace Feeders is only enabled for base or leaf level cells but not consolidated cells.

If a cube contains a rule with SKIPCHECK and FEEDERS statements, you can use the Rules Tracer to Check Feeders to check that TM1 properly feeds the components of the consolidation.

- If the Trace pane is empty, it means the consolidation is fed, otherwise you will see 'not fed' displayed.

## Demo 2: Troubleshoot rules and feeders

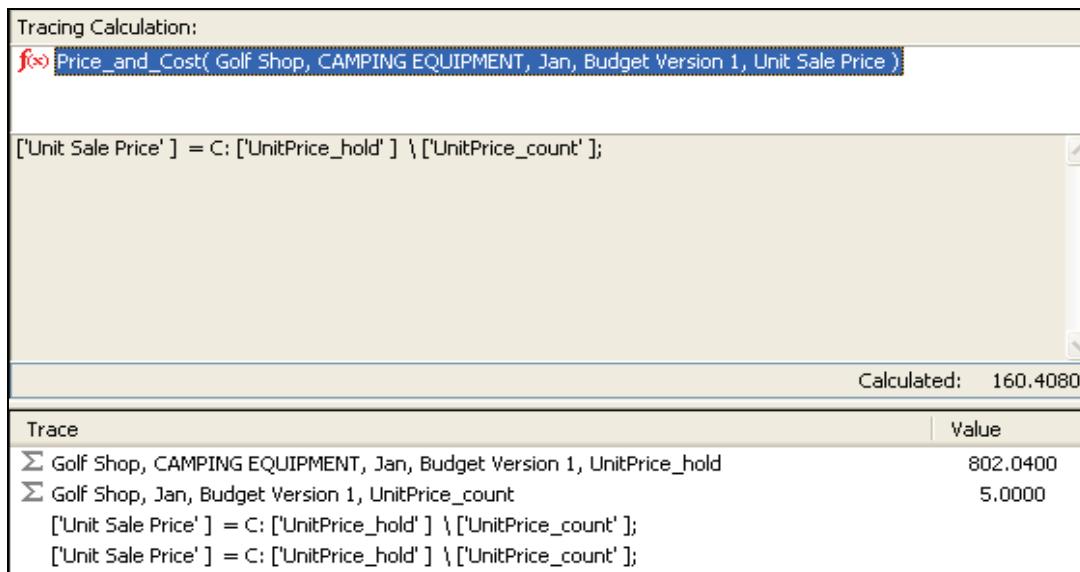
### Purpose:

You want to verify your calculations are consolidating properly. In addition you want to ensure that your feeders are working as expected. You will use the Rules Tracer to review your calculations and the associated feeder statements.

Server: localhost  
 User/Password: admin/<none>  
 Package: greatoutdoors  
 TM1 Architect: Server Explorer

### Task 1. Trace your calculations.

1. Open the AverageRule view of the Price\_and\_Cost cube.
2. Right-click the CAMPING EQUIPMENT for Unit Sale Price cell and then click Trace Calculation.



The top pane shows the calculation (consolidation) and the bottom pane shows the values that make up the calculation above.

You can double click a value in the bottom pane to trace it.

3. Double click the first line in the bottom pane (**Golf Shop, CAMPING EQUIPMENT, Jan, Budget Version 1, UnitPrice\_hold**).

**Tracing Calculation:**

- ↳ Price\_and\_Cost( Golf Shop, CAMPING EQUIPMENT, Jan, Budget Version 1, Unit Sale Price )
- Σ Price\_and\_Cost( Golf Shop, CAMPING EQUIPMENT, Jan, Budget Version 1, UnitPrice\_hold )

**Trace**

Trace	Value
↳ Cooking Gear	40.3100
↳ Golf Shop, Lanterns, Budget Version 1, UnitPrice_hold	29.1200
↳ Golf Shop, Packs, Budget Version 1, UnitPrice_hold	175.3700
↳ Golf Shop, Sleeping Bags, Budget Version 1, UnitPrice_hold	103.8300

Consolidated: 802.0400

Under Tracing Calculation, it displays the path that you are tracing. You can double-click on an expression here to return to a previously traced expression.

The bottom Trace will show the base level elements with a gray bullet, indicating there is nothing further to trace.

4. Click **File** and then click **Exit** to close the window.
5. Leave the **AverageRule** view open for the next task.

---

TM1 may not always follow the same element path to the underlying values, instead you should verify the values and the elements make sense in your own session.

## Task 2. Use the Rules Tracer to trace a feeder.

1. Right-click **Cooking Gear/Unit Sale Price** (40.31) and then click **Trace Feeders**.

The results appear as follows:

The screenshot shows the 'Tracing Feeders' window with the following details:

- Tracing Feeders:** A list containing a single item: `Price_and_Cost( Golf Shop, Cooking Gear, Jan, Budget Version 1, Unit Sale Price )`.
- Expression:** `['Unit Sale Price' ] => ['UnitPrice_hold'], ['UnitPrice_count' ];`
- Simple Value:** `40.3100`
- Trace:** A table with two rows, each showing a trace expression and its status ('fed').

Trace	Value
<code>f(x)Price_and_Cost( Golf Shop, Cooking Gear, Jan, Budget Version 1, UnitPric...</code>	fed
<code>f(x)Price_and_Cost( Golf Shop, Cooking Gear, Jan, Budget Version 1, UnitPric...</code>	fed

In the bottom pane it shows the exact cells that are being fed by the feeder displayed in the top pane.

You see 'fed' in the value column. It displays the feeder using this cell in the gray area and the expressions being fed at the bottom.

If you had selected a consolidated element such as, CAMPING EQUIPMENT, Trace Feeders would be disabled on the shortcut menu because Trace Feeders is only available at the base or n level elements.

2. Close the **Trace Feeders** window.

## Task 3. Check feeders.

1. Right-click **CAMPING EQUIPMENT/UnitPrice\_hold** (40.31) and then click **Check Feeders**.

Nothing is displayed in the Rules Tracer except for the calculation. This is the desired result. This is a consolidation that is fed properly.

2. Close the Rules Tracer.
3. Click **Price\_and\_Cost\_Measures**, click **All** and then click **OK**.
4. Click **Recalculate**.
5. Right-click **CAMPING EQUIPMENT/UnitCost\_hold** (0) and then click **Check Feeders**.

The results appear as follows:

Checking Feeders:	
$\Sigma$	Price_and_Cost( Golf Shop, CAMPING EQUIPMENT, Jan, Budget Version 1, UnitCost_hold )
Consolidated: 0.0000	
Trace	Value
↗(Golf Shop, Cooking Gear, Jan, Budget Version 1 ↗(Golf Shop, Lanterns, Jan, Budget Version 1 ↗(Golf Shop, Packs, Jan, Budget Version 1 ↗(Golf Shop, Sleeping Bags, Jan, Budget Version 1 ↗(Golf Shop, Tents, Jan, Budget Version 1	

Here we see the elements making up this consolidation and none of them are being used to feed it.

6. Close the **Rules Tracer** and the view without saving changes.

Note: You can only check the FEEDERS for a cell once per TM1 session. The action of checking the FEEDERS actually feeds the components of the consolidation. Any subsequent checking of the FEEDERS does not yield accurate results. If you want to check the FEEDERS for a cell more than once, you must recycle (shut down and restart) the TM1 server before every check.

**Results:**

**You have used the Rules Tracer to review your calculations. You have also verified the leaf level cells feeding other calculations and calculations are being fed by the correct base cells.**

## Feeding String Rules

- Rules applied to string cells may need to be fed when:
  - The rule pulls the string into another cube
  - The rule is populating a pick list
  - The target is using suppress zeros
- Use FEEDSTRINGS; in the rule file
  - insert it at the top of the file
  - create feeder statements in the feeders section

**Cognos**  
software



FEEDSTRINGS;

Once this declaration is in place, you can set up feeders for string cells in a cube view, and rely on the string to be available to other rules even if the view is zero-suppressed. Statements that define feeders for string cells should be created below the FEEDERS declaration in your rule.

To enable feeding of string cells, insert the FEEDSTRINGS declaration as the first line of your rule.

Feeder statements are placed under the FEEDERS declaration, after the rule statements.

Feeders from string cells fire whenever a value changes.

---

There is no demo for this because we have no stored string values. If you try to feed the string rules in the Rules cube (from the first demo) nothing will appear because none of the values are stored (in any cube)

## Summary

- At the end of this course, you should be able to:
  - discuss consolidations and sparsity
  - enhance consolidation performance using SKIPCHECK
  - use FEEDERS to optimize aggregations
  - check accuracy of rules and feeders



Cognos.  
software

## Workshop 1: Feed Rules in the GreatOutdoors Model

Now that you have computed the Average Unit Cost and Average Unit Sale Price for the Price\_and\_Cost cube, you will need to feed it to ensure the consolidations work properly and efficiently. In addition, you need to feed rules in the Sales\_Plan cube.

- Add feeders to the rules for Price\_and\_Cost.
- Verify the correct data in the Price\_and\_Cost cube.
- Add SKIPCHECK and FEEDERS to the Sales\_Plan rules.
- Verify the correct data in the Sales\_Plan cube.

For more detailed information outlined as tasks, see the Task Table section.

For the final results, see the Workshop Results section that follows the Task Table section.

## Workshop 1: Task Table

Task	Where to Work	Hints
1. Add FEEDERS to Price_and_Cost rules.	Price_andCost: Rules Editor	<ul style="list-style-type: none"> <li>Open the Rules Editor.</li> <li>Compose feeder statements for UnitCost_hold and UnitCost_count rules.</li> <li>'[Unit Cost'] =&gt; ['UnitCost_hold'], ['UnitCost_count'];</li> </ul>
2. Verify correct data is calculated in Price_and_Cost.	Price_andCost: Cube Viewer	<ul style="list-style-type: none"> <li>Open the Average_Rule view and recalculate.</li> </ul>
3. Add SKIPCHECK and FEEDERS to Sales_Plan rules.	Sales_Plan: Rules Editor	<ul style="list-style-type: none"> <li>Open the Rules Editor.</li> <li>Compose feeder statements for rules.</li> <li>'[Quantity'] =&gt; ['GROSS SALES REVENUE'], ['COST OF SALES'];</li> </ul>
4. Verify correct data is calculated in Sales_Plan.	Sales_Plan: Cube Viewer	<ul style="list-style-type: none"> <li>Ensure feeder statements for UnitCost_hold and UnitCost_count rules.</li> </ul>

If you need more information to complete a task, see the Step-by-Step instructions at the end of the Workshop.

## Workshop 1:Results

The result of the workshop is a report that appears as follows:

Task 1. Add feeders to Price\_and\_Cost rules.

```
11 FEEDERS;
12 ['Unit Sale Price'] => ['UnitPrice_hold'], ['UnitPrice_count'];
13 ['Unit Cost'] => ['UnitCost_hold'], ['UnitCost_count'];
```

Task 2. Verify correct data is calculated in Price\_and\_Cost.

Price_and_Cost_Measures									
Products	Unit Sale Price	Unit Cost	Margin	Margin %	UnitPrice_hold	UnitPrice_count	UnitCost_hold	UnitCost_count	
-- TOTAL PRODUCT	\$166.54	\$109.79	\$0.00	0.00	3497.35	21	2305.6	21	
-- CAMPING EQL	\$160.41	\$128.11	\$0.00	0.00	802.04	5	640.57	5	
Cooking Gear	\$40.31	\$29.87	\$0.00	0.00	40.31	1	29.87	1	
Lanterns	\$29.12	\$21.14	\$0.00	0.00	29.12	1	21.14	1	
Packs	\$175.37	\$126.51	\$0.00	0.00	175.37	1	126.51	1	
Sleeping Bag	\$103.83	\$78.01	\$0.00	0.00	103.83	1	78.01	1	
Tents	\$453.41	\$385.04	\$0.00	0.00	453.41	1	385.04	1	

Task 3. Add SKIPCHECK and FEEDERS to Sales\_Plan rules.

```
1 SKIPCHECK;
2 ['Unit Sale Price'] = N: DB('Price_and_Cost', !Channels, !Products, !Months, !Versions, 'Unit Sale Price') ;
3                                     C: ['GROSS SALES REVENUE'] / ['Quantity'] ;
4 ['Unit Cost'] = N: DB('Price_and_Cost', !Channels, !Products, !Months, !Versions, 'Unit Cost') ;
5                                     C: ['COST OF SALES'] / ['Quantity'] ;
6 ['GROSS SALES REVENUE'] = N: ['Quantity'] * ['Unit Sale Price'] ;
7 ['COST OF SALES'] = N: ['Quantity'] * ['Unit Cost'] ;
8
9 FEEDERS;
10 ['Quantity'] => ['GROSS SALES REVENUE'], ['COST OF SALES'];
```

## Task 4. Verify correct data is calculated in Sales\_Plan.

GO Americas		Department Store		
Cooking Gear		Budget Version 1		
		Months		
Sales_Plan_Measu	-- Total Year	-- Q1	Jan	Feb
Quantity	455,384.00	102,767.00	47,742.00	11,935.00
Unit Sale Price	\$42.33	\$42.33	\$42.33	\$42.33
Unit Cost	\$31.36	\$31.36	\$31.36	\$31.36
-- GROSS MARGIN	\$1,187,933.39	\$268,087.20	\$124,549.09	\$31,131.18
-- NET SALES RE	\$15,470,369.47	\$3,491,220.01	\$1,621,905.31	\$405,454.55
GROSS SALE	\$19,274,355.49	\$4,349,664.66	\$2,020,704.02	\$505,154.84
-- TOTAL DIS	\$1,869,190.82	\$421,820.21	\$195,960.34	\$48,990.80
Returns a	\$841,215.30	\$189,836.73	\$88,190.61	\$22,047.61
Cash Disc	\$622,292.18	\$140,432.69	\$65,239.03	\$16,310.20
Volume D	\$405,683.34	\$91,550.79	\$42,530.70	\$10,632.99
Freight	\$1,934,795.20	\$436,624.44	\$202,838.37	\$50,709.49
COST OF SALES	\$14,282,436.08	\$3,223,132.80	\$1,497,356.22	\$374,323.37
GROSS MARGIN %	0	0	0	0
Seasons				

Checking Feeders:

$\Sigma$  Sales\_Plan( GO Americas, Department Store, Cooking Gear, Q1, Budget Version 1, GROSS SALE )

Consolidated: 4349664.6585

Trace	Value

## Workshop 1: Step-by-Step Instructions

TM1 Server: **greatoutdoors**  
 User/Password: **admin/<none>**  
 TM1 Architect: **Server Explorer**

### Task 1. Add feeders to Price\_and\_Cost rules.

1. Right-click **Price\_and\_Cost** cube and then click **Edit Rule**.
2. On the last line (13) type the following:  
`['Unit Cost'] => ['UnitCost_hold'], ['UnitCost_count'] ;`
3. Click **Check Syntax** and correct any errors.
4. Click **Save** and then close the Rules Editor.

### Task 2. Verify correct data is calculated in Price\_and\_Cost.

1. Double-click the **Price\_and\_Cost** cube.
2. From the **Select View** list, click **AverageRule**.
3. Click **Price\_and\_Cost\_Measures**, click **All** and then click **OK**.
4. Click **Recalculate**.
5. Verify the **CAMPING EQUIPMENT** for **UnitCost\_hold** and **UnitCost\_count** are now displaying values instead of zeros.
6. Close the **Cube Viewer** without saving changes.

### Task 3. Add SKIPCHECK and FEEDERS to Sales\_Plan rules.

1. Right-click **Sales\_Plan** cube and then click **Edit Rule**.
2. At the beginning of the first line, press **Enter**.
3. On line 1 type:  
**SKIPCHECK;**

4. At the end of the last line, press **Enter**, and then type:  
**FEEDERS;**
5. Press **Enter**, and then type:  
**['Quantity' ] => ['GROSS SALES REVENUE' ], ['COST OF SALES' ];**
6. Click **Check Syntax** and correct any errors.
7. Click **Save** and close the Rules Editor.

#### **Task 4. Verify correct data is calculated in Sales\_Plan.**

1. Double-click the **Sales\_Plan** cube.
2. From the **Select View** list, click **Revenue**.
3. Click **Months**.
4. Click the **All** button and then click **OK**.
5. Click **Recalculate**.
6. Verify the **Q1/GROSS SALES REVENUE** displays values instead of zeros.
7. Right-click **Q1/GROSS SALES REVENUE** and choose **Check Feeders** and you should not see any statements below indicating it is not fed.
8. Close the **Rules Tracer**.
9. Close the **Cube Viewer** without saving changes.
10. Close TM1 Architect: Server Explorer and then close the greatoutdoors server (saving changes).



**Information Management**



## **Transfer Data into Your Model Using Scripts**

IBM Cognos TM1 9.5



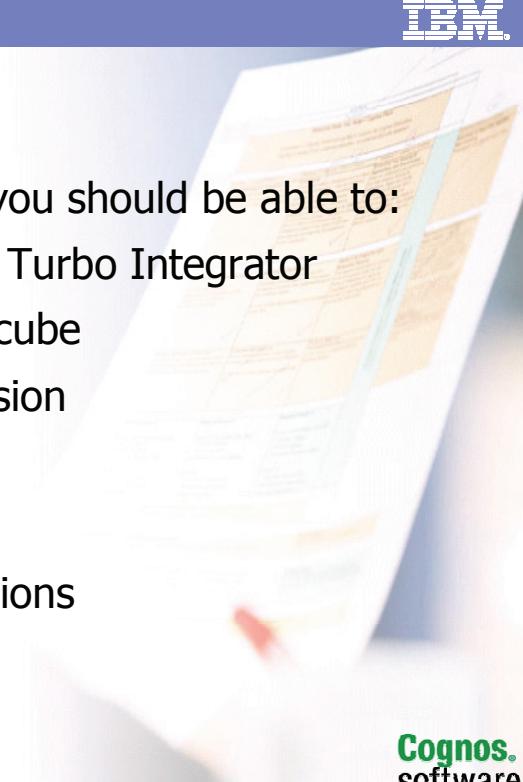
**Cognos.**  
software

© 2009 IBM Corporation



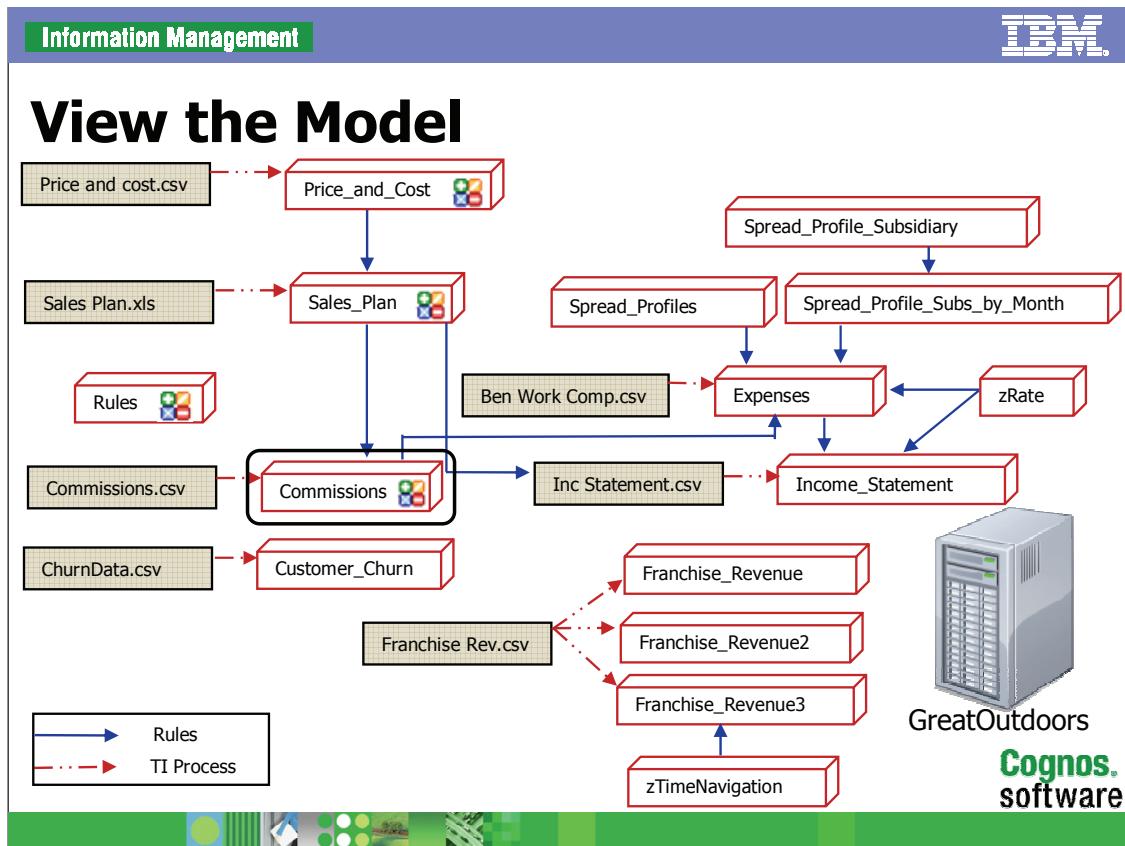
# Objectives

- At the end of this module, you should be able to:
  - review Advanced tabs in Turbo Integrator
  - load custom data into a cube
  - add a subset to a dimension
  - use logic in scripts
  - export data to ASCII
  - move data between versions
  - construct chores



Cognos  
software





In this module, you will create one new cube:

- Commissions

You will create a number of Turbo Integrator scripts.

## Link Cubes with Different Dimensions

- Link data between cubes using the DB() function.
- The DB() function uses references to:
  - the cube where the data resides.
  - an element from each of the dimensions in the source cube.
- Cubes may or may not have the same dimensions or dimensionality.
  - find an element to represent the missing dimension

 Cognos  
software

If the target cube contains a dimension not found in the source, there will be a reference for it included in the area definition of the rule.

- In the following example, there is a Day dimension in the target cube that is not found in the source. The 'Jan 1' identifies the missing element.
- ['Jan 1', 'Cost']=DB('Price\_and\_Cost', !Products, !Date, 'Cost');

If the source cube contains a dimension not found in the target cube, there will be a reference for it included in the DB() function dimension variables. The apex of the hierarchy is often used (i.e. Total Products, Total Channels, etc.)

- In the following example, there is a Market dimension in the source cube that does not reside in the target cube.
- ['Cost']=DB('Price\_and\_Cost', !Products, !Date, 'Total Market', 'Cost')

## Review Turbo Integrator

- Turbo Integrator is an Extraction, Transformation, and Load (ETL) tool.
- Turbo Integrator components you have used:
  - description
  - variables
  - maps

Cognos  
software



Turbo Integrator (TI) can be used to define dimensions, create cubes, and populate cubes from ASCII files, ODBC sources, TM1 cube views, and TM1 dimension subsets. TI may also be used to write data to ODBC sources.

TI allows you to create processes that automate data importation, metadata management, and other tasks.

TM1 TI includes a complete set of functions that you can use to enhance process capabilities. You can use these functions to create scripts that export data to ASCII files and ODBC sources, or that use conditional expressions to control processing. In addition to these TI functions, you can also incorporate most TM1 Rules functions.

**INTERACTION - Whiteboard:** ask participants to list some of the other activities they have automated using TI. Have each person pick a different color text and write it on the whiteboard.

## Additional TI Process Components

- These components may be customized to perform model updates and maintenance:
  - prolog
  - metadata
  - data
  - epilog
  - parameters



Cognos  
software

A prolog procedure consists of a series of actions to be executed before the data source is processed.

A metadata procedure consists of a series of actions that update or create cubes, dimensions, and other metadata structures.

A data procedure consists of a series of actions to be executed for each record in the data source.

An epilog procedure is to be executed after the data source is processed.

A set of parameters can be used to generalize a process so it can be used in multiple situations.

Every TI process is made up of these four procedures. They are executed in succession. The Advanced tabs show the statements generated using the TI interface and you may customize your own procedures by using TI functions. Any customization must be added outside of the reserved area.

## Benefits of Custom Scripts

- Easier to maintain
- Faster to rebuild or update
- Consistency when adding data
- Use in chores

Cognos  
software

TI is very useful as an ETL tool just by using the basic steps to create variables, map data and run the resulting processes. However, your TM1 model may become much easier to update and maintain once you become familiar with customizing and generating your own TI process scripts.

A few of the things you can do include:

- Schedule processes to run periodically
- Create subsets automatically
- Extract data from a view and move it into another one
- Archive a cube

---

The following demo is to be used as a TI refresher.

### INTERACTION - Survey:

How many have customized scripts? Follow with discussion via microphone and ask for some examples they may have seen or can think of.

## Demo 1: Load Data from Multiple Columns to a Single Column

### Purpose:

You want to create and populate the Commissions cube.

First you will create the Commission Measures dimension and the Commissions cube. Next, you will write rules that define how data will be populated, indicating that Gross Sales Revenue comes from another cube, and that it is used in other calculations in the same measure. Finally, you will populate the cube using a Turbo Integrator process.

### Task 1. Create the Commission Measures dimension.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

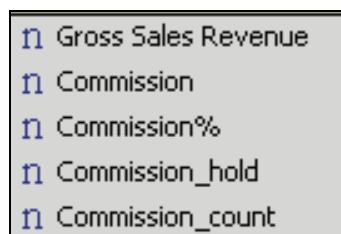
Password: **<blank>**

1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. Right-click **Dimensions**, and then click **Create New Dimension**.
4. From the **Edit** menu, click **Insert Element**.

5. Add the following elements, clicking **Add** after each one:
  - **Gross Sales Revenue**
  - **Commission**
  - **Commission%**
  - **Commission\_hold**
  - **Commission\_count**

6. Click **OK**.

The result appears as shown below:



7. Save the dimension as **Commission\_Measures**, and then close the **Dimension Editor**.

## **Task 2. Create the Commissions cube.**

1. Right-click **Cubes**, and then click **Create new cube**.
2. In the **Cube Name** box, type **Commissions**.
3. In the **Available Dimensions** list, double-click the following **Dimensions** to move them to the **Dimensions in new Cube** list:
  - **Subsidiaries**
  - **Months**
  - **Versions**
  - **Commission\_Measures**
4. Click **Create Cube**.

The Commissions cube appears in the list of cubes.

## Task 3. Write rules.

When you use the Rules Editor, TM1 automatically turns off its fast consolidation algorithm, which skips empty cells. To ensure that performance is not slowed down, the first line you add to the Rules Editor is "SKIPCHECK", which tells TM1 to skip all empty cells.

1. Under **Cubes**, right-click the **Commissions** cube, and click **Create Rule**.
2. On line 1, type **SKIPCHECK**:

Next you want the data from the Sales\_Plan cube to be used in the Commissions cube.

3. On line 2, type **# Pull data from the Sales\_Plan cube**.  
Lines with comments start with '#'.  
4. On line 3, click **Brackets**, double-click **Commission\_Measures**.
5. Click **Gross Sales Revenue** and then click **OK**.
6. Type **=**, click **Insert Cube Reference**  and then in the **Cube** list click **Sales\_Plan**.
7. Click the **Subset** icon next to Channels, click **ALL CHANNELS**, and then click **OK**.

8. Repeat step 7, with the following:

- Products, click **TOTAL PRODUCTS**
- Sales\_Plan\_Measures, click **GROSS SALES REVENUE**

The result appears as shown below:

Cube:		
Sales_Plan		
<input type="checkbox"/> Include Control Cubes		
Dimensions: 6		<b>Reset All</b>
	Name	Reference
0	Subsidiaries	!Subsidiaries
1	Channels	ALL CHANNELS
2	Products	TOTAL PRODUCTS
3	Months	!Months
4	Versions	!Versions
► 5	Sales_Plan_Measures	GROSS SALES REVENUE

The Commissions cube does not contain all the dimensions that the Sales\_Plan cube does. This is why you must select specific elements from those dimensions (Channels, Products, and Sales\_Plan\_Measures). You do not want to exclude any data from those dimensions so you need to choose the highest levels or the totals in the dimensions.

9. Click **OK**, and then after the bracket type ; at the end of the line.  
 10. After **!Subsidiaries**, press **Enter**, and then press **Tab**.

This makes the expression more readable.

```

1 SKIPCHECK;
2 # Pull data from the Sales_Plan cube
3 ['Gross Sales Revenue'] = DB('Sales_Plan', !Subsidiaries,
4 'ALL CHANNELS', 'TOTAL PRODUCTS', !Months, !Versions, 'GROSS SALES REVENUE');

```

## Task 4. Calculate average Commission% at the consolidated level.

1. On line 6, type the following comment:

# Calculate average Commission% at the consolidated level

2. On line 7, click **Brackets** and then double-click **Commission\_Measures**.
3. Click **Commission\_hold** and then click **OK**.
4. Click after the bracket, click **Insert Qualifier or Operator** icon and then click **=N:**.
5. Click **Brackets**, double-click **Commission\_Measures**, and then click **Commission%**.
6. Click **OK**, and then after the bracket type **;**.
7. On lines 8 and 9, type the following:

**'[Commission\_count']=N:IF(['Commission%']<>0,1,0);**

**'[Commission%']=C:(['Commission\_hold'])\['Commission\_count']);**

The conditional statement on line 8 indicates that if the Commission% is not zero, the Commission\_count = 1, otherwise Commission\_count=0.

The "\\" on line 9 indicates that Commision% is equal to Commission\_hold divided by Commision\_count, when Commission\_count is not equal to zero. When Commission\_count is equal to zero, then Commission% is also equal to zero.

```

1 SKIPCHECK;
2 # Pull data from the Sales_Plan cube
3 ['Gross Sales Revenue'] = DB('Sales_Plan', /Subsidiaries,
4      'ALL CHANNELS', 'TOTAL PRODUCTS', /Months, /Versions, 'GROSS SALES REVENUE');
5
6 # Calculate average Commission% at the consolidated level
7 ['Commission_hold'] = N: ['Commission%'];
8 ['Commission_count']=N:IF(['Commission%']<>0,1,0);
9 ['Commission%'] = C:[['Commission_hold']\['Commission_count']];

```

## Task 5. Calculate commission and add Feeders.

1. On lines 11 and 12, type the following:

# Calculate commission

**[Commission'] = N: ([Gross Sales Revenue]\*[Commission%])/100;**

2. On lines 14 and 15, enter the following:

**FEEDERS;**

**[Commission%]=>[Commission],[Commission\_hold],**  
**[Commission\_count];**

The result appears as shown below:

```

1  SKIPCHECK;
2  # Pull data from the Sales_Plan cube
3  ['Gross Sales Revenue'] = DB('Sales_Plan', !Subsidiaries,
4      'ALL CHANNELS', 'TOTAL PRODUCTS', !Months, !Versions, 'GROSS SALES REVENUE');
5
6  # Calculate average Commission% at the consolidated level
7  ['Commission_hold'] = N: ['Commission%'];
8  ['Commission_count'] = N: IF(['Commission%']<>0,1,0);
9  ['Commission%'] = C: ([['Commission_hold']\['Commission_count']]);
10
11 # Calculate commission
12 ['Commission'] = N: ([Gross Sales Revenue]*[Commission%])/100;
13
14 FEEDERS;
15 ['Commission%']=> ['Commission'], ['Commission_hold'], ['Commission_count'];

```

3. Save and close the **Rules Editor**.

## Task 6. Load data.

1. Right-click **Processes**, and then click **Create New Process**.

- Datasource Type: **Text**
- Data Source Name:  
**C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Commissions.csv**
- Number of title records: **1**

## 2. Click Preview.

The result appears as follows:

The screenshot shows the 'Data Source' configuration window with the 'Preview' button highlighted. The configuration includes settings for Data Source Name (C:\Edcognos\TM9402\Data Files\Commissions.csv), Delimiter Type (Delimited), and various separator and quote character options. The preview pane displays a table of data with columns: Subsidiary, Versions, Commission, Jan, Feb, Mar. The data rows show values for GO Americas, GO Central Europe, GO Northern Europe, and GO Southern Europe across two budget versions and three months.

	Subsidiary	Versions	Commission	Jan	Feb	Mar
1	GO Americas	Budget version 1	Commission%	1.305525077	1.228448156	1.326075312
2	GO Americas	Budget version 2	Commission%	1.436077585	1.351292972	1.458682844
3	GO Central Europe	Budget version 1	Commission%	1.649154351	1.523350769	1.618627465
4	GO Central Europe	Budget version 2	Commission%	1.814069787	1.675685846	1.780490212
5	GO Northern Europe	Budget version 1	Commission%	1.568747398	1.562637251	1.6348923
6	GO Northern Europe	Budget version 2	Commission%	1.725622138	1.718900976	1.79838153
7	GO Southern Europe	Budget version 1	Commission%	1.453862843	1.391657939	1.457711455
8	GO Southern Europe	Budget version 2	Commission%	1.599249127	1.530823733	1.603482601

3. Click the **Variables** tab and set the following:

- For **Subsidiary**, **Versions** and **Commission**, in the **Contents** column click **Element**.
- From **Jan** to **Dec**, in the **Variable Type** column, click **Numeric**, and then in the **Contents** column click **Data**.

The result appears as follows:

	Variable Name	Variable Type	Sample Value	Contents
1	Subsidiary	String	▼ GO Americas	Element
2	Versions	String	▼ Budget version 1	Element
3	Commission	String	▼ Commission%	Element
4	Jan	Numeric	▼ 1.305525077	Data
5	Feb	Numeric	▼ 1.228448156	Data
6	Mar	Numeric	▼ 1.326075312	Data
7	Apr	Numeric	▼ 1.299245057	Data
8	May	Numeric	▼ 1.270079146	Data
9	Jun	Numeric	▼ 1.220289262	Data
10	Jul	Numeric	▼ 1.207225336	Data
11	Aug	Numeric	▼ 1.202830774	Data
12	Sep	Numeric	▼ 1.206813743	Data
13	Oct	Numeric	▼ 1.20733458	Data
14	Nov	Numeric	▼ 1.210185072	Data
15	Dec	Numeric	▼ 1.205840612	Data

4. Click the **Maps** tab.

5. Below **Cube Action**, click **Update Cube**, and then in the **CubeName** box, click **Commissions**.

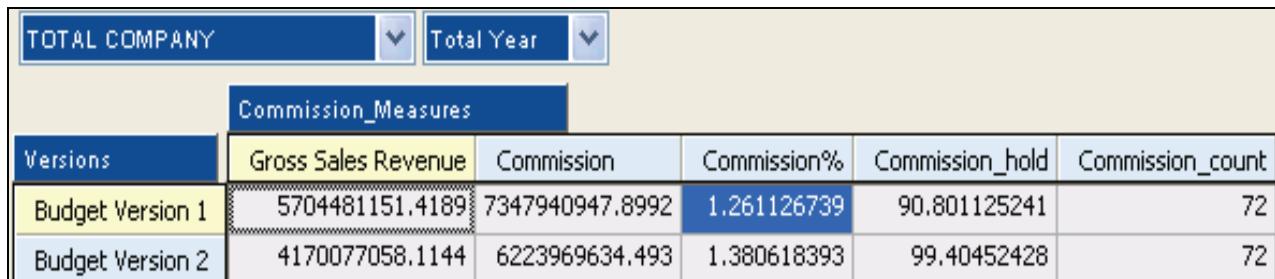
6. Click the **Dimensions** tab.

7. Beside each Element Variable, click the appropriate Dimension:

Element Variable	Dimension
Subsidiary	Subsidiaries
Versions	Versions
Commission	Commission Measures
(Data Variables)	Months

8. Click the **Data** tab.
9. In the Element column, type **Jan** to **Dec**, matching the contents of the **Data Variable** column.
10. Click the **Advanced** tab, and click through the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.
- You need to scroll through the different tabs to regenerate the scripts.
11. Save the process as **LoadCommissionPctCSV**, and then click **Run**.
12. Click **OK** and then close TurboIntegrator.
13. Open the **Commissions** cube, and then click **Recalculate**.

The result appears as follows:



The screenshot shows a 'TOTAL COMPANY' cube viewer with 'Total Year' selected. The table displays data for 'Commission\_Measures' across two versions: 'Budget Version 1' and 'Budget Version 2'. The columns are: Versions, Gross Sales Revenue, Commission, Commission%, Commission\_hold, and Commission\_count. The data is as follows:

Commission_Measures		Gross Sales Revenue	Commission	Commission%	Commission_hold	Commission_count
Versions						
Budget Version 1		5704481151.4189	7347940947.8992	1.261126739	90.801125241	72
Budget Version 2		4170077058.1144	6223969634.493	1.380618393	99.40452428	72

14. Close the **Cube Viewer** without saving the view.
15. In Server Explorer, right click **Commission\_Measures**, and then click **Edit Element Attributes**.

16. Click and drag to select **Gross Sales Revenue** and **Commission** and then click **Format**.
17. Click **Currency** and then click **OK** twice.
18. Double-click the **Commissions** cube, and then click **Recalculate**.

The results appear as follows:

TOTAL COMPANY		Total Year				
Commission_Measures						
Versions	Gross Sales Revenue	Commission	Commission%	Commission_hold	Commission_count	
Budget Version 1	\$5,704,481,151.42	\$73,479,409.48	1.261126739	90.801125241		72
Budget Version 2	\$4,170,077,058.11	\$62,239,696.34	1.380618393	99.40452428		72

19. Close the cube viewer

## Results:

You created and populated the Commissions cube. First you created the Commission Measures dimension, then you created the Commissions cube, and then you wrote rules that defined how data is populated. Finally you populated the cube using a Turbo Integrator process.

## Update a Dimension

- TI can create or update a dimension including the:
  - hierarchy
  - elements
  - attributes
  - sort order
- TI Functions must be used to create, update or destroy Subsets for a dimension.

The tabs that guide you through building a TI process are very useful but they may only perform a set number of tasks. You can perform additional tasks using TI and Rules functions directly in the TI process scripts.

## Demo 2: Create a Dimension Including a Subset with Turbo Integrator

### Purpose:

You want to create Product Type subsets that are available to users.  
You will use functions in Turbo Integrator to do this.

### Task 1. Create a new process in Turbo Integrator.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Right-click **Processes**, and then click **Create New Process**.
  - Data Source Type: **Text**
  - Data Source Name:  
**C:\Edcognos\P6502\GreatOutdoors\SourceFiles\Products.csv**
2. Click **Preview**.

The result appears as follows:

V1	V2	V3
Cooking Gear	CAMPING EQUIPMENT	TOTAL PRODUCTS
Lanterns	CAMPING EQUIPMENT	TOTAL PRODUCTS
Packs	CAMPING EQUIPMENT	TOTAL PRODUCTS
Sleeping Bags	CAMPING EQUIPMENT	TOTAL PRODUCTS
Tents	CAMPING EQUIPMENT	TOTAL PRODUCTS
Golf Accessories	GOLF EQUIPMENT	TOTAL PRODUCTS
Irons	GOLF EQUIPMENT	TOTAL PRODUCTS
Putters	GOLF EQUIPMENT	TOTAL PRODUCTS
Woods	GOLF EQUIPMENT	TOTAL PRODUCTS
Climbing Accessorie	MOUNTAINEERING	TOTAL PRODUCTS

3. Click the **Variables** tab and set the following:

- Rename **V1** as **vProducts**
- Rename **V2** as **vProductType**
- Rename **V3** as **vTotalProducts**
- Set the **Contents** for all three variables as **Other**

The result appears as follows:

Data Source Variables Maps Advanced Schedule					
	Variable Name	Variable Type	Sample Value	Contents	Formula
1	vProducts	String	Cooking Gear	Other	
2	vProductType	String	CAMPING EQUIPMENT	Other	
3	vTotalProducts	String	TOTAL PRODUCTS	Other	

Marking the Contents as Other allows the variables to still be available to TI processes (unlike Ignore) but does not force you to map them under the Maps tab.

This allows you to use them to write custom scripts on the advanced tabs.

4. Click the **Advanced** tab, click **Metadata**, and then after the generated statements add the following:

**# Create Subsets for each Product Type**

**IF**

**(SubsetExists('Products',vProductType)=0);**

**SubsetCreate('Products',vProductType);**

**ENDIF;**

Note: You can copy the syntax of the SubsetCreate function from the online help, to minimize typos. You will need to right-click, and then click Copy.

Statements with # before them are comments.

- SYNTAX: SubsetExists(DimName, SubsetName);
    - This checks to see if the specified Subset (SubsetName) exists for the specified dimension (DimName). You are not over-writing an existing subset.
  - SYNTAX: SubsetCreate(DimName, SubName);
    - This creates the specified Subset (SubName) for the specified dimension (DimName).
  - IF, ENDIF, (ELSEIF)
    - The If statement allows a process to execute a statement or series of statements when a given expression is true. You can use arithmetic operators, logical operators, and comparison operators to construct an expression.
    - The Turbo Integrator If statement differs from the Rules IF function in that the Turbo Integrator statement can accept multiple ElseIf statements to evaluate multiple expressions, while the Rules IF function can evaluate only one expression.
    - You can nest up to 20 If statements in a TurboIntegrator process. If you exceed 20 nested If statements, you will receive an error when attempting to save the process.
5. Press **Enter** after the ENDIF statement, and then type the following:
- ```
# Insert elements from vProducts under the Subset name in
vProductType
```

6. Press Enter and then type the following:

```
SubsetElementInsert('Products', vProductType, vProducts, 0);
```

SYNTAX: SubsetElementInsert(DimName, SubName, ElName, Position);

Products (DimName) is the dimension name, vProductType (SubName) is the variable containing the subset names you created on the Prolog tab, vProducts (ElName) is the variable containing the element names corresponding to the vProductType, and 0 (Position) indicates the first position in the index.

The result appears as follows:

|                                                                                                                                                                                                                                                                                                                                                               |           |          |          |          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------|----------|----------|
| Data Source                                                                                                                                                                                                                                                                                                                                                   | Variables | Maps     | Advanced | Schedule |
| Parameters                                                                                                                                                                                                                                                                                                                                                    | Prolog    | Metadata | Data     | Epilog   |
| <pre>#****Begin: Generated Statements**** #****End: Generated Statements****  #Create Subsets for each Product Type  IF   (SubsetExists('Products',vProductType)=0);   SubsetCreate('Products',vProductType); ENDIF;  #Insert elements from vProducts under the Subset name in vProductType SubsetElementInsert('Products', vProductType, vProducts,0);</pre> |           |          |          |          |

7. Save the process as **CreateProductType\_subsets**, run the process and then close the Turbo Integrator.
8. In the Subset Editor, open **Products**.

9. Click the **Select Subset** box, to view the subsets.

The result appears as follows:



10. Close the Subset Editor.

**Purpose:**

You created a Product Type subset which are available to users. As you add new elements, you may run this process and the new elements will be added to the appropriate subset. You used functions in Turbo Integrator to do this.

## Dealing with Data

- Data may come from a variety of sources:
  - general ledger accounts
  - transactional databases
  - text files
  - spreadsheets
- Sometimes data must be transformed before it can fit into the model.

TI functions may be used to alter the data as it comes into the model. You have used some functions to change data types (NUMBERTOSTRING() and STRINGT tonumber()) or to extract just the necessary components from the data (MONTHS()). Some functions require editing the process scripts directly.

Customizing the hierarchy to accommodate uneven levels can only be resolved by TI if the data is altered. Another option is to customize the scripts to adjust the parentage to support the uneven levels.

## Demo 3: Create a Dimension with Uneven Hierarchy

### Purpose:

You want to create a dimension that organizes data by subsidiary and by region, so all the subsidiaries in a region are aggregated at the region level.

In some cases, there is only one subsidiary in a region, and no region is specified. This is also known as having an uneven hierarchy because the 'parent' has no child objects.

To resolve any process errors, you will create a Turbo Integrator process that terminates the process at different levels for different hierarchies.

### Task 1. Create an uneven Subsidiary process.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

User Name: **admin**

Password: **<blank>**

1. Right-click **Processes**, and then click **Create New Process**.
  - Data Source Type: **Text**
  - Data Source Name:  
**C:\Edcognos\P6502\GreatOutdoors\SourceFiles\UnevenSubsidiaries.csv**

2. Click **Preview**.

The result appears as follows:

|   | V1                  | V2              | V3            |
|---|---------------------|-----------------|---------------|
| 1 | GO Americas         | GO Americas     | TOTAL COMPANY |
| 2 | GO Central Europe   | GO EUROPE GMBH  | TOTAL COMPANY |
| 3 | GO Northern Europe  | GO EUROPE GMBH  | TOTAL COMPANY |
| 4 | GO Southern Europe  | GO EUROPE GMBH  | TOTAL COMPANY |
| 5 | GO Asia Pacific     | GO Asia Pacific | TOTAL COMPANY |
| 6 | GO Accessories GmbH | GO EUROPE GMBH  | TOTAL COMPANY |

Note: V1 and V2 have the same value for Go Americas and GO Asia Pacific.

TM1 requires that elements (either leaf level or string level) be unique.

Otherwise Turbo Integrator creates dimensions with parent (consolidations) that have no child objects.

There are two ways to resolve this:

- Alter the names of the elements which have the same name on both levels.  
You can see this in the subsidiary.csv file.
- Create a Turbo Integrator process to create the uneven hierarchy.

You will first create a Turbo Integrator process that does not address the duplicates, and then later you will correct the process.

3. Click the **Variables** tab and set the following:

- Rename **V1** as **vSubsidiary**
- Rename **V2** as **vRegion**
- Rename **V3** as **vTotalCompany**

4. Set the **Contents** as shown below:

| Variables |               |               |               |               |  |
|-----------|---------------|---------------|---------------|---------------|--|
|           | Variable Name | Variable Type | Sample Value  | Contents      |  |
| 1         | vSubsidiary   | String        | GO Americas   | Element       |  |
| 2         | vRegion       | String        | GO Americas   | Consolidation |  |
| 3         | vTotalCompany | String        | TOTAL COMPANY | Consolidation |  |

## Task 2. Map the variables.

1. Click **Maps**, click the **Dimensions** tab, and then in the **Dimension** column, beside **vSubsidiary**, type **UnevenSubsidiary**.  
You want to set the Element Order to "by Hierarchy".
2. In the **Element Order** column, click **By Input**, click **Automatic**, click **Hierarchy**, and then click **OK**.

| Element Variable | Sample Value | Dimension        | Order In Cube | Action   | Element Type | Element Order  |
|------------------|--------------|------------------|---------------|----------|--------------|----------------|
| vSubsidiary      | GO Americas  | UnevenSubsidiary | ▼             | ▼ Create | Numeric      | ▼ By Hierarchy |

3. Click the **Consolidations** tab, beside **vRegion**, under **Child Variable**, click **vSubsidiary** and then beside **vTotalCompany** click **vRegion**.  
Next you want to set each consolidation variable to order its components by name.
4. Beside **vRegion**, click **By Input**, click **Automatic**, and then repeat for **vTotalCompany**.

The result appears as follows:

| Cons. Variable | Dimension          | Child Variable | Weight   | Sample Value  | Component Order |
|----------------|--------------------|----------------|----------|---------------|-----------------|
| vRegion        | UnevenSubsidiary ▶ | vSubsidiary ▶  | 1.000000 | GO Americas   | By Name         |
| vTotalCompany  | UnevenSubsidiary ▶ | vRegion ▶      | 1.000000 | TOTAL COMPANY | By Name         |

5. Click the **Advanced** tab, and then click through the **Prolog**, **Metadata**, **Data**, and **Epilog** tabs.
6. Save the process as **CreateUnevenSubsidiary**, and then run the process.  
You will get the message that the process completed with errors.
7. Click **Yes**, to view the log file.

8. Double-click the first row.

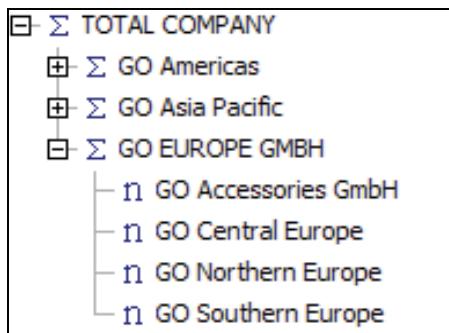
The circular reference indicates that the items at the leaf level are not unique.

9. Close the log files and leave TI open.

The log file may be found in the TM1 Server Data Directory (C:\Edcognos\P6502\GreatOutdoors\LogFiles on the class image). The file name is TM1ProcessError\_xx\_xx.log. The first xx represents the date/timestamp and the 2<sup>nd</sup> xx represents the TI Process name. This file can be opened in any text editor.

10. Open the **UnevenSubsidiary** dimension, click **All**  and then click **Hierarchy Sort** .

The result appears as follows:



Go Americas and GO Asia Pacific should be identified as leaf level elements (**n** level) since they have no children.

Instead GO Americas and GO Asia Pacific are defined as Consolidations with no children. You cannot load data to these consolidated elements.

Next you will edit or process to handle the duplicate data and terminate the hierarchies at different levels for the different elements.

11. Save as **Subset1** and then close the **Subset Editor**.

### Task 3. Modify the process to terminate the hierarchies at different levels for different elements.

1. Double-click the **CreateUnevenSubsidiary** process to open it.
2. Click the **Maps** tab.
3. On the **Maps** tab, click the **Dimensions** tab, and change the Action to **Recreate**.

Because the dimension has already been created, you cannot make any other changes to the process until you change the action.

4. On the **Advanced** tab, click the **Prolog** tab.
5. Copy everything between **Begin:** and **End:**, and then paste it outside the generated script space.

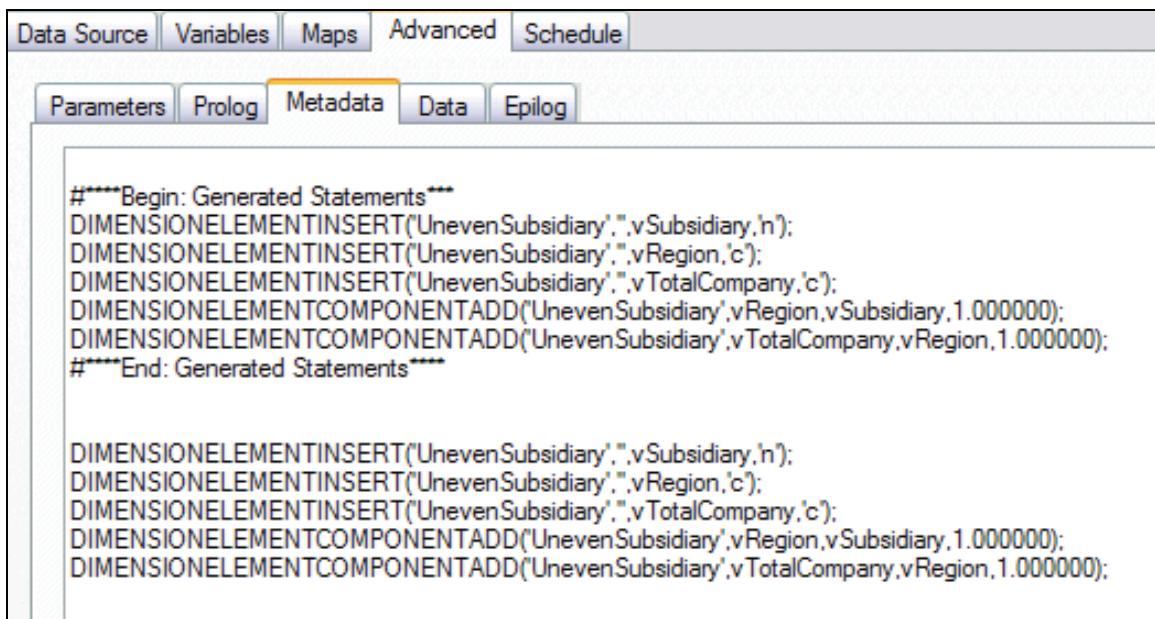
Note: You will need to right-click to copy and paste in TM1.

The result appears as follows:

|                                                                                                                                                                                                                                                                                                                                                              |           |      |          |          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------|----------|----------|
| Data Source                                                                                                                                                                                                                                                                                                                                                  | Variables | Maps | Advanced | Schedule |
| Parameters    Prolog    Metadata    Data    Epilog                                                                                                                                                                                                                                                                                                           |           |      |          |          |
| <pre>#***Begin: Generated Statements*** DIMENSIONDELETEALLELEMENTS('UnevenSubsidiary'); DIMENSIONSORTORDER('UnevenSubsidiary','ByName','ASCENDING','ByHierarchy','ASCENDING'); #***End: Generated Statements*** DIMENSIONDELETEALLELEMENTS('UnevenSubsidiary'); DIMENSIONSORTORDER('UnevenSubsidiary','ByName','ASCENDING','ByHierarchy','ASCENDING');</pre> |           |      |          |          |

6. Click the **Metadata** tab, and repeat step 5.

The result appears as follows:



The screenshot shows a software interface with a top navigation bar containing tabs: Data Source, Variables, Maps, Advanced, Schedule, Parameters, Prolog, **Metadata**, Data, and Epilog. The **Metadata** tab is currently selected. Below the tabs, there is a large text area containing the following code:

```

*****Begin: Generated Statements*****
DIMENSIONELEMENTINSERT('UnevenSubsidiary','vSubsidiary','n');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','vRegion','c');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','vTotalCompany','c');
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vRegion,vSubsidiary,1.000000);
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vTotalCompany,vRegion,1.000000);
*****End: Generated Statements****

DIMENSIONELEMENTINSERT('UnevenSubsidiary','vSubsidiary','n');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','vRegion','c');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','vTotalCompany','c');
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vRegion,vSubsidiary,1.000000);
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vTotalCompany,vRegion,1.000000);

```

There are no other statements we will need to update on the other tabs.

7. Click the **Variables** tab, and then change the **Contents** column for all variables to **Other**.

You need to copy the statements before changing the Contents to Other because once you change them to Other, the variables and any statements using them are removed from the Generated Statements.

8. Click the **Advanced** tab, and then click the **Prolog** tab.

The Generated Statements section should be empty but everything pasted below the Generated Statements section remains.

9. Click the **Metadata** tab, and ensure that the section between the Generated Statements is empty.

Next you will make some changes to the scripts on the Metadata tab.

You will add an IF statement to evaluate the vSubsidiary. If it does NOT match vRegion, put this element into a three-level hierarchy, if it does match put it into a two-level hierarchy. Since you only have two levels with a possible conflict you can stop here. Otherwise you could add multiple ELSEIF statements.

10. After the Generated Statements, type the following:

**IF (vSubsidiary@<>vRegion);**

**# create the 3 level hierarchy**

11. After the DimensionsElementComponentAdd functions, type the following comment:

**# assigns elements to parent and weights to elements in hierarchy**

Now you will create the ELSE block and close the conditional statement.

Note: For the next step, you can copy the last five rows from the IF block, to avoid typing all the rows. (In the first DimensionElementInsert function, change **C** to **N**. Delete the first Division ElementComponentAdd function. Change the comment lines)

12. At the end of the IF block, type the following:

**ELSE;**

```
# create the 2 level hierarchy changing the vRegion to be the leaf level element
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vRegion,'N');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vTotalCompany,'c');
# assign parent/child relationship and weights to elements in hierarchy
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vTotalComp
any,vRegion,1.000000);
ENDIF;
```

```
*****Begin: Generated Statements*****
*****End: Generated Statements*****
```

```
IF (vSubsidiary@<>vRegion):
# create the 3 level hierarchy

DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vSubsidiary,'n');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vRegion,'c');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vTotalCompany,'c');
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vRegion,vSubsidiary,1.000000);
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vTotalCompany,vRegion,1.000000);

# assigns elements to parent and weights to elements in hierarchy

ELSE:
# create the 2 level hierarchy changing the vRegion to be the leaf level element
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vRegion,'N');
DIMENSIONELEMENTINSERT('UnevenSubsidiary','','vTotalCompany,'c');

# assign parent/child relationship and weights to elements in hierarchy
DIMENSIONELEMENTCOMPONENTADD('UnevenSubsidiary',vTotalCompany,vRegion,1.000000);
ENDIF;
```

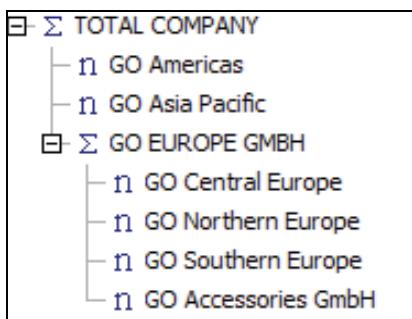
13. Click through the **Advanced** tabs, and then save and run the process.

The process runs without errors.

14. Close the **Turbo Integrator**.

15. Double-click the **UnevenSubsidiary** dimension.
16. Click **All** and then click **Hierarchy Sort**.

The result appears as follows:



17. Close the **Subset Editor** without saving.

### Results:

You created a Turbo Integrator process that terminates the process at different levels for different hierarchies. You then ran the process and viewed the dimension that was created.

## Use TM1 as a Data Source

- TM1 Cube View
  - choose:
    - use an existing view
    - create a new view (recommended)
- TM1 Dimension Subset
  - pick an existing subset

When you create the view in Turbo Integrator, the view is properly formatted for exporting. A view that has been formatted for viewing may not be properly formatted for exporting.

TM1 may be used as a source of data for external sources by exporting dimensions or cube views to text files. This allows data to be shared among a large variety of systems.

Sometimes, it is useful to have one cube push data into another cube. TI can do this by using a TM1 view as a data source. This allows data to move from one cube to another in a static manner as opposed to sharing data with a rule, which will dynamically update the data between cubes.

- Archive data (and freeze rules)
- Summarize data (update on a scheduled basis i.e. monthly)

## Demo 4: Export Data to ASCII Text Using Turbo Integrator

### Purpose:

You want to export data for Budget Version 1 from the Sales\_Plan cube to an ASCII file, so the data can be transferred to another system. You will make use of the Turbo Integrator TextOutput function to do this.

### Task 1. Set data source.

TM1 Server: **greatoutdoors**

TM1 Architect: **Server Explorer**

UserName: **admin**

Password: **<blank>**

1. Create a new process with Datasource Type: **TM1 CubeView**.
2. Beside **Data Source Name**, click **Browse**, and then click **greatoutdoors:Sales\_Plan**.
3. Click **Create View**, and then ensure that **Skip Consolidated Values**, and **Skip Zero/Blank Values** are selected.
4. Beside **Versions**, click **Subset Editor** , click **Budget Version 1**, and then click **OK** twice.
5. In the **TM1 Save View** box, type **zExport view**, and then click **OK**.  
You cannot chose to make this a private view because TI may only use public views.
6. Below **Select View**, click **zExport view**, and then click **OK**.

### Instructor Notes

It is a good idea to give the view a name that will identify it as a view used in a process or a system view. In this example, you precede the name with the letter z so it will be at the end of the list. Another option would be to name it something like, 'system.ExportView' or use another predetermined prefix. Alternatively, you could precede the name with a } and TM1 will automatically hide the display as a control object.

## 7. Click Preview.

The result appears as follows:

The screenshot shows a software interface for configuring a data source. At the top, there are tabs: Data Source, Variables, Maps, Advanced, and Schedule. The Data Source tab is active. On the left, a 'Datasource Type' panel lists several options with radio buttons: ODBC, Text, ODBO, Cube (selected), TM1 (selected), SAP, and None. To the right of this panel is a 'Data Source Name' field containing 'greatoutdoors:Sales\_Plan->zExport view' and a 'Browse...' button. Below the preview table, there is a 'Preview' button. The preview table has columns: Subsidiaries, Channels, Products, Months, Versions, and Sales\_Plan\_Meas. The first row shows: TOTAL COMPANY, ALL CHANNELS, TOTAL PRODUCTS, Total Year, Budget Version 1, and Quantity.

|   | Subsidiaries  | Channels     | Products       | Months     | Versions         | Sales_Plan_Meas |
|---|---------------|--------------|----------------|------------|------------------|-----------------|
| 1 | TOTAL COMPANY | ALL CHANNELS | TOTAL PRODUCTS | Total Year | Budget Version 1 | Quantity        |

## Task 2. Set up variables, and then add a function.

1. Click the **Variables** tab.
2. Beside **Value**, set the **Variable Type** to **String**, and then for all elements set the **Contents** column to **Other**.

When exporting data to text or ASCII files, all variables must be Strings, even the data. If the Value is not text, TI will generate an error message.

The result appears as follows:

|   | Variable Name       | Variable Type | Sample Value     | Contents |
|---|---------------------|---------------|------------------|----------|
| 1 | Subsidiaries        | String        | TOTAL COMPANY    | Other    |
| 2 | Channels            | String        | ALL CHANNELS     | Other    |
| 3 | Products            | String        | TOTAL PRODUCTS   | Other    |
| 4 | Months              | String        | Total Year       | Other    |
| 5 | Versions            | String        | Budget Version 1 | Other    |
| 6 | Sales_Plan_Measures | String        | Quantity         | Other    |
| 7 | Value               | String        | 58781344.000000  | Other    |

3. Click the **Advanced** tab, and then click on the **Metadata** tab.

You want to insert the Turbo Integrator TextOutput function.

This function writes a comma-delimited record to a text file.

`TextOutput(FileName, String1, String2, ...Stringn);`

Where FileName is the full path to the text file you are writing to and String1...Stringn corresponds to each field you want to create in the text file.

4. Type the following to the **Metadata** tab, below the Generated Statements:

```
TextOutput('TextOutput.csv', Subsidiaries, Channels, Products,  
Months, Versions, Sales_Plan_Measures, Value);
```

The result appears as follows:

A screenshot of a software interface showing a tabbed panel. The tabs are labeled 'Parameters', 'Prolog', 'Metadata' (which is highlighted in yellow), 'Data', and 'Epilog'. Below the tabs is a code editor window containing the following text:

```
#****Begin: Generated Statements***  
#****End: Generated Statements****  
  
TextOutput('TextOutput.csv', Subsidiaries, Channels, Products, Months, Versions, Sales_Plan_Measures, Value);
```

5. Click through all the tabs on the **Advanced** tab, save the process as **CreateTextOutput**, and then run the process.

The output file, TextOutput.csv is created on your TM1 data server (C:\Edcognos\P6502\GreatOutdoors\DataFiles\TextOutput.csv).

6. Browse to **C:\Edcognos\P6502\GreatOutdoors\DataFiles**, and then double-click **TextOutput.csv** to open in MS Excel.

|   | A           | B         | C       | D     | E         | F          | G        | H |
|---|-------------|-----------|---------|-------|-----------|------------|----------|---|
| 1 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | Quantity   | 47742    |   |
| 2 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | GROSS S    | 2020704  |   |
| 3 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | Returns ar | 88190.61 |   |
| 4 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | Cash Disc  | 65239.03 |   |
| 5 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | Volume Di  | 42530.7  |   |
| 6 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | Freight    | 202838.4 |   |
| 7 | GO Americas | Departmer | Cooking | G Jan | Budget Ve | COST OF    | 1497356  |   |
| 8 | GO Americas | Departmer | Cooking | G Feb | Budget Ve | Quantity   | 11935    |   |

7. Close TextOutput.csv and Turbo Integrator.

### Results:

You exported the data for Budget Version 1 from the Sales\_Plan cube to an ASCII file, so the data can be transferred to another system. You made use of the Turbo Integrator TextOutput function to do this.

## Scripting in Turbo Integrator Tips

- Users may run TI scripts:
  - pass variables into the scripts
  - use action buttons
- Views and subsets should be created in scripts.
- Views or subsets used only within a script, it should be removed after running the script.



Administrators can create TI scripts and users may run them (with appropriate security settings). Users can run scripts via Server Explorer, TM1 Web or use an Action Button.

Custom scripting should be planned and well documented. Doing so will make it much easier to maintain and update the model. With a little planning, your model will be much easier to maintain and update.

Even if you can create a view or subset manually, you may wish to create a TI script to recreate it. This will be easier to document and recreate as data is added or changed.

---

Action buttons are discussed in Appendix C. They allow users to run TI scripts in TM1 Web or MS Excel spreadsheets.

## Demo 5: Move Data Between Versions

### Purpose:

The Unit figures for Budget Version 1 in the Sales\_Plan cube need to be approved. You want to move the data from Budget Version 1 to Submit Version, so users can easily identify the finalized data.

You will manually script the processes to copy the data. You will perform the following tasks:

**Create Parameters to specify the source, target and month data**

**Create a view and subset to hold the data to copy**

**Load the data to the new location in the cube**

**Delete the views and subsets used to move data**

### Task 1. Create a new version.

1. Right-click the **Versions** dimension, and then click **Edit Dimension Structure**.

If you double-click Versions, you will open the dimension in Subset Editor.

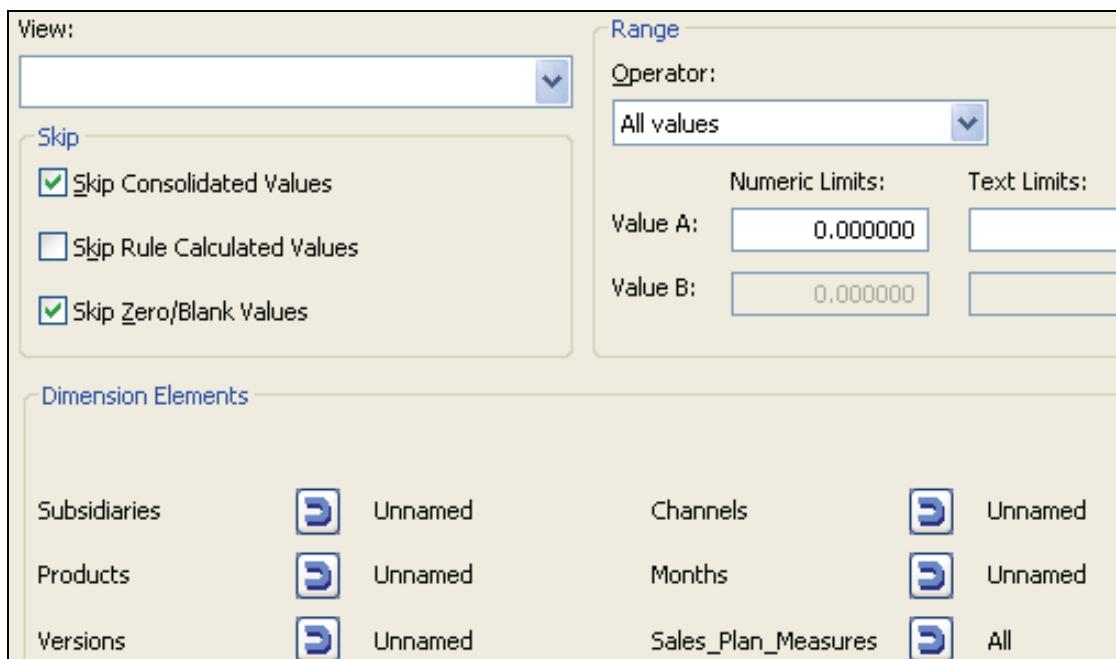
2. Click **Budget Version 2**.
3. Click **Edit** and then click **Insert Element**
4. Name it **Submit Version**, click **Add** and then click **OK**.
5. Save and close the **Dimension Editor**.

## Task 2. Set data source and variables.

1. Create a new process with Datasource Type: **TM1 CubeView**.
2. In the **Data Source Name** box, click **Browse**, click **greatoutdoors:Sales\_Plan**, and then click **Create View**.
3. Beside each of the following dimensions, click the **Subset Editor**, click **All**, click **Filter by Level**, click **0** and then click **OK** twice:
  - Subsidiaries
  - Products
  - Channels
  - Months
  - Versions

Leave Sales\_Plan\_Measures as All.

The results appear as follows:



4. Click **OK**, name the view as **ztrf\_MonthVersion** and then click **OK**.
5. Below **Select View**, click **ztrf\_MonthVersion**, and then click **OK**.
6. Click **Preview**.

The results appear as follows:

|   | Subsidiaries | Channels         | Products     | Months | Versions         | Sales_Plan_Meas |
|---|--------------|------------------|--------------|--------|------------------|-----------------|
| 1 | GO Americas  | Department Store | Cooking Gear | Jan    | Budget Version 1 | Quantity        |

7. Click the **Variables** tab.

8. Change the **Contents** column for all variables to **Other**.

The results appear as follows:

|   | Variable Name       | Variable Type | Sample Value     | Contents |
|---|---------------------|---------------|------------------|----------|
| 1 | Subsidiaries        | String        | GO Americas      | Other    |
| 2 | Channels            | String        | Department Store | Other    |
| 3 | Products            | String        | Cooking Gear     | Other    |
| 4 | Months              | String        | Jan              | Other    |
| 5 | Versions            | String        | Budget Version 1 | Other    |
| 6 | Sales_Plan_Measures | String        | Quantity         | Other    |
| 7 | Value               | Numeric       | 47742.000000     | Other    |

9. Click the **Advanced** tab.

### Task 3. Create the parameters.

1. On the tab, click **Insert** three times.

The results appear as follows:

| Parameters | Prolog  | Metadata      | Data            | Epilog |
|------------|---------|---------------|-----------------|--------|
| Parameter  | Type    | Default Value | Prompt Question |        |
| P0         | Numeric |               |                 |        |
| P1         | Numeric |               |                 |        |
| P2         | Numeric |               |                 |        |

2. Click on each parameter and rename them to:

P0 = pTargetMonth

P1 = pVersionTarget

P2 = pVersionSource

In this example, you have used the p to indicate this value comes from a Parameter and not a variable defined on the Variables tab. This will make it easier to identify its purpose in the script.

3. Click **String** under Type for each Parameter.

| Parameters     | Prolog | Metadata | Data          | Epilog          |
|----------------|--------|----------|---------------|-----------------|
| Parameter      | Type   |          | Default Value | Prompt Question |
| pTargetMonth   | String | ▼        |               |                 |
| pVersionTarget | String | ▼        |               |                 |
| pVersionSource | String | ▼        |               |                 |

4. Under Default Value type:

- **Jan**
- **Submit Version**
- **Budget Version 1**

These should match the elements in the Months and Versions dimensions.

5. Under Prompt Question, type the following:

- **Enter Month (Mmm)**
- **Enter Target Version**
- **Enter Version to Copy**

This will be displayed in the window that is displayed to the user when they run the script.

| Parameters     | Prolog | Metadata | Data             | Epilog               |
|----------------|--------|----------|------------------|----------------------|
| Parameter      | Type   |          | Default Value    | Prompt Question      |
| pTargetMonth   | String | ▼        | Jan              | Enter Month (Mmm)    |
| pVersionTarget | String | ▼        | Submit Version   | Enter Target Version |
| pVersionSource | String | ▼        | Budget Version 1 | Enter Version to Cop |

6. Click the **Prolog** tab.

## Task 4. Script the Prolog process.

1. Press **Enter** twice below #\*\*\*\*End: Generated Statements\*\*\*\*.
2. Type the following two comment lines and then press **Enter**:

**# This TI process copies Sales\_Plan Cube data from a source Month & Version to a target Month & Version.**

**#\*\*\*\*DECLARE VARIABLES\*\*\*\***

It is often helpful to open the Reference Guide for TI functions when scripting. Click Help, click Contents and Index and click TM1 TurboIntegrator Functions to see the list of available functions and their syntax.

You will now create two variables to hold the name of the cube and the name of the view & subset to hold the data to move in the cube.

3. Press **Enter**, and then type the following:

**Cube = 'Sales\_Plan';**

**Object = '\$Transfer' | Cube;**

4. Press **Enter** and then type the following:

**#\*\*\*\*TURN CUBE LOGGING ON\*\*\*\***

**#LogFlag = CUBEGETLOGCHANGES(Cube);**

**#CUBESETLOGCHANGES(Cube, 0);**

This line is optional and commented out but it may be useful if there is a problem moving the data. It turns Cube logging on when you remove the # from the 2nd and 3rd lines.

You should destroy the view before trying to create it. This way if the process failed before removing the view, it will be removed now without generating an error (and halting the process).

5. Press **Enter**, and then type the following:

**#\*\*\*\*DELETE OLD VIEW (IF ANY)\*\*\*\***

**VIEWDESTROY(Cube, Object);**

Destroy old subsets that will be created in this view.

6. Press **Enter**, and then type the following:

**#\*\*\*\*DELETE OLD SUBSETS (IF ANY)\*\*\*\***

**SUBSETDESTROY('Months', Object);**

**SUBSETDESTROY('Versions', Object);**

Create the new Subsets (for the target view) and add the elements from the user input (parameters).

7. Press **Enter**, and then type the following:

**#\*\*\*\*CREATE NEW SUBSETS\*\*\*\***

**SUBSETCREATE('Months', Object);**

**SUBSETELEMENTINSERT('Months', Object, pTargetMonth, 1);**

**SUBSETCREATE('Versions', Object);**

**SUBSETELEMENTINSERT('Versions', Object, pVersionTarget, 1);**

Create the new target View using the subsets created above.

8. Press **Enter**, and then type the following:

**#\*\*\*\*CREATE VIEW\*\*\*\***

**VIEWCREATE(Cube, Object);**

**VIEWSUBSETASSIGN(Cube, Object, 'Months', Object);**

**VIEWSUBSETASSIGN(Cube, Object, 'Versions', Object);**

Clear the view, destroy it and delete the subsets associated with it (as a precaution).

9. Type the following:

**#\*\*\*\*CLEAR VIEW\*\*\*\***

**VIEWZEROOUT(Cube, Object);**

**#\*\*\*\*DELETE THE CLEARED VIEW\*\*\*\***

**VIEWDESTROY(Cube, Object);**

**#\*\*\*\*DELETE ASSOCIATED SUBSETS\*\*\*\***

**SUBSETDESTROY('Months', Object);**

**SUBSETDESTROY('Versions', Object);**

Create new views to be used as the data source.

10. Type the following:

**#\*\*\*\*CREATE NEW VIEWS TO USE AS DATA SOURCES FOR  
NEXT STEP\*\*\*\***

**#\*\*\*\*CREATE SOURCE SUBSETS\*\*\*\***

**SUBSETCREATE('Months', Object);**

**SUBSETELEMENTINSERT('Months', Object, pTargetMonth, 1);**

**SUBSETCREATE('Versions', Object);**

**SUBSETELEMENTINSERT('Versions', Object, pVersionSource, 1);**

Create the source views and designate them as the source.

11. Type the following:

**#\*\*\*\*CREATE SOURCE VIEWS\*\*\*\***

**VIEWCREATE(Cube, Object);**

**VIEWSUBSETASSIGN(Cube, Object, 'Months', Object);**

**VIEWSUBSETASSIGN(Cube, Object, 'Versions', Object);**

**#\*\*\*\*REESTABLISH THE NEW VIEW AS THE DATA SOURCE\*\*\*\***

**DATASOURCECUBEVIEW = Object;**

In the above script you have:

- created a target view to zero out the data (and then destroyed it and its subsets)
- created a view for the data transfer based on the source (and then destroyed it and its subsets).

You typically add a destroy subset to:

- ensure the subset is based on the input parameter
- clean up the system so there will not be public views left on the system for users

Nothing is needed on the Metadata script.

## **Task 5. Script the Data process.**

1. Click the **Data** tab.
2. Add two returns below **#\*\*\*\*End: Generated Statements\*\*\*\***.

---

From C:\Edcognos\P6502\TextFiles, open Demo5\_Mod7.txt and copy the lines in between \*\*\*Prolog\*\*\* and \*\*\*Data Tab\*\*\*. Paste them into the Prolog script below the reserved section (do not copy the lines \*\*\*Prolog\*\*\* and \*\*\*Data Tab\*\*\*).

3. Type the following comment line and then press **Enter**:

### # Check if the Value is a String or Number

The following statements will check to see if you are copying numeric or string data.

Next, type the following conditional check using a TI Local Variable called **Value\_Is\_String**.

4. Type the following:

```
IF (Value_Is_String <> 0);
#-----String
```

```
CELLPUTS(SValue, Cube, Subsidiaries, Channels, Products,
pTargetMonth, pVersionTarget, Sales_Plan_Measures);
```

```
ELSE;
```

```
#-----Number
```

```
# Ignore values where the cell cannot be updated
```

```
IF (CELLISUPDATEABLE(Cube, Subsidiaries, Channels,
Products, pTargetMonth, pVersionTarget, Sales_Plan_Measures)=0);
ITEMSKIP;
```

```
ELSE;
```

```
CELLPUTN(Value, Cube, Subsidiaries, Channels, Products,
pTargetMonth, pVersionTarget, Sales_Plan_Measures);
```

```
ENDIF;
```

```
ENDIF;
```

Note: Any functions on this tab are executed for each line of data in the source.

Any dimensions or cubes that should be updated with data need to be created before the functions updating them may be processed.

5. Click the **Epilog** tab.

---

Open Demo5\_Mod7.txt and copy the lines in between \*\*\*Data Tab\*\*\* and \*\*\*Epilog Tab\*\*\* for Task 5. Paste them into the Prolog script below the reserved section (do not copy the lines \*\*\*Prolog\*\*\* and \*\*\*Epilog Tab\*\*\*).

For Task 6 copy the lines following \*\*\* Epilog Tab\*\*\*

## Task 6. Script the Epilog process.

1. After #\*\*\*\*End: Generated Statements\*\*\*\*, press **Enter** twice.
2. Type the following:

```
#****DELETE OLD VIEW (IF ANY)****
```

```
VIEWDESTROY(Cube, Object);
```

```
#****DELETE OLD SUBSETS (IF ANY)****
```

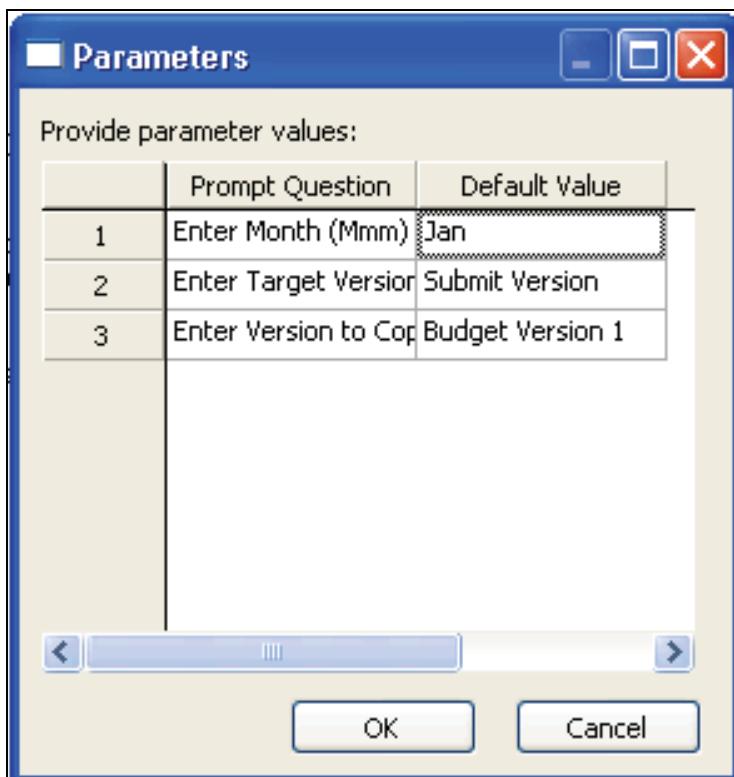
```
SUBSETDESTROY('Months', Object);
```

```
SUBSETDESTROY('Versions', Object);
```

3. Save the process as **Transfer\_Versions\_Month**.
4. Run the process.

5. Enter the following values:

- Jan
- Submit Version
- Budget Version 1



6. Click **OK**.
7. Close **Turbo Integrator**.
8. Double-click the **Sales\_Plan** cube.
9. Double-click the **Versions** dimension, click **All**, click **Submit Version** and then click **OK**.

10. Recalculate the view.

The result appears as follows:

| TOTAL COMPANY       |  | ALL CHANNELS     |        | Lanterns |        |        |        |        |        |        |        |        |        |  |  | Submit Version |
|---------------------|--|------------------|--------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--|--|----------------|
|                     |  | Months           |        |          |        |        |        |        |        |        |        |        |        |  |  |                |
| Sales_Plan_Measu    |  | Jan              | Feb    | Mar      | Apr    | May    | Jun    | Jul    | Aug    | Sep    | Oct    | Nov    | Dec    |  |  |                |
| Quantity            |  | 429,208.00       | 0.00   | 0.00     | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   |  |  |                |
| Unit Sale Price     |  | \$0.00           | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| GROSS SALES REV     |  | \$0.00           | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| Returns and Allow.  |  | \$554,201.70     | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| Cash Discount       |  | \$409,972.30     | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| Volume Discount     |  | \$267,268.39     | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| + TOTAL DISCOUNT    |  | \$1,231,442.39   | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| Freight             |  | \$1,274,663.86   | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| + NET SALES REVENUE |  | (\$2,506,106.25) | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| Unit Cost           |  | \$0.00           | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| COST OF SALES       |  | \$0.00           | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| + GROSS MARGIN      |  | (\$2,506,106.25) | \$0.00 | \$0.00   | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 | \$0.00 |  |  |                |
| GROSS MARGIN %      |  | 0.00             | 0.00   | 0.00     | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   | 0.00   |  |  |                |

You moved data from one portion of the cube to another by scripting.

11. Close the **Cube Viewer** without saving the changes if prompted.

### Results:

**You moved the data from Budget Version 1 to Submit Version in the Sales\_Plan cube, so users can easily identify the data to be approved. This script may be run by many users as needed from Server Explorer or from an Action Button.**

---

If you have difficulty copying the script from the Demo5\_Mod7.txt file, you can instead shut down the GreatOutdoors server, copy the Transer\_Versions\_Month3.pro script from the Module7 solutions. You will need to change the name of the view for the source data to be ztrf\_MonthVersion and then click Keep All Variables when prompted and run this script instead. Copying from the txt file should work but I have seen it generate errors on occasion.

# Use Chores with Turbo Integrator

- Chores execute:
  - one or more Turbo Integrator processes
  - at a user-defined frequency
- Chores may be:
  - run on demand
  - scheduled for automatic execution



## Demo 6: Create and Schedule a Chore

### Purpose:

The grocery store part of your enterprise updates its inventory regularly. You want to create a chore that creates and populates the products dimension and then creates the Product Type subsets. Next you want to schedule the chore to run once every week.

Since you already have processes to accomplish these tasks, you can use them in the chore.

### Task 1. Create a chore.

1. In Server Explorer, right-click **Chores**, and then click **Create New Chore**.
2. In the **Available** column, double-click **CreateProductsCSV** and **CreateProductType\_subsets** to add them to the **Selected** column.
3. Click **Next**.

You want to schedule this chore to run once every 7 days.

4. Click **Activate Chore**, and then beside **Days** type **7**.

If at some point in time, you decide that the chore no longer needs to be run, you can deactivate the chore.

5. Click **Finish**.
6. Save the chore as **CreateProducts\_ProductTypeSubsets**.

### Result:

You created a chore that creates and populates the products dimension and then creates the Product Type subsets. Next you scheduled the chore to run once every week.

### Instructor Notes

If you wish to edit a chore, you must first Deactivate it. This will prevent the chore from trying to execute while someone is making changes to it. You must also deactivate a chore when editing a Turbo Integrator process the chore will execute.

## Summary

- At the end of this module, you should be able to:
  - review Advanced tabs in Turbo Integrator
  - load custom data into a cube
  - add a subset to a dimension
  - use logic in scripts
  - export data to ASCII
  - move data between versions
  - construct chores

Cognos.  
software



## Workshop 1: Create Subsets within a Turbo Integrator Process

Your users would like to have subsets in the Subsidiaries dimension for each region. You have decided to automate this by creating a Turbo Integrator Process to create the subsets. You will then couple this process and the process to create the Subsidiaries dimension into a chore and schedule it to run at 1 a.m. the first of every month.

To accomplish this:

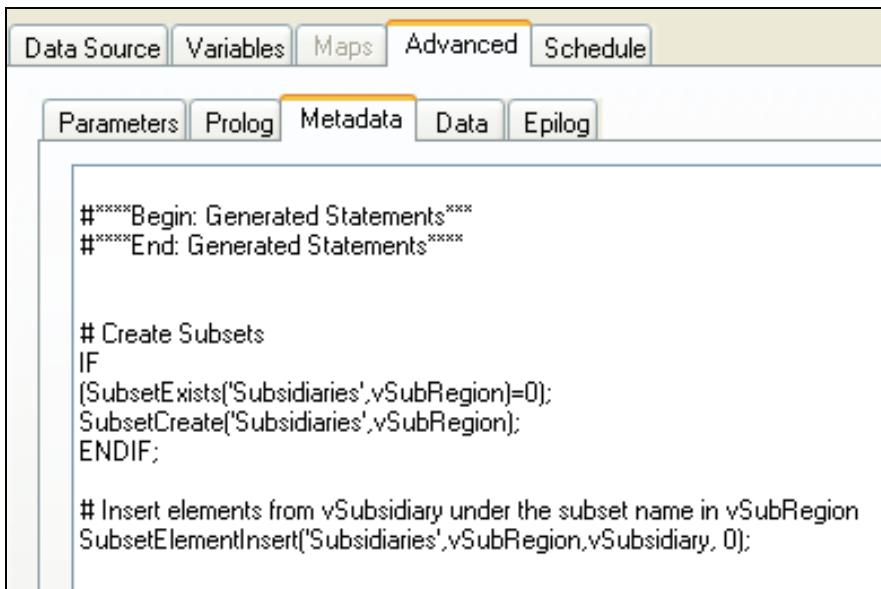
- Create a Turbo Integrator Process to automatically create subsets and apply elements to them.
- Create a chore that will execute:
  - the process to create the subsidiaries dimension
  - the process to create the subsets for the subsidiaries dimension.
- Schedule the process to run monthly.

## Task 1: Task Table

| Task                                              | Where to Work    | Hints                                                                                                                                                                             |
|---------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Create a Turbo Integrator process for subsets. | Turbo Integrator | <ul style="list-style-type: none"> <li>• CreateSubsidiariesDimCSV</li> <li>• Metadata Tab: SubsetCreate (), SubsetElementInsert()</li> </ul>                                      |
| 2. Create a chore.                                | Chores           | <ul style="list-style-type: none"> <li>• Contains CreateSubsidiariesCSV and CreateSubsidiariesRegion_Subset.</li> <li>• SubsidiariesChore</li> <li>• Run every 30 days</li> </ul> |

## Workshop 1:Workshop Results

The Metadata tab appears as follows:



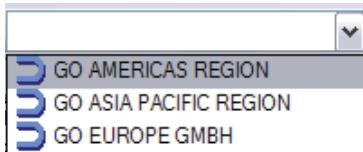
The screenshot shows the 'Advanced' tab selected in the top navigation bar. Below it, the 'Metadata' tab is also selected. The main area displays the following code:

```
#****Begin: Generated Statements****
#****End: Generated Statements****

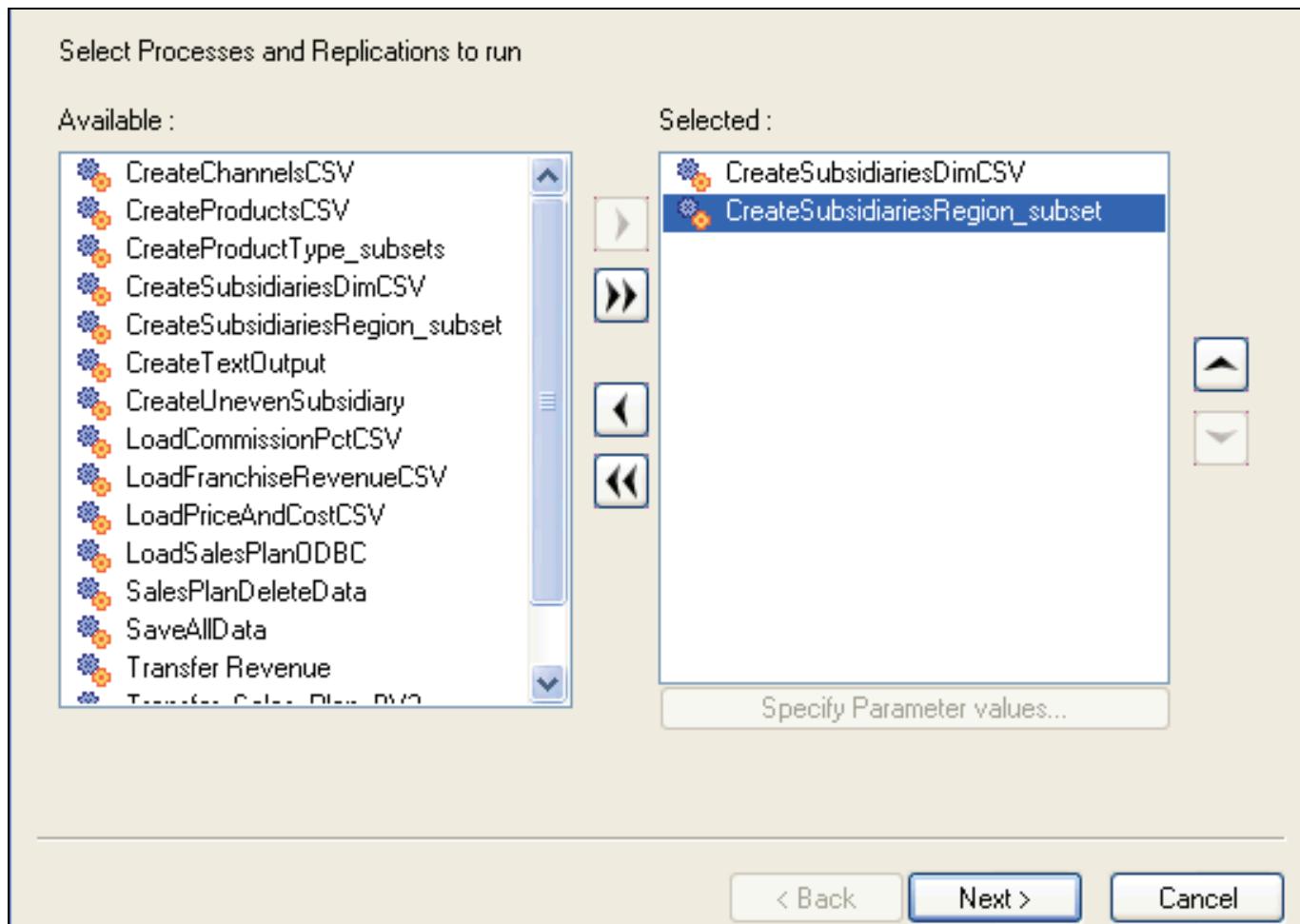
# Create Subsets
IF
(SubsetExists('Subsidiaries',vSubRegion)=0);
SubsetCreate('Subsidiaries',vSubRegion);
ENDIF;

# Insert elements from vSubsidiary under the subset name in vSubRegion
SubsetElementInsert('Subsidiaries',vSubRegion,vSubsidiary, 0);
```

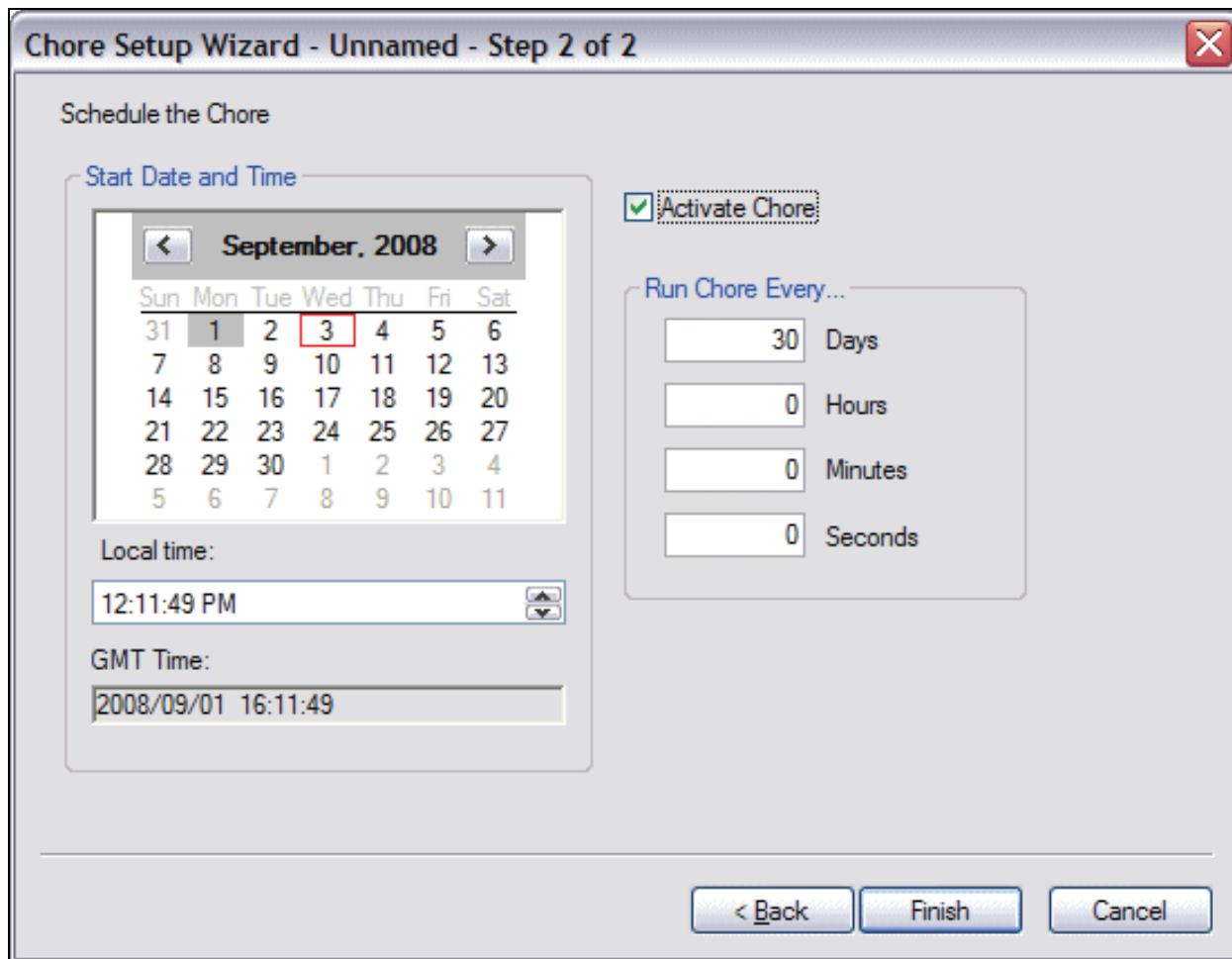
The subsets appear as follows:



The chore appears as follows:



The chore schedule appears as follows:



# Workshop 1: Step-by-Step Instructions

## Task 1. Create a Turbo Integrator process for subsets.

1. Open **CreateSubsidiariesDimCSV** process.
2. Click the **Maps** tab and then click **Dimensions**.
3. Change the Action to **Update**.
4. Click the **Advanced** tab, and then click all tabs to regenerate the scripts.
5. Click **File** and then click **Save As**.
6. In the name box type **CreateSubsidiariesRegion\_Subset** and then click **OK**.
7. Click the **Variables** tab and change the contents to **Other** for all.
8. Click the **Advanced** tab, and then click the **Metadata** tab.
9. Type the following (after the Generated Statements):

**# Create Subsets**

**IF**

**(SubsetExists('Subsidiaries',vSubRegion)=0);**

**SubsetCreate('Subsidiaries',vSubRegion);**

**ENDIF;**

10. Press **Enter** and type the following (below **ENDIF**):

**# Insert elements from vSubsidiary under the subset name in  
vSubRegion**

**SubsetElementInsert('Subsidiaries',vSubRegion,vSubsidiary, 0);**

11. Save and run the process.

12. Click **OK** when completed successfully.

If it does not complete successfully, you may need to delete the subsets before running a second time.

You can also add SubsetDestroy statements prior to the SubsetCreate statements to remove them before recreating them.

13. Close the process and open the **Subsidiaries** dimension in the Subset editor.
14. Click the **Select Subset** list to see the Subset list.

## **Task 2. Create a chore.**

1. In Server Explorer, right-click **Chores** and click **Create New Chore**.
2. Double-click:
  - **CreateSubsidiariesDimCSV**
  - **CreateSubsidiariesRegion\_Subset**
3. Click **Next**.
4. Click **Activate Chore**.
5. Under **Run Chore Every**, beside **Days**, type **30**.
6. Click **Finish**.
7. Save as **SubsidiariesChore**.
8. Close all open windows and then close the TM1 server, saving changes when prompted.

**Information Management**



## Customize Drill Paths

IBM Cognos TM1 9.5



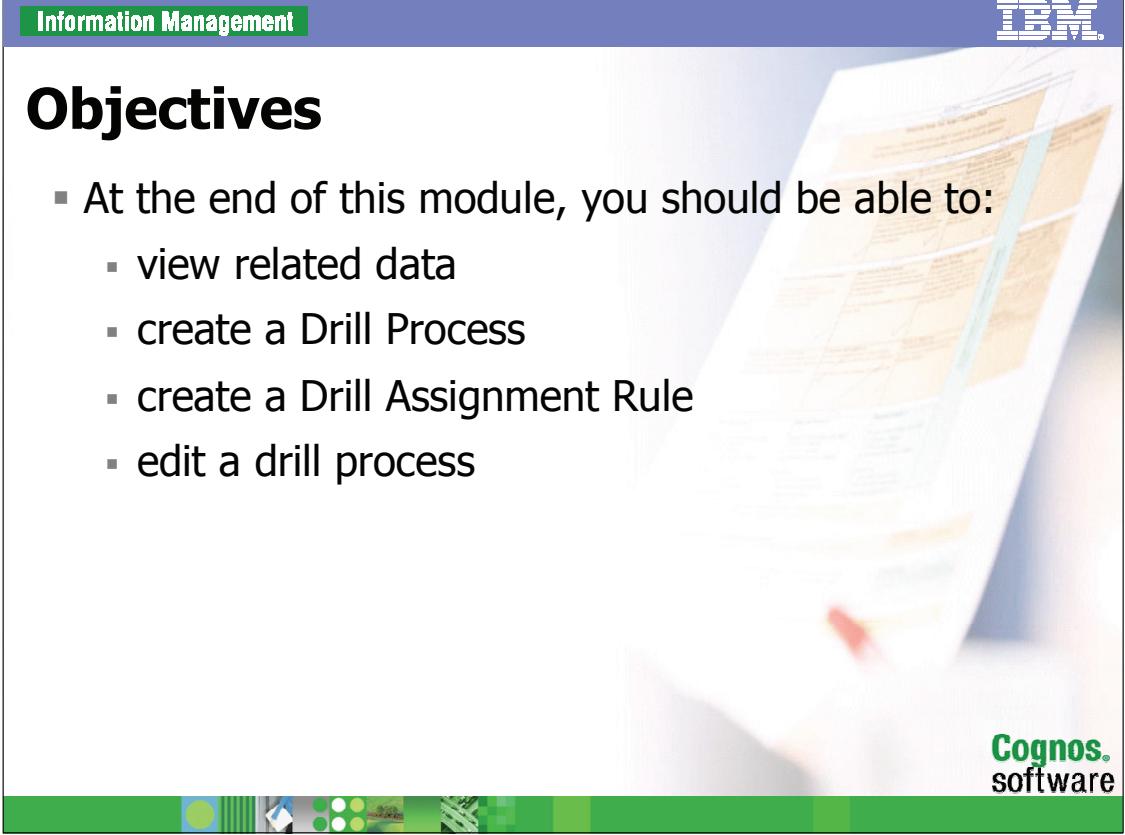
**Cognos.**  
software

© 2009 IBM Corporation



# Objectives

- At the end of this module, you should be able to:
  - view related data
  - create a Drill Process
  - create a Drill Assignment Rule
  - edit a drill process



Cognos.  
software



**INTERACTION - Star Sticker:** Use a star next to each objective when discussing it

## View Related Data

- Drill through from a cell to:
  - more detail
  - other relevant information



Cognos  
software

If a cell is associated with more than one drill processes, a list of available processes will appear when the drill-through process is executed.

A TM1 drill-through process can drill to a cube view, an Open DataBase Connectivity (ODBC) data source or any other data source supported by Turbo Integrator.

---

When drilling to an external source you must have the connection information and any SQL or queries formatted properly for the source data. You may wish to test the ODBC connection and/or SQL statements outside of TM1 to ensure they are valid for your data source.

## Create a Drill Path

- Drill-through consists of two components
  - drill process to connect
  - drill assignment rule

The drill process defines the detailed data you want to associate with a cell.

- It populates parameters with elements from each dimension in the originating cube which may be used to limit the resulting data.
- It specifies the data source and any connection information.

The drill assignment rule defines the relationship between the cell and the detailed data.

- A string rule provides a list of drill processes.
- Place the list of processes in single quotes, separated by commas.

It is a good idea to drill-through to relational data sources only from leaf-level cells. If not, the relational query could be quite large and complicated due to differences in the data structures. Use view to view drill through for higher level drill through.

**INTERACTION - X Sticker:** place a sticker next to each bullet as it is discussed.

## Demo 1: Create a Drill-through Path to a View in another Cube

### Purpose:

You need to look at additional Price and Cost information when reviewing the Unit Sale Price in the Sales\_Plan cube. You will create a drill path from the Sales\_Plan cube to go to the default view of the Price\_and\_Cost cube.

### Task 1. Create a public default view of the Price\_and\_Cost cube.

TM1 Server: **greatoutdoors**

TM1 - Architect: Server Explorer

UserName: **admin**

Password: <blank>

1. Ensure that the greatoutdoors TM1 server is started in the task bar, and then launch **TM1 Architect: Server Explorer**.
2. Expand **TM1**, double-click **greatoutdoors**, in the **UserName** box, type **admin**, and then click **OK**.
3. Expand **Cubes**.
4. Double-click the **Price\_and\_Cost** cube.  
You will create a view to drill to and then map to it later.
5. Swap the **Price\_and\_Cost\_Measures** dimension with the **Months** dimension.
6. In the **Channels** dimension, select **ALL CHANNELS**, and then in the **Versions** dimension, select **Budget Version 1**.
7. Double-click **Months** dimension, click **All**, select **Total Year**, and then click **OK**.

8. Double-click the **Price\_and\_Cost\_Measures** dimension, Ctrl+click the **Unit Sale Price** and **Unit Cost** elements and then click **OK**.
9. Click **Recalculate**.

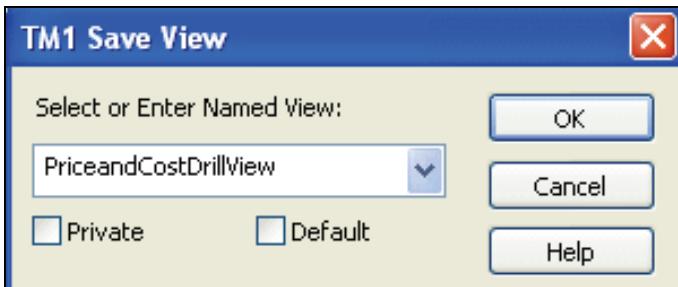
The result appears as follows:

The screenshot shows a TM1 interface with three dropdown menus at the top: 'ALL CHANNELS', 'Budget Version 1', and 'Total Year'. Below these is a table titled 'Price\_and\_Cost\_Measures' with the following data:

| Products         | Unit Sale Price | Unit Cost |
|------------------|-----------------|-----------|
| -- TOTAL PRODUCT | \$166.63        | \$110.08  |
| + CAMPING EQL    | \$159.04        | \$126.55  |
| + GOLF EQUIPM    | \$445.31        | \$254.85  |
| + MOUNTAINEER    | \$112.91        | \$81.35   |
| + OUTDOOR PR     | \$7.84          | \$4.51    |
| + PERSONAL AC    | \$103.23        | \$70.72   |

10. Click **File** and then click **Save As**.
11. In the **Select or Enter Named View** box, type **PriceandCostDrillView**, and then clear the **Private** box.

The result appears as follows:



12. Click **OK**, and then close the view.

### Instructor Notes

You need a public view for drill-through.

## Task 2. Create the drill process.

1. In Server Explorer, right-click the **Sales\_Plan** cube, point to **Drill**, and then click **Create Drill Process**.

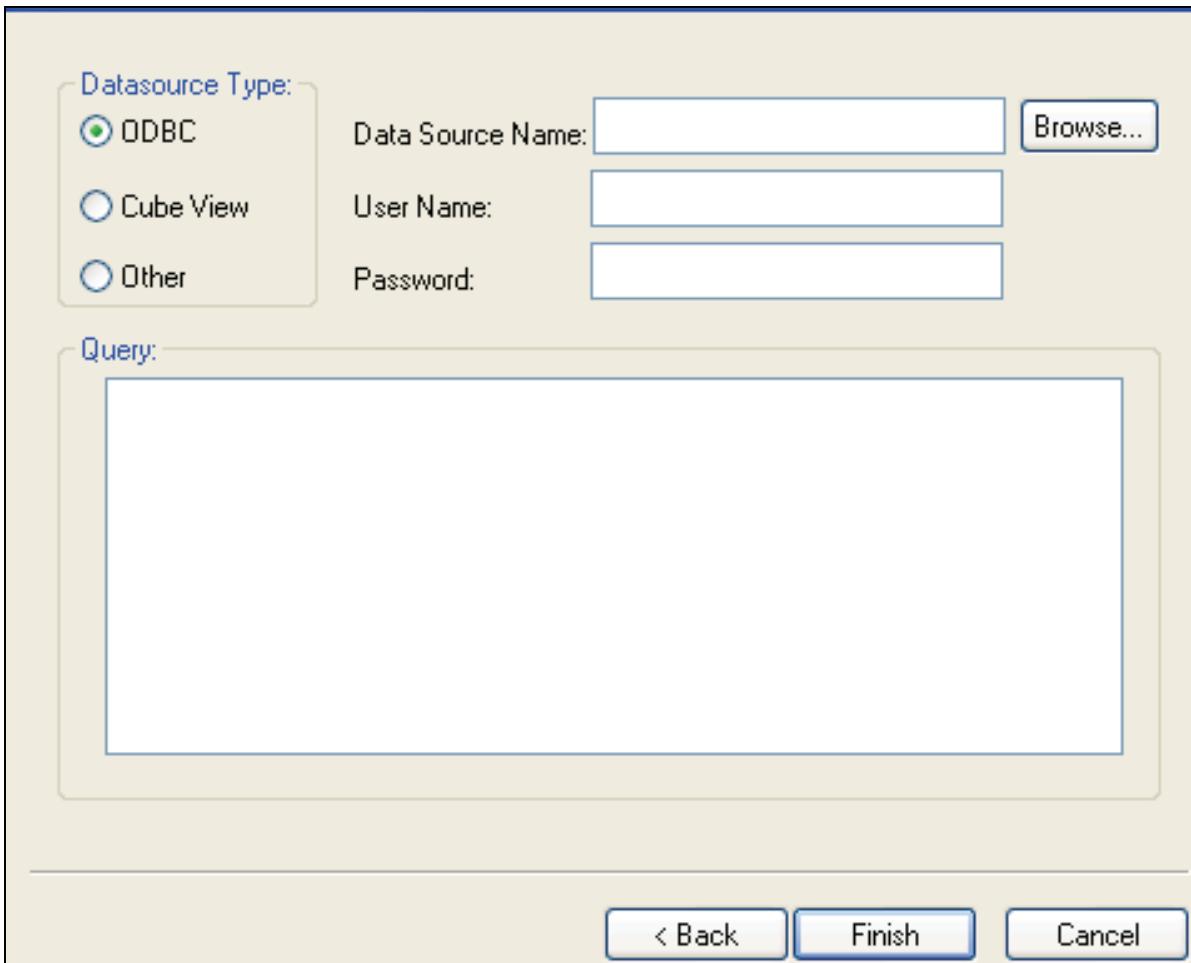
Ensure that your Parameter Values match the results below:

|  | Parameter Name     | Parameter Value  |
|--|--------------------|------------------|
|  | cubename           | Sales_Plan       |
|  | Subsidiaries       | TOTAL COMPANY    |
|  | Channels           | ALL CHANNELS     |
|  | Products           | TOTAL PRODUCTS   |
|  | Months             | Total Year       |
|  | Versions           | Budget Version 1 |
|  | Sales_Plan_Measure | Quantity         |

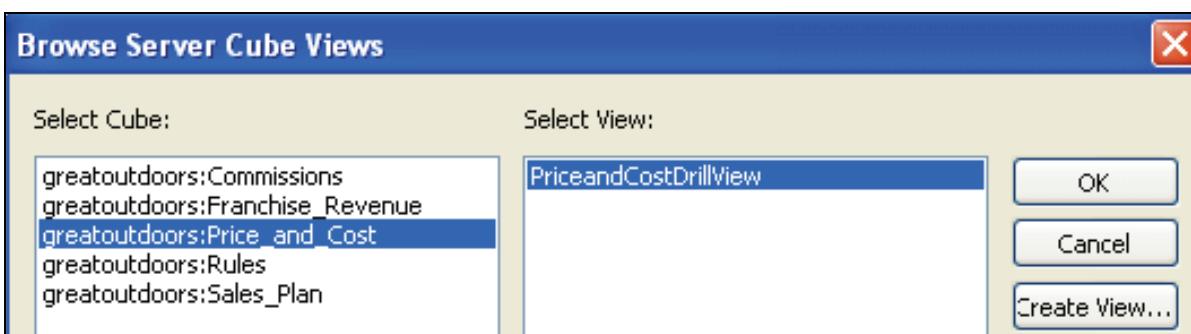
If they do not, click the Parameter Name and then click Select Element to open the Subset Editor window and choose the correct element.

2. In the Drill Process Setup Wizard click **Next**.

3. Click **Cube View** and then click **Browse**.



4. Under Select Cube, click **greatoutdoors:Price\_and\_Cost**, under Select View, click **PriceandCostDrillView**.



5. Click **OK**, and then click **Finish**.
6. In the Name box, type **SalesPlanToPriceandCost**, and then click **OK**.

### **Task 3. Create a drill assignment rule.**

1. In Server Explorer, right-click the **Sales\_Plan** cube, point to **Drill**, and then click **Create Drill Assignment Rules**.
2. Click the **Brackets** and then double-click **Sales\_Plan\_Measures**.
3. Click **Unit Sale Price** and then click **OK**.
4. Click to the right of the bracket, and then type = **S: 'SalesPlanToPriceandCost'**;

This indicates the drill process that will be run when drilling from any cell in the Unit Sale Price dimension.

The result appears as follows:

```
[1] ['Unit Sale Price'] = S: 'SalesPlanToPriceandCost';
```

5. Click **Save**, and then close the Rules Editor.

When you create a drill assignment rule, it places the rule on a control cube called **{CubeDrill\_Sales\_Plan** (**{CubeDrill\_cubename**}).

### **Task 4. Drill to the Price\_and\_Cost cube.**

1. In Server Explorer, double-click the **Sales\_Plan** cube.
2. Double-click **Subsidiaries**, expand **GO AMERICAS REGION**, select **GO Americas**, and then click **OK**.
3. Under **Channels**, click **Department Store**.
4. Under **Products**, click **Cooking Gear**.
5. Under **Versions**, click **Budget Version 1**.

6. Ensure that the dimensions are arranged as follows:

| GO Americas         | ▼              | Department Store | ▼              |
|---------------------|----------------|------------------|----------------|
| Cooking Gear        | ▼              | Budget Version 1 | ▼              |
| Months              |                |                  |                |
| Sales_Plan_Measu    | Jan            | Feb              | Mar            |
| Quantity            | 47,742.00      | 11,935.00        | 43,090.00      |
| Unit Sale Price     | \$42.33        | \$42.33          | \$42.33        |
| GROSS SALES REV     | \$2,020,704.02 | \$505,154.84     | \$1,823,805.80 |
| Returns and Allow.  | \$88,190.61    | \$22,047.61      | \$79,598.51    |
| Cash Discount       | \$65,239.03    | \$16,310.20      | \$58,883.46    |
| Volume Discount     | \$42,530.70    | \$10,632.99      | \$38,387.10    |
| + TOTAL DISCOUNT    | \$195,960.34   | \$48,990.80      | \$176,869.07   |
| Freight             | \$202,838.37   | \$50,709.49      | \$183,076.58   |
| + NET SALES REVENUE | \$1,621,905.31 | \$405,454.55     | \$1,463,860.14 |
| Unit Cost           | \$31.36        | \$31.36          | \$31.36        |
| COST OF SALES       | \$1,497,356.22 | \$374,323.37     | \$1,351,453.21 |
| + GROSS MARGIN      | \$124,549.09   | \$31,131.18      | \$112,406.93   |
| GROSS MARGIN %      | 0.00           | 0.00             | 0.00           |
| Seasons             |                |                  |                |

7. Click **Recalculate**, and then save the view as **DrillView**.

8. Right-click **Unit Sale Price** for Jan, and then click **Drill**.

The result appears as follows:

**Cube Viewer: greatoutdoors->Sales\_Plan->DrillView [Private]**

File Edit View Options Help

DrillView

GO Americas Department Store

Cooking Gear Budget Version 1

| Sales_Plan_Measu | Jan     | Feb     | Mar     | Apr     | May     | Jun     |
|------------------|---------|---------|---------|---------|---------|---------|
| Quantity         | 47742   | 11935   | 43090   | 17024   | 4739    |         |
| Unit Sale Price  | 42.3255 | 42.3255 | 42.3255 | 42.3255 | 42.3255 | 42.3255 |

**Cube Viewer: greatoutdoors->Price\_and\_Cost->PriceandCostDrillView**

File Edit View Options Help

PriceandCostDrillView

Department Store Budget Version 1 Jan

| Products         | Unit Sale Price | Unit Cost |
|------------------|-----------------|-----------|
| -- TOTAL PRODUCT | \$174.87        | \$115.28  |
| + CAMPING EQL    | \$168.43        | \$134.52  |
| + GOLF EQUIPM    | \$452.70        | \$259.08  |
| + MOUNTAINEER    | \$119.20        | \$86.08   |
| + OUTDOOR PR     | \$7.80          | \$4.41    |
| + PERSONAL AC    | \$103.82        | \$70.88   |

Here you can see the Department Store and Jan data was pulled over from the Sales\_Plan cube. The original PriceandCostDrillView had ALL CHANNELS and Total Year in the title.

9. Close the **PriceandCostDrillView**.

10. In the Sales\_Plan cube, right-click in **Quantity for January**.

You will not be able to click Drill because you have not made your drill process available to elements other than the Unit Sale Price.

11. Close all Cube Viewer windows without saving the changes if prompted.

**Results:**

**You needed to look at additional Price and Cost information when reviewing the Unit Sale Price in the Sales\_Plan cube. You created a drill path from the Sales\_Plan cube to go to the default view of the Price\_and\_Cost cube.**

## Demo 2: Create a Drill Path to an ODBC Source

### Purpose:

In addition to looking at the data in the Price\_and\_Cost cube, you will need to review the source data. You will create another drill path that will be available to users at the leaf level elements to review data in the ODBC source from which you loaded the data. You will use the same ODBC connection created to load the data. This process will only be available to the leaf elements in the cube. You will use a function to determine the leaf elements of the cube.

### Task 1. Create the drill process.

1. In Server Explorer, right-click on the **Sales\_Plan** cube, point to **Drill**, and then click **Create Drill Process**.
2. In the **Create Drill Process Wizard**, click **Next**.
3. In the Drill Process Setup Wizard, ensure **ODBC** is selected, and then click **Browse**.
4. In the ODBC Data Sources list, click **Sales\_Plan**, and then click **OK**.
5. Click **Finish**.
6. In the Name box, type **SalesPlanToSalesPlanXLS**, and then click **OK**.

### Task 2. Edit the drill process.

1. In Server Explorer, right-click the **Sales\_Plan** cube, point to **Drill**, and then click **Edit Drill Process**.
2. In the Select list, click **salesplantosalesplanxls**, and then click **OK**.
3. Click **OK** to dismiss the error message.

4. In the **Query** box, type:

```
SELECT * FROM Sales
WHERE Subsidiaries = '?Subsidiaries?'
AND Channel = '?Channels?'
AND "Product Type" = '?Products?'
AND Version = '?Versions?'
```

5. Click **Preview**.

You will not see any results other than the column names from the data source. The result appears as follows:

The screenshot shows the 'Data Source' configuration window with the 'Advanced' tab selected. On the left, under 'Datasource Type', 'QOdbc' is selected. The 'Data Source Name' is set to 'Sales\_Plan'. The 'Query' field contains the previously typed SQL SELECT statement. A checkbox 'Use Unicode' is checked. At the bottom, there is a preview table with columns: Subsidiaries, Channel, Product Type, Version, and Sales Staff.

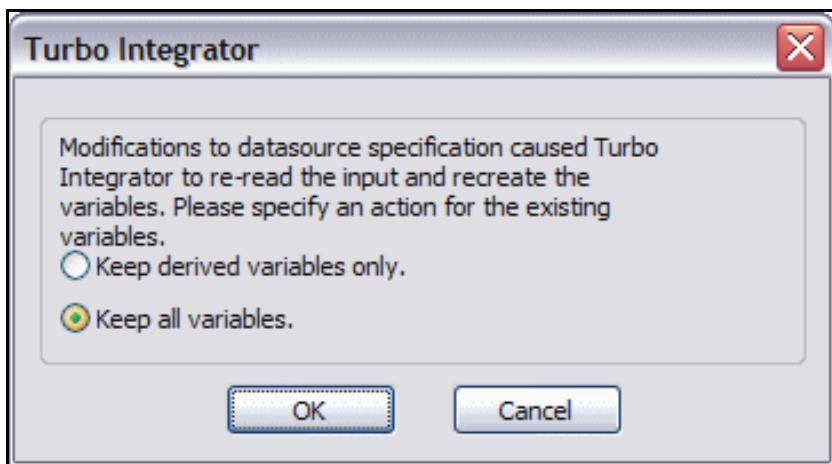
|  | Subsidiaries | Channel | Product Type | Version | Sales Staff |
|--|--------------|---------|--------------|---------|-------------|
|  |              |         |              |         |             |

---

In the query, the column names to the left of the = refer to the columns in the data source. The reference to the right (enclosed in single quotes and question marks) refer to the TM1 dimension names in the cube.

6. Click **Save**.

The results appear as follows:



7. Ensure that **Keep all variables** is selected, and then click **OK**.
8. Close **Turbo Integrator**.

### **Task 3. Edit the drill assignment rule.**

1. In Server Explorer, right-click **Sales\_Plan**, point to **Drill**, and then click **Edit Drill Assignment Rules**.
2. Place your cursor at the end of the existing rule, and then press **Enter** twice.
3. On row 3 type **[] = S:IF (ISLEAF=1, 'SalesPlanToSalesPlanXLS', '')**;

The results appear as follows:

```
[ 'Unit Sale Price' ] = S: 'SalesPlanToPriceandCost';

[] = S: IF (ISLEAF=1, 'SalesPlanToSalesPlanXLS', '') ;
```

Note: Enter two single quotes before the closing bracket, not a double quote.

4. Click **Save** and close the editor.

#### **Instructor Notes**

If you need to give people a list of drill paths, enclose all the drill process names in single quotes and separate them by commas.

Example: S: 'DrillProcess1, DrillProcess2, ...DrillProcessN';

## Task 4. Drill to the ODBC source.

1. Open DrillView in the cube viewer.
2. Double-click Months.
3. Click All, expand all four quarters and then click OK.
4. Click Recalculate.
5. Right-click Quantity for Jan and then click Drill.

The result appears as follows:

The screenshot shows the 'Cube Viewer: greatoutdoors->Sales\_Plan->DrillView [Private]' window. The main grid displays monthly sales data for 'Cooking Gear' across six months (Total Year, Q1, Jan, Feb, Mar, Q2). A specific cell for 'Quantity' in January is selected. A 'Relational Drill Through' dialog box is open, listing 9 rows of data corresponding to the selection. The columns in the dialog are Subsidiaries, Channel, Product Type, and Version.

|   | Subsidiaries | Channel          | Product Type | Version        |
|---|--------------|------------------|--------------|----------------|
| 1 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 2 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 3 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 4 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 5 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 6 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 7 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 8 | GO Americas  | Department Store | Cooking Gear | Budget version |
| 9 | GO Americas  | Department Store | Cooking Gear | Budget version |

6. Right-click **Unit Sale Price** for **Jan** and then click **Drill**.

The default drill view of **Price\_and\_Cost** appears.

7. Right-click **Quantity** for **Q1**.

The Drill menu item is not available. This is because we used the ISLEAF function in the rule to only assign the drill rule to leaf level or base level elements.

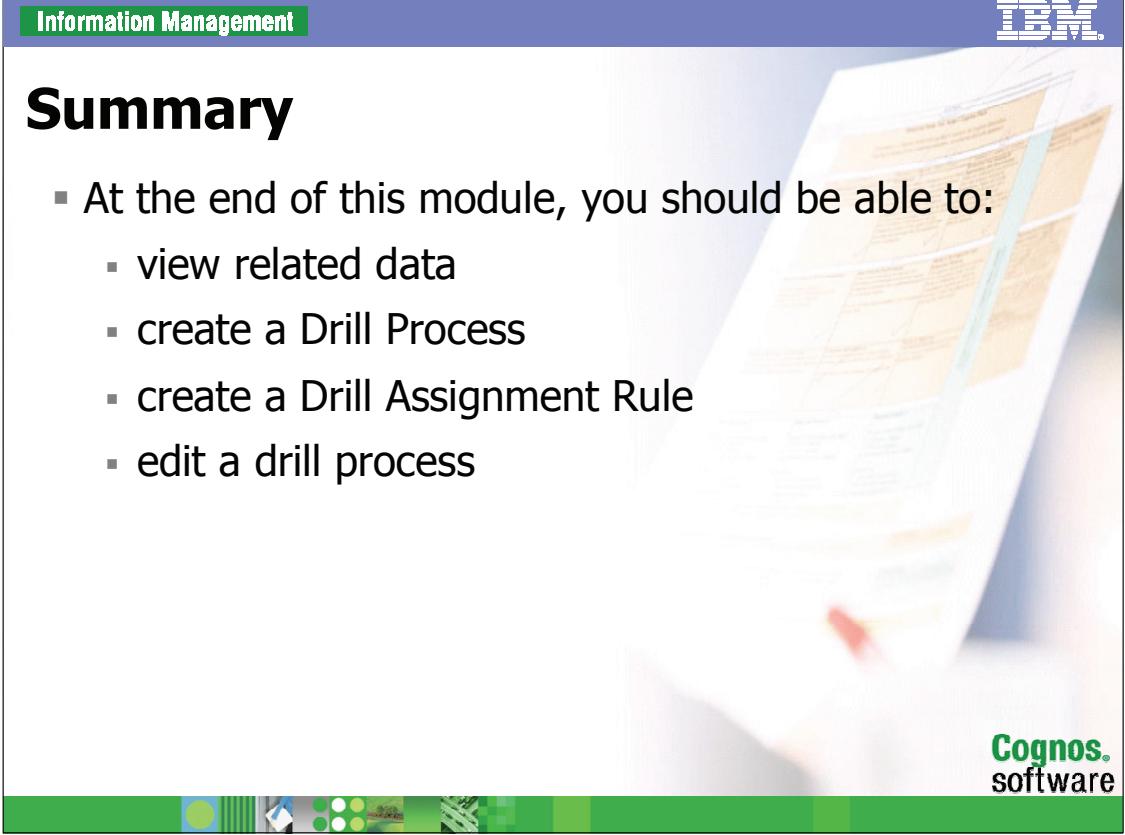
8. Close all **Cube Viewer** windows, without saving if prompted.

**Result:**

**You have created a drill process to display relational data from an ODBC source when requested from a leaf element in a cube.**

## Summary

- At the end of this module, you should be able to:
  - view related data
  - create a Drill Process
  - create a Drill Assignment Rule
  - edit a drill process



Cognos.  
software



## Workshop 1: Create Drill Paths for Commissions

Your users have expressed a desire to confirm data in the Sales\_Plan default view as they review Commissions. You will create a public default view of the Sales\_Plan cube and make it available to the Commissions cube. You will also give users the option to drill to the text in the source file as another option.

To accomplish this:

- Create a public default view of the Sales\_Plan cube (if it does not already exist).
- Create a drill process from the Commissions cube called CommissionsToSalesPlan to open the default Sales\_Plan view.
- Create a drill process from the Commissions cube called CommissionsToSalesPlan.
- Create a drill assignment rule from Commissions that is available at the leaf level cells only.

## Workshop 1: Task Table

| Task                                                      | Where to Work           | Hints                                                                                                                                     |
|-----------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Create a default view of Sales_Plan.                   | Sales_Plan Cube Viewer  | <ul style="list-style-type: none"> <li>Save a public default view.</li> </ul>                                                             |
| 2. Create a drill process to the Sales_Plan default view. | Drill Process           | <ul style="list-style-type: none"> <li>Commissions cube</li> <li>Create Drill Process</li> <li>Name it CommissionsToSalesPlan.</li> </ul> |
| 3. Create a drill assignment rule.                        | Drill Assignment rule   | <ul style="list-style-type: none"> <li>[ ] = S: IF ((ISLEAF=1), 'CommissionsToSalesPlan', "");</li> </ul>                                 |
| 4. Create a view to test the Drill.                       | Commissions Cube Viewer | <ul style="list-style-type: none"> <li>Create a view displaying leaf level cells</li> <li>Drill should be available.</li> </ul>           |

## Workshop 1: Workshop Results

The result of Task 2 appears as follows:

|                                            |                                                     |
|--------------------------------------------|-----------------------------------------------------|
| <b>Datasource Type:</b>                    |                                                     |
| <input type="radio"/> ODBC                 | Data Source Name: greatoutdoors:Sales Plan->Default |
| <input checked="" type="radio"/> Cube View |                                                     |
| <input type="radio"/> Other                |                                                     |

The result of Task 3 appears as follows:

```
[ ] = S: IF ((ISLEAF=1), 'CommissionsToSalesPlan', '');
```

The result Task 4 Step 6 appears as follows:

| Budget Version 1 |                    | Gross Sales Revenue |                  |                  |
|------------------|--------------------|---------------------|------------------|------------------|
|                  |                    | Months              |                  |                  |
| Subsidiaries     | -- Total Year      | -- Q1               | Jan              | Feb              |
| -- TOTAL COMPANY | \$5,704,481,151.42 | \$1,021,889,848.84  | \$277,163,120.35 | \$208,866,645.09 |
| -- GO AMERICAS   | \$1,267,006,475.25 | \$280,560,372.96    | \$108,379,815.32 | \$65,052,876.92  |
| GO Americas      | \$1,267,006,475.25 | \$280,560,372.96    | \$108,379,815.32 | \$65,052,876.92  |
| + GO ASIA PACI   | \$1,167,687,954.61 | \$300,204,692.56    | \$86,556,090.71  | \$82,376,059.56  |
| + GO EUROPE G    | \$3,269,786,721.56 | \$441,124,783.32    | \$82,227,214.31  | \$61,437,708.61  |

The result Task 4 Step 7 appears as follows:

| GO Americas        | ALL CHANNELS   | Lanterns       | Budget Version 1 |                |                |                |           |
|--------------------|----------------|----------------|------------------|----------------|----------------|----------------|-----------|
|                    |                | Months         |                  |                |                |                |           |
| Sales_Plan_Measu   | Jan            | Feb            | Mar              | Apr            | May            | Jun            | Jul       |
| Quantity           | 187,367.00     | 70,471.00      | 250,115.00       | 174,469.00     | 89,604.00      | 167,097.00     | 134,0     |
| Unit Sale Price    | \$29.66        | \$28.93        | \$29.68          | \$29.32        | \$29.43        | \$29.30        | \$        |
| GROSS SALES REV    | \$5,557,633.83 | \$2,038,570.06 | \$7,422,169.37   | \$5,116,084.94 | \$2,637,254.55 | \$4,895,533.55 | \$3,983,8 |
| Returns and Allow. | \$249,994.11   | \$94,025.68    | \$333,715.80     | \$232,781.48   | \$119,552.22   | \$222,945.70   | \$178,8   |
| Cash Discount      | \$184,933.58   | \$69,555.94    | \$246,866.68     | \$172,200.96   | \$88,439.25    | \$164,924.73   | \$132,2   |
| Volume Discount    | \$120,561.62   | \$45,344.71    | \$160,936.47     | \$112,259.98   | \$57,655.70    | \$107,519.18   | \$86,2    |
| + TOTAL DISCOUNT   | \$555,489.31   | \$208,926.33   | \$741,518.95     | \$517,242.42   | \$265,647.17   | \$495,389.61   | \$397,2   |

## Workshop 1: Step-by-Step

### Task 1. Create a default view of Sales\_Plan.

1. Double-click **Sales\_Plan**.
2. From the **File** menu, click **Save As**.
3. Click **Default** and then click **Private**, to clear it.  
This saves the current view as the Public Default view.
4. Click **OK**, and then close the view.

### Task 2. Create a drill process to the Sales\_Plan default view.

1. Right-click the **Commissions** cube, point to **Drill**, and then click **Create Drill Process**.
2. Click **Next**.
3. Click **Cube View**, and then click **Browse**.
4. Click **greatoutdoors: Sales\_Plan**, click **Default**, and then click **OK**.
5. Click **Finish**.
6. Name the process **CommissionsToSalesPlan**, and then click **OK**.

### Task 3. Create a drill assignment rule.

1. Right-click the **Commissions** cube, point to **Drill**, and then click **Create Drill Assignment Rules**.
2. Type the following:  
`[ ] = S: IF ((ISLEAF=1), 'CommissionsToSalesPlan', "");`
3. Save the rule and close the editor.

## Task 4. Create a view to test the Drill.

1. Double-click the **Commissions** cube.
2. Swap the **Commission\_Measures**, and **Months** dimensions.
3. Swap the **Versions**, and **Subsidiaries** dimensions.
4. Place **Versions: Budget Version 1** in the title.
5. Place **Commissions Measures: Gross Sales Revenue** in the title.
6. Click **Recalculate**.
7. Expand **GO AMERICAS REGION**, right-click **GO Americas** for Jan and then click **Drill**.  
You may change the view to show Total Years and Quarters.
8. Close all open windows without saving, and then close the TM1 Server, saving if prompted.

