SAP® Business Planning and Consolidation

- A complete overview of SAP BPC solutions
- Master the features, functions, and integration of SAP BPC with other components and applications
- Updated and revised for SAP BPC 10.0

Sridhar Srinivasan
Kumar Srinivasan

Galileo Press

SAP PRESS is a joint initiative of SAP and Galileo Press. The know-how offered by SAP specialists combined with the expertise of the Galileo Press publishing house offers the reader expert books in the field. SAP PRESS features first-hand information and expert advice, and provides useful skills for professional decision-making.

SAP PRESS offers a variety of books on technical and business-related topics for the SAP user. For further information, please visit our website: http://www.sap-press.com.

Peter Jones
Implementing SAP Business Planning and Consolidation
2013, 650 pp., hardcover
ISBN 978-1-59229-422-0

Rogerio Faleiros and Alison Kreis Ryan
Configuring Controlling in SAP ERP
2012, 547 pp., hardcover
ISBN 978-1-59229-401-5

Marco Sisfontes-Monge
Controlling-Profitability Analysis with SAP (2nd Edition)
2012, 550 pp., hardcover
ISBN 978-1-59229-386-5

John Jordan
Product Cost Controlling with SAP (2nd Edition)
2012, 652 pp., hardcover
ISBN 978-1-59229-399-5

Sridhar Srinivasan and Kumar Srinivasan

SAP® Business Planning and Consolidation

Galileo Press

Bonn • Boston

Dear Reader,

For the last several years, Sridhar and Kumar Srinivasan's book on SAP Business Planning and Consolidation has guided consultants, implementation and management teams, and corporate and functional finance teams. I'm pleased to announce that these experienced SAP PRESS authors have done it again: If you're looking for an essential guide to the latest release of SAP BPC, you've opened the right book.

In this updated third edition, Sridhar and Kumar offer you the industry know-how and insight that will bring your understanding of this indispensable business tool up to speed. Let your implementation of SAP BPC 10.0 benefit from their expertise as they guide you through the newest features and functionality for the NetWeaver version. They consistently provide important context for easy-to-follow, step-by-step workflows, putting a practical understanding of this key business tool well within your grasp.

We appreciate your business and value your feedback. Incorporating reader input is an integral part of updating and revising best-selling books like this one, so I hope you'll visit www.sap-press.com and share your thoughts about this third edition.

Thank you for purchasing a book from SAP PRESS!

Emily Nicholls
Editor, SAP PRESS

Galileo Press
Boston, MA

emily.nicholls@galileo-press.com
http://www.sap-press.com

Notes on Usage

Imprint

This e-book is a publication many contributed to, specifically:

Editor Emily Nicholls
Acquisitions Editor Katy Spencer
Copyeditor Julie McNamee
Cover Design Graham Geary
Photo Credit iStockphoto.com/cgering
Production E-Book Graham Geary
Typesetting E-Book Publishers' Design and Production Services, Inc.

We hope that you liked this e-book. Please share your feedback with us and read the Service Pages to find out how to contact us.

The Library of Congress has cataloged the printed edition as follows:
Srinivasan, Kumar.
[SAP BusinessObjects planning and consolidation.]
SAP business planning and consolidation / Kumar Srinivasan and Sridhar Srinivasan. — 3 Edition.
pages cm
Revised edition of SAP BusinessObjects planning and consolidation.
ISBN-13: 978-1-59229-445-9
ISBN-10: 1-59229-445-6
1. BusinessObjects. 2. Business planning. 3. Strategic planning. I. Srinivasan, Sridhar.
II. Srinivasan, Sridhar. SAP BusinessObjects planning and consolidation. III. Title.
HD30.28S65 2012
658.4'01028553—dc23
2012040862

ISBN 978-1-59229-445-9 (print)
ISBN 978-1-59229-639-2 (e-book)
ISBN 978-1-59229-640-8 (print and e-book)

3rd edition 2013

Contents

Foreword

Solutions around planning, budgeting, forecasting, and consolidation are not new to SAP. After all, the "P" in ERP stands for Planning, which is SAP's forte. A little more than 10 years ago, SAP launched the Strategic Enterprise Management (SEM) suite of products. This included Business Planning and Simulation (BPS), Business Consolidation (BCS), and a number of other solutions under the SEM umbrella. These solutions were installed as an add-on to the Business Warehouse (BW), which is SAP's data warehouse offering.

Fortune 500 companies and other large customers that had an SAP R/3 footprint were the typical consumers of SEM applications. SEM was primarily designed to be owned and maintained by an IT audience while Finance, as the end user, provided some support in the form of power users. BPS and BCS were separate products and customers were able to integrate the data, but this was not an out of-the-box process. Both products were also shipped about six months after a BW release, so customers who had an embedded solution were dependent on the availability of an SEM release in order to upgrade their BW system. Some customers installed standalone systems and moved data between them to provide more flexibility, but this came with a higher total cost of ownership (TCO). They also had to support separate security models, reporting tools, metadata, and so on.

SAP worked closely with various user groups (including the Americas SAP User Group, called ASUG) to gather requirements and feedback for the future development of planning and consolidation solutions. Customers requested tighter integration with BW and the ability to leverage the same tools, which would reduce implementation time and effort to train end users. In 2004, SAP decided to create a next-generation planning solution called Integrated Planning (BW-IP), which would be the successor to BPS. The solution combined the analytic (OLAP) and planning engines into one, and utilized the same queries, variables, security, etc., as BW. It was designed to be a planning platform upon which other product areas within SAP—as well as ISVs, partners, and customers—could build their own planning applications.

On the surface, BW-IP looked like it would be a strong solution, and there was strong demand from customers and partners to get their hands on the beta version (ramp-up). However impressive, the technical advantages of BW-IP only solved part of the equation; the reality was that many businesses were growing tired of IT departments having to own and maintain planning and consolidation solutions, and they demanded solutions that the business could own and maintain with IT support, not ownership.

In early 2007, SAP formed a new organization focused on the business user. This was a big departure from the traditional transaction-oriented and task worker-based applications that SAP was traditionally known for. So an Enterprise Performance Management (EPM) team was launched under this area to address the needs being expressed by the CFO suite. Strategy Management was the first application in the portfolio and resulted from SAP's acquisition of Pilot. It was a very simple and easy-to-use solution for the business to leverage in defining strategy, initiatives, and results monitoring.

In parallel, the EPM development team created planning prototypes on top of the BW-IP platform. The challenges they found were centered mostly on the constraints of Business Explorer (BEx) for Planning and Reporting. Since its first release in 1997, BEx had moved far from its roots as a power user tool to be more open to the needs of a casual user, but it had not come far enough to deliver a world-class user experience. Although the interfaces were Excel- and web-based, the desired ease-of-use was difficult to achieve at the level customers expected and SAP strived for. So the EPM team evaluated potential acquisitions that had the desired ease-of-use and simplicity, and discovered them with OutlookSoft. The user environment was native Excel, which Finance users live and breathe by, but it also had another huge advantage: The founders of OutlookSoft came from Hyperion and had endeavored to build a product that integrated planning and consolidation into one. This was a first in the industry and provided one solution to the customer. While the concept seemed promising, would OutlookSoft solve the problems our customers were telling us were important in their planning and consolidation decision making processes?

Meanwhile, feedback from a BW-IP early adopter indicated that although BW-IP was a big improvement over BPS, the customer decided to purchase the OutlookSoft product so the business could own the solution. This customer validation of the strength of the OutlookSoft solution was important, and prompted internal

stakeholders at SAP to divine some proof that OutlookSoft was as strong as the business case stated, and that it could be implemented as easily as claimed.

Members of OutlookSoft and SAP were summoned to SAP headquarters in Walldorf. A business scenario was created and two teams were formed. The first was from SAP and included BW experts, ABAP developers, web application development members, and finance personnel. In the other corner was the OutlookSoft team made up of a couple finance users with no data warehousing or development skills. The two teams split up and started building their planning applications using the two different solutions, BW-IP and OutlookSoft. After thirty minutes, the OutlookSoft team came back with a completed application; it took the SAP team over six hours to build the same application using BW-IP. This sealed the deal within SAP and the acquisition of OutlookSoft gave SAP the best user experience in the industry, planning and consolidation in one product, and a solution that is much faster and easier to implement. This was welcome news to our customers, as you might imagine.

Prior to the acquisition, there was only about a 15% overlap between OutlookSoft and SAP customers. The OutlookSoft product was a Microsoft-centric application that leveraged SQL Server, Analysis Services, Reporting Services, and other Microsoft technologies. In order to give customers a choice, part of the agreement to acquire OutlookSoft was to offer the solution in another flavor—as part of SAP NetWeaver. There are over 16,000 BW customers and if this solution were to eventually become the strategic planning and consolidation solution at SAP, then it needed to have the tight integration not only with BW but also ERP. The development timelines were aggressive; the first BW-based version was due in one year. What is now known as SAP Business Planning and Consolidation for SAP NetWeaver was completed in July 2008 as planned, and the first beta customers started their implementations in August. The rest is history.

The adoption of the product has been nothing short of impressive, and SAP BPC is one of most demonstrated products to customers and top sellers. In addition, one of the most compelling features of SAP BPC 7.5 is the integration with the SAP BusinessObjects Business Intelligence (BI) tools. This allows customers to leverage the world-class features of the top BI tools in the market with not only SAP BPC but also the entire EPM suite, and provides one comprehensive set of reporting, query, analysis, and information management tools.

The second version (SAP BPC 7.5 for NetWeaver) entered ramp-up in December 2009, and prior to the start, the customer pipeline was two times greater than a

key performance indicator projected. As of February 2010, we achieved our Key Performance Index, with many more customers in the pipeline, making this one of the most sought-after products in SAP history. The product became generally available in August 2010, and finished ramp-up with three times the number of customers expected. This comes with another set of challenges: supporting a product that is in such high demand requires all resources—including development, solution management, field services, and partners—to take an active role. It's a nice problem to have.

The third version (SAP BPC 10.0 for NetWeaver) entered ramp-up in May 2011 and became generally available in April 2012; it further extends the market-leading capabilities by introducing a new state-of-the-art EPM Office add-in (Excel, PowerPoint, and Word) and web user experience, and productized integration with BI 4.0 and GRC 10. It also takes advantage of SAP HANA for in-memory processing. SAP BPC on HANA, as its known, enables real-time analysis and planning, effectively allowing customers to react faster to volatile economic conditions, dynamic planning, and forecasting to reduce planning cycle time and improve forecast accuracy. Our customers can make better investment decisions and use integrated business planning to facilitate cross-organizationally aligned planning processes.

The analyst community has also overwhelmingly validated SAP's vision, strategy, and execution. Gartner rated SAP as a leader in the CPM Suites Magic Quadrant for 2011/2012 and designated it as the strongest vision in the market for the fourth year in a row. The 2009 Forrester Business Performance Solutions Wave Report not only listed SAP as one of four leaders in the market, but it also awarded SAP with the highest score for current product offering out of all vendors. IDC's 2011 annual analysis of the software markets reported SAP as the market leader in performance management and analytic application with a 26.1 percent market share and the highest revenue of that year. The Ventana Research Value Index 2011 for Financial Performance Management rated SAP as a "hot" vendor; we were awarded the highest combined ranking across all categories for the second year in a row.

The authors of this book have strong SAP backgrounds—specifically with BW-IP and BW—and were among the early implementers of the SAP BPC for NetWeaver product. Kumar and Sridhar Srinivasan are able to leverage their deep expertise in planning, consolidation, and data warehousing to compare and contrast the SAP solutions and provide in-depth knowledge of the SAP BPC for NetWeaver product. This book should serve as an excellent reference and implementation and information source for customers and partners. Whether you are sitting in an airport

terminal and have some time to catch up on reading or have a quick question while visiting a customer's site, I hope that you enjoy this book. It is something that you can utilize over and over whether you are an executive or a member of a project team implementing SAP BPC for NetWeaver. We hope SAP BPC for NetWeaver is a huge success for you now and in the future!

Bryan Katis
Group Vice President and General Manager
Enterprise Performance Management Solutions
SAP Analytics

Acknowledgments

This book has been produced with the help of several people who have reviewed the materials and offered invaluable suggestions. Their contributions and advice have helped in the realization of this edition of the book.

We would like to offer our thanks and special appreciation to the following people for their contributions:

- The acquisitions editor of this book, Katy Spencer, for her assistance through the proposal and approval process and for getting us started with this third edition.
- Our project manager and developmental editor, Emily Nicholls, for guiding us through book development.
- The copyeditor of this book, Julie McNamee, and production editor, Graham Geary, for managing the production processes.
- Several readers of previous editions of this book, who provided valuable feedback that has been incorporated in this edition.
- Reddy Cheruku at Principle IT, who reviewed the chapter related to consolidation and provided feedback.
- Gilbert Guymer at Lufkin Industries, who helped edit the first chapter of the book.

We would like to thank our families for providing us the motivation and enthusiasm and for their support and patience as we spent long hours writing the book.

Sridhar Srinivasan and **Kumar Srinivasan**

Additional Information

Additional information on the topics covered in this book can be found in the following places:

- SAP Help Portal: *http://help.sap.com*
- SAP Developer Network: *http://www.sdn.sap.com*
- SAP Service Marketplace: *http://www.service.sap.com*

Introduction

Planning and financial consolidation are two areas that every organization implements to ensure the smooth functioning of its business. The main objective of planning is thinking ahead and coming up with the activities a company will do in the future. Consolidation is a process that is used to accurately report the financial results of an organization either for legal or management reporting purposes.

Every organization, big or small, has to plan for it to be successful in meeting its objectives and to stay competitive in the market place. A good planning system can help the organization to think ahead and come up with the right set of activities to get ahead of its competition. It is not uncommon to see how an organization in a good business has failed due to a lack of or inadequate planning. On the other hand, we often see how an organization with a good planning system has surpassed its competition and become immensely successful in its history.

Gone are the days where you could be in a good business and expect to stay successful forever. Now, competition is rampant and technology is changing rapidly. Although a business may not be directly related to technology, the effect of technology and innovation in any industry makes it necessary to have a clear plan to address these changes and developments. Planning is of the utmost importance for meeting these needs effectively.

Planning affects every area and sphere of a business. It is relevant for different activities, including production planning, profitability planning, human resources planning, and financial planning. In addition, there are different timelines to planning that correlate to the type of plan that needs to be adopted. Planning can be strategic planning for the long term or could be operational planning for the short term. In essence, a good planning system can help an organization spot opportunities, trends, and new ideas in the industry where it is operating and come up with the right kind of plan to be successful in its activities.

On the other hand, organizations report consolidated financial results in a particular format based on the country/countries where it operates. It is important to have a

sound system that can report the results of the organization as a whole in a truthful manner and, at the same time, employ the best practices of accounting.

The process of producing consolidated results involves the following distinct tasks:

- Gather financial data of all entities
- Translate data in local currency to one or more reporting currencies
- Run business rules to perform calculations/validation (e.g., cash-flow)
- Reconcile intercompany transactions
- Eliminate intercompany transactions
- Consolidation of investments
- Post top-side entries using journals
- Report the consolidated results

These steps require setting up of the entity structure of the organization, account structure, ownership tree, exchange rates, data model, and business rules to perform the tasks. An efficient system will reduce time to perform financial closing and make the process more reliable by providing accurate results.

Software Tools

It is in the context of each organization's need to meet the requirements of planning and financial consolidation that the opportunities of software planning tools assume importance.

Many organizations use Microsoft Excel spreadsheets to maintain data related to planning and to prepare the financial results. Excel spreadsheets are excellent end-user tools for planning and consolidation. However, they are not robust in terms of security and don't have the level of control and functionality required to manage and maintain sensitive data related to planning and consolidation.

SAP Business Planning and Consolidation (SAP BPC) for NetWeaver is an add-on tool available in the SAP NetWeaver Business Warehouse (BW) system that can address the requirements of planning and financial consolidation of a business. SAP BPC provides the familiar Microsoft Excel front-end interface for users to configure and perform planning and consolidation. At the same time, it addresses the limitations of using Microsoft Excel as the primary tools for these requirements

by securing the data and by providing robust functionality to perform planning and consolidation tasks. The tools available in SAP BPC help an organization to configure the application that is right for its business and at the same time provide the best practices for application development and maintenance. The application provides out-of-the box tools to perform planning and consolidation tasks. These tools can be further customized for an organization via business rules. Financial data extracted from one or more source systems into the SAP NetWeaver BW system can be used as source data to feed into a planning or financial consolidation application in the SAP BPC system.

With all these advantages, SAP BPC is a sophistical tool that reduces the total cost of ownership (TCO) for an organization to develop and maintain applications. SAP BPC provides for reduced planning and consolidation cycle times, accelerated and automated financial closing process, and reduced compliance risk with accurate reporting that meets statutory requirements.

Using the SAP BPC system will not only enable an organization to perform planning and consolidation, but will also enable the organization to monitor the whole process and ensure delivery in a timely manner. The business process flow functionality that can be enabled in SAP BPC would allow the organization to track the status of where an organization stands in the planning or consolidation process. The ability to integrate the actual data in the SAP NetWeaver BW system with data in SAP BPC makes it a compelling reason for companies to use this software as the application of choice.

How This Book Can Help

The objective of this book is to help technical developers, functional analysts, consultants, and managers who work in the area of planning and financial consolidation understand and make best use of the capabilities of the SAP Businesss Planning and Consolidation for NetWeaver tool. This book is unique in that it explains the features of the software through extensive examples in simple and easy-to-understand manner.

Structure of the Book

This book contains 10 chapters. Let's take a glimpse at the contents of each chapter:

- **Chapter 1: Overview of Enterprise Performance Management**
  This chapter provides an overview of enterprise performance management (EPM) and introduces you to concepts in planning, budgeting, forecasting, and consolidation.
- **Chapter 2: Overview of SAP BPC**
  This chapter provides an overview of the SAP Businesss Planning and Consolidation software. It discusses architecture of SAP BPC on a NetWeaver platform and discusses objects created on the SAP NetWeaver BW and SAP BPC systems.
- **Chapter 3: Designing a Model with SAP BPC for NetWeaver**
  The third chapter will introduce you to a case study of a model company that has decided to plan its gross margin by implementing the NetWeaver version of SAP BPC. This chapter will explain the steps involved in modeling objects in SAP NetWeaver BW and SAP BPC systems.
- **Chapter 4: Loading, Scheduling, and Managing Data in SAP BPC for NetWeaver**
  In this chapter, we will discuss how to load data in the SAP NetWeaver BW and SAP BPC environments.
- **Chapter 5: Reporting, Planning and Analysis in SAP BPC**
  The fifth chapter will discuss how to report, enter data using input templates, and analyze data using the SAP Business Planning and Consolidation software, while describing the various reporting and analysis options available in the tool.
- **Chapter 6: Developing Business Logic in SAP BPC**
  In the sixth chapter, we will discuss different options available in SAP Business Planning and Consolidation for defining logic and automating the process of deriving data and enforcing business rules. We will discuss dimension formulas, logic script, high-level usage of business rules, and usage of BAdI to define business logic. We will describe how currency conversion and allocation are handled in SAP BPC. We will also discuss business rules in detail in Chapter 9 when discussing tasks related to consolidation.
- **Chapter 7: Process Management and Collaboration**
  This chapter introduces you to the collaboration tools and features available in SAP Business Planning and Consolidation that facilitate the sharing and exchange of data. It discusses configuring work status that is used to monitor and restrict changes to data in an SAP BPC application, adding and viewing comments, distribution of input schedules to offline users, and publishing books that can facilitate the planning and consolidation process.

- **Chapter 8: Essential Tools for Building Models**
  In this chapter we cover a range of topics that are important to understand when implementing a planning or consolidation application using the SAP BPC software, including the following:
  - Transport: The objects developed in the SAP BPC development environment have to be transported to the quality and production environments. This topic explains in detail the process of transporting objects.
  - Locking: This topic involves describes the locking mechanism used in SAP BPC to prevent two users from updating the same data at the same time.
  - Administration parameters: Parameters are used to influence the behavior of applications. Three types of parameters can be configured in BPC—functional parameters, configuration parameters, and system parameters. We'll discuss the different parameters settings that are available and how those settings impact the application.
  - Statistics: SAP BPC provides the tools to measure statistics and to gather information about how the system is used. Using these statistics, you can identify bottlenecks and take timely action before they become major issues. This topic describes how to collect statistics for an SAP BPC application.
  - Audit: When sensitive data resides in an application and when changes to that data are to be monitored, auditing assumes importance. This topic describes how to configure and use the audit functionality in SAP BPC.
  - Documents: In this section, we'll discuss how to use the content management tool in SAP BPC to store, manage, and enforce the security of unstructured data
  - Security: This topic lists the different objects you can configure to limit user access to data and tasks inside an SAP BPC application. The steps of setting up users and teams are discussed.
- **Chapter 9: Consolidation with SAP BPC**
  The ninth chapter discusses in detail the process of how to consolidate data in SAP BPC. The first part of the chapter will introduce you to the steps involved in the consolidation process. We will discuss topics related to prepare, collect, and consolidate steps and detail usage of different types of business rules that can be configured in SAP Business Planning and Consolidation. Business rules allows users to set up standard business processes such as carry forward balances, account transformation, intercompany booking, intercompany

elimination, and controls without having to develop code. We will discuss the need to perform matching of intercompany transactions and steps to match intercompany transactions. We will discuss intercompany elimination using US elimination and automatic adjustment business rules. We will describe scenarios to illustrate consolidation of investments using different consolidation methods. Finally, we will discuss the use of journals to create top-side entries.

- **Chapter 10: Business Process Flow and Enhancements**
  In the tenth chapter, we discuss how to use the Business Process Flow feature to organize the execution of a business process and to monitor the status of planning or consolidation processes in the SAP BPC system. This final chapter highlights some of the new features and enhancements introduced by SAP in recent support packs.

Summary

The introduction and overview provided in this chapter explain the key components and use of this book. In Chapter 1, we will provide an overview of Enterprise Performance Management and discuss concepts related to planning, budgeting, forecasting, and financial consolidation.

SAP Business Planning and Consolidation software is a component of the SAP Enterprise Performance Management (SAP EPM) suite of products. This chapter introduces different SAP EPM components and outlines basic concepts for understanding planning and consolidation applications.

1 Overview of SAP Enterprise Performance Management

In Section 1.1, we'll discuss the definition and importance of SAP Enterprise Performance Management (EPM) and its relevance in different areas of business. In Section 1.2, we'll discuss concepts in planning, budgeting, and forecasting, including methods and best practices in these areas. In Section 1.3, we'll discuss basic concepts and processes involved in consolidation.

1.1 Enterprise Performance Management

The importance of EPM is growing in this era of rapid technology development, global competition, modernization, and ever-changing customer needs. The need to make accurate decisions—and, at the same time, meet regulatory requirements—is important to gaining a competitive edge, ensuring control of business operations, and providing precise business status information to statutory bodies, shareholders, and business partners.

EPM deals with bridging the gap between the strategies set by top management and the execution of the strategy by operational staff. It streamlines the execution of strategy and ensures accountability. Its objective is also to provide timely and compliant reporting and to help companies optimize costs and improve profitability. EPM provides the framework for establishing processes that are needed to meet strategic objectives and for monitoring the execution of these processes.

The following are the broad themes for SAP EPM tools, according to SAP:

- Harmonize: This theme provides the same user experience, same languages, same visualization, and same platforms.
- Connect: The objective of this theme is to provide the ability for software to connect to SAP BusinessObjects BI solutions, SAP ERP, SAP NetWeaver, SAP BusinessObjects GRC solutions, and other SAP BusinessObjects EPM applications while remaining open and heterogeneous to non-SAP sources and technologies.
- Extend: This theme extends the products with innovative customer enhancements so that they remain best-in-class solutions.

EPM solutions from SAP provide the following software tools:

- **SAP Strategy Management**
  This application allows you to prioritize and communicate strategic initiatives and effectively monitor, measure, and collaborate on strategy.
- **SAP Business Planning and Consolidation**
  This application helps you to perform planning, budgeting, and forecasting tasks, along with financial consolidation and reporting. The objective of this book is to discuss the functionality available in this application in detail.

  The evolution of what is now SAP BPC began in 2007 with SAP's acquisition of OutlookSoft. One of the strategic decisions behind this acquisition was to leverage the rich features of OutlookSoft and integrate it with the SAP NetWeaver platform to provide a competitive edge to customers to meet their requirements for the planning and consolidation applications. At the same time, SAP announced that it was committed to preserving the investment made by existing OutlookSoft customers who were running on Microsoft SQL servers. This commitment was followed up with a dual release plan for SAP NetWeaver-based solutions and Microsoft-based solutions for every major release of SAP BPC.

  The first major integration timeline of the software with SAP NetWeaver was the release of SAP BusinessObjects Planning and Consolidation for NetWeaver 7.0 version in April 2009. This release enabled SAP BPC customers to leverage the strengths of the SAP NetWeaver environment and to use data in the SAP NetWeaver Business Warehouse (BW) system for building, planning, and consolidation applications. At the same time, SAP released a version for Microsoft with a number of improvements over the OutlookSoft 5.0 version. The SAP

BusinessObjects Planning and Consolidation 7.0 version for NetWeaver provided a sound infrastructure and foundation for developing planning and consolidation applications. This version was able to replicate and provide most of the features available in the Microsoft solution of the product. The release of SAP BusinessObjects Planning and Consolidation for NetWeaver 7.5 version and Microsoft 7.5 version in August 2010 was an enhancement to the 7.0 version and brought SAP's strengths into the fore, making this a compelling software offering that every customer could benefit from by using it for planning and/or statutory consolidation requirements.

The new SAP BPC 10.0 for NetWeaver and Microsoft takes the return on investment (ROI) objective of customers further by providing the following benefits:

- Zero-based footprint in the use of software
- Unifying solutions from various EPM-based applications
- Top-class features for implementing planning and consolidation applications

SAP is continuously investing time and effort to confirm that its EPM applications are in tune with the changing needs of businesses and to provide the best returns for the business.

- **SAP Financial Consolidation**
  This application allows companies to perform financial consolidation and reporting.
- **SAP XBRL Publishing**
  The eXtensible Business Reporting Language (XBRL) is a new global standard for exchanging financial information that allows accounting jurisdictions such as banks, regulators, and agencies to download, codify, and analyze financial data. The XBRL Publishing application by UBmatrix allows you to create XBRL documents based on data available in SAP BPC, SAP Financial Consolidation, and SAP Business Suite software.
- **SAP Financial Information Management**
  This application lets you access, map, and load data from different source systems to the SAP Financial Consolidation and SAP Profitability and Cost Management applications.
- **SAP Intercompany**
  This application lets you reconcile intercompany balances in real-time via the web, enabling your organization to close its books more quickly.

- **SAP Profitability and Cost Management**
  This application allows you to accurately and effectively measure product, customer, and channel profitability, along with developing and dynamically testing ways to improve profitability.
- **SAP Spend Performance Management**
  This application provides you with insight into savings opportunities and compliance by enabling access to aggregated and enriched spend data.
- **SAP Supply Chain Performance Management**
  This application helps you improve the effectiveness of supply chain operations such as lowering costs and improving return on working capital. It also provides an accurate measure of whether you are meeting your supply chain goals, warns you of potential bottlenecks, and alerts you to new opportunities.
- **SAP Disclosure Management**
  This application is designed to manage the publication of financial statements and reports to help organizations with XBRL and IFRS reporting compliance. It can easily integrate with other SAP systems and serve as a central point to manage complex financial processes.
- **SAP Sustainability Performance Management**
  Sustainable development is geared toward balancing human needs with the ecosystem. To this end, organizations are required to comply with government regulations and provide information on their operations. This application provides the tools to maintain and report organizational operations data with a view to comply with the goal of sustainable development.

The roadmap for EPM offerings in SAP's vision is shown in Figure 1.1.

SAP is a leader in the enterprise resource planning software arena, which helps reengineer business processes and execute them efficiently, and its solution in the area of EPM complements its existing software product offerings. SAP's offerings in other areas—such as business intelligence, supplier relationship management, supply chain management, customer relationship management, and industry-focused solutions—help accelerate innovation and improve ROI. Continuing this trend, SAP's acquisition of Business Objects in October 2007 ensures its ongoing leadership and its ability to help you devise effective business strategies.

Figure 1.2 shows how SAP software solutions continually integrate insight, strategy, and decisions across governance, risk, and compliance (GRC), EPM, and business intelligence platforms to execute tasks efficiently and effectively, easily implement

new processes, and monitor their execution. It also shows how these solutions bridge into business applications that can run on any technology platform. This closed-loop business performance optimization allows companies to obtain a complete picture of their business performance.

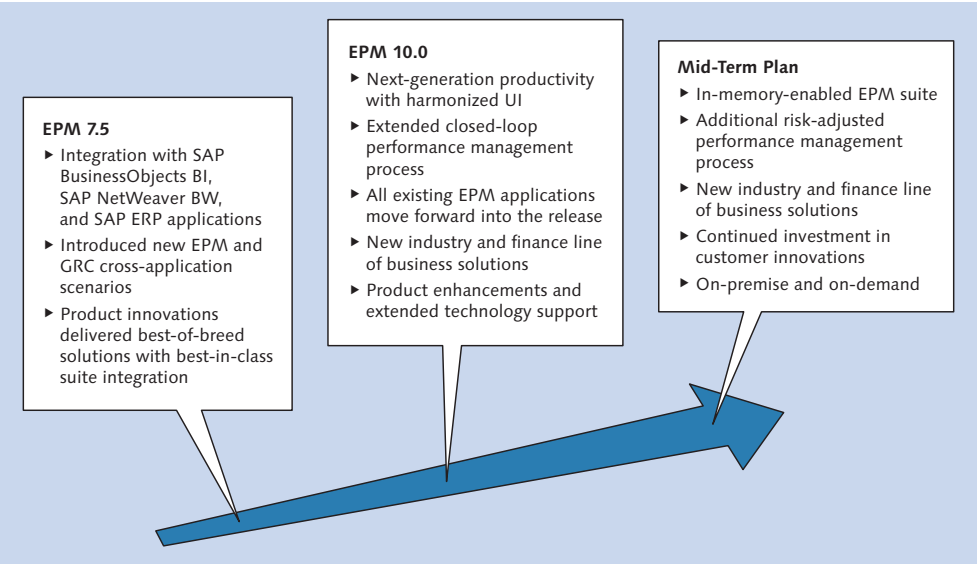


Figure 1.1 EPM Summary Roadmap

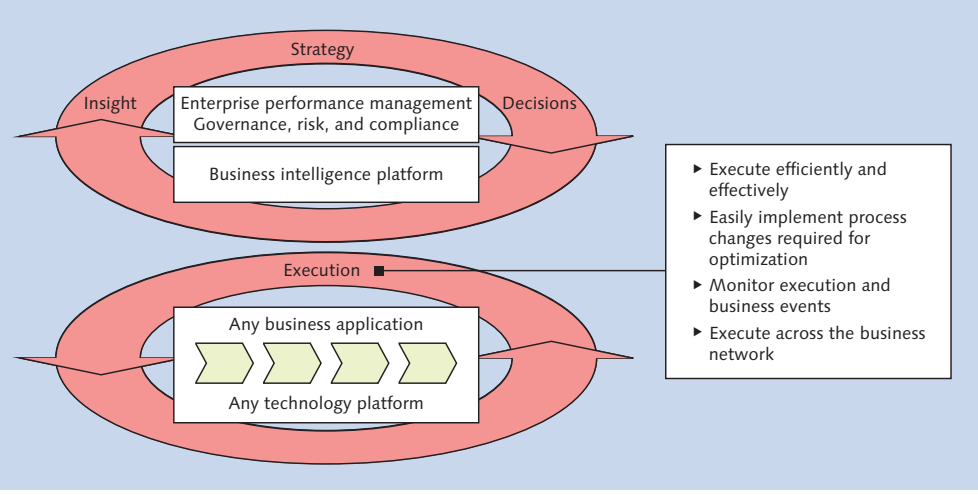


Figure 1.2 Closed-Loop Business Performance Optimization

In the next section, we'll review basic concepts used in planning, budgeting, and forecasting.

1.2 Planning, Budgeting, and Forecasting

In simple terms, *planning* involves thinking ahead and formulating a set of activities you'll execute in the future; in other words, it is the process of modeling or projecting future business activities. Companies strive for profitability and growth in complex business environments where they have to deal with changing scenarios related to competition, technology, geopolitical situations, and new business needs so planning is a key management cycle component that allows companies to position themselves in these complex environments.

The process of planning identifies the individual tasks involved in reaching a goal, constructs the time frame for executing the tasks, and determines the resources needed for successful completion of the tasks. It's an iterative activity where actual performance is measured against set targets and refined accordingly. To ensure its continuance, a business has to plan for multiple resources and to confirm that these activities are undertaken in an optimal manner. A good idea or a good product doesn't necessarily contribute to a company's success; the company must make decisions to sell the right product at the right time through the right channel, and more importantly, increase revenue and reduce costs in that process. To best achieve the company's goals, the plan must be clearly defined and refined during the planning process.

Although strategic planning objectives are set by top management, a company may formulate and carry out multiple plans at lower levels to achieve the targets set at the higher level. So, it is imperative for the company to ensure complete coordination and unity of objectives among the different plans. This should be taken into account when planning decisions are made.

Note

Many organizations have used Microsoft Excel spreadsheets for financial planning, but their limitations are many: accessibility issues, lack of security, insufficient control mechanisms on who can modify data, and an inability to clearly understand how the planning data was derived. SAP software can help in this area, so we will come back to this topic in Chapter 2.

When discussing EPM, it's important to understand the differences among planning, budgeting, and forecasting.

As discussed earlier, *planning* is the process of modeling and projecting future business resources. Depending on the type of planning, the planning horizon can be short-, medium-, or long-term.

Budgeting is the process of allocating resources after the company's management has approved and accepted the planning process. It's the start of the action phase that follows the planning process. The budgeting process, in which requisite details are worked out for the implementation of the plan, is usually executed before the start of the budget year. This is also the period where financial allocations are made to various departments such as finance, sales, information technology, human resources, and so on.

Forecasting deals with the realization of the plan and is used as a monitoring mechanism to facilitate the success of planning. Business environments do not remain the same—what was planned for yesterday may not be the same today. Forecasting is done during the course of the current budget year with a key objective to provide visibility on the current state of the business in a timely manner so that corrective action can be taken when there is a significant difference between the current state of the future and what was planned for during the planning process.

In the subsections that follow, we'll discuss several aspects of the planning process: the planning horizon, planning types, planning areas, common scenarios for planning, and some of the important business elements you should take into account when planning.

1.2.1 Planning Horizon

The planning horizon determines the time frame for planning. As mentioned earlier, planning horizons are divided into short-, medium-, and long-term planning, which we'll discuss in more detail.

Short-Term Planning

The planning time frame in short-term planning is usually a year or less. You use this type of planning when the business has clear-cut short-term goals, such as cutting costs, increasing labor productivity by freezing new hires, and effectively training the workforce. Short-term planning is also suitable in situations where the

industry in which the business is operating is constantly changing and where it's difficult to make reliable long-term projections.

Medium-Term Planning

Medium-term planning usually covers a period of one to three years and is applicable if the business can reasonably plan the outlook for this time period. You might develop a medium-term plan to increase market share in a particular segment of the business, for example.

Long-Term Planning

Planning is considered long term when the duration of the plan exceeds three years. This type of planning usually involves the investment of a large amount of capital to achieve company objectives. This is also applicable for businesses where projects have a long gestation period. In these types of industries, there is a long time gap between initial investment and final realization of sales and profits; this is true for companies in the utility, steel, and biotechnology industries, for example.

It's important to understand that the level of detail in short-term planning is high because it represents the immediate future. In medium-term planning, the level of detail is reduced. In long-term planning, the level of detail again is very high. Although you can use different time frames for planning, there is no single formula to decide on the option to be used; instead, the period selected depends on the business requirements.

In reality, you may use different terms for different areas of your business. In general, routine business operation plans that address the immediate future are good candidates for short-term planning. Programs and plans that involve large capital investment and that take more time to mature and yield results are categorized as long-term plans. For example, your business may come up with a short-term plan to reduce costs in the immediate future, but also have a long-term plan in place to gain the highest market share in your industry.

1.2.2 Planning Types

The planning type, which is based on the planning horizon and the granularity of the planning process, provides more clarity to the process by clearly differentiating the objective of the plan. We'll discuss each of the planning types next.

Operative Planning

Operative planning is typically used for the short term and is generally conducted at a company's operational level. This type of planning usually has an immediate objective in mind (e.g., to improve productivity by controlling variable costs). Operative plans rely on a bottom-up planning approach, in which the planners involved in executing the plan are also involved in the creation of the plan. Employees at the bottom level of the hierarchy may be part of the planning process to make it successful. Upon completion of the plan, they send the plan to their supervisors, and after approval by the supervisors, plan implementation begins.

Bottom-up planning processes are very helpful for achieving employee commitment to the planning process. Involving employees in the planning process increases the probability that the plan will be successful.

Tactical Planning

Tactical planning is associated with planning for the medium term and is generally conducted at a company's division level. It's suitable for projects that can be completed in fewer than three years. For example, you might start an initiative to improve the quality of a product, which may take two years to realize. Although the tactical plan may be in consonance with the long-term plan, it may be developed at the middle management level.

Strategic Planning

Strategic planning is associated with planning for the long term and is generally conducted at the organizational level. Company management usually plays a major role in this type of planning. Let's say a company decides to finance research and development to develop a new line of products. The products may take more than three years to develop, so the company may want to develop a plan to finance the new program using strategic planning.

This top-down planning process usually begins at the company's top level and trickles down to lower levels. The company's top management agrees to the planning objective and process, which may be split into subplans during the implementation process.

Figure 1.3 shows how the different types of planning are categorized based on the *granularity* of the plan (high level versus detailed level) and the *gestation period* (time) to achieve the plan objectives. Strategic planning has the longest gestation

period and the lowest level of granularity. Operational planning has the shortest gestation period and the highest level of granularity.

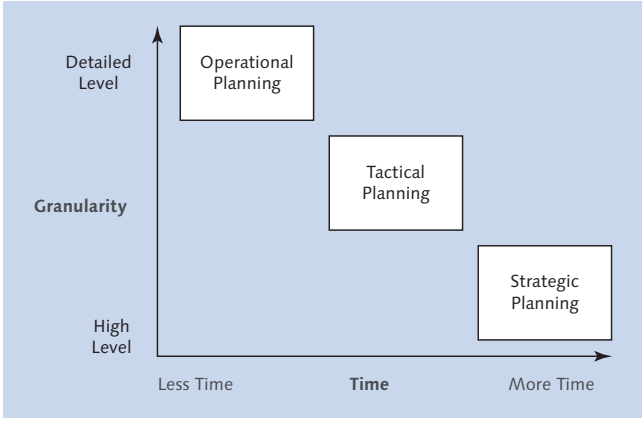


Figure 1.3 Planning Types

1.2.3 Planning Areas

Planning is a generic function applicable to all areas of business. The planning area is the subarea of the business for which you develop a plan. Next, we'll discuss how to understand the relevance of financial planning in several subareas of business.

Finance

The finance department is the control room of any business; it's responsible for planning, implementing, and directing the financial aspects of the business. Examples of financial planning include investing in a new venture, maintaining an optimal cash position (liquidity), reducing costs, and increasing profits. The focus of this book is to illustrate the development of a financial planning application for a company that has an objective to increase its gross profit margin.

In this context, it may help to discuss certain areas of financial planning:

- **Liquidity planning**
  Liquidity planning focuses on planning for maintaining optimal cash flow to be able to run the business. It involves taking the company's current cash position into account, estimating the inflow of cash from customers and the outflow of

cash to vendors and employees, and planning the liquidity over the next few months.

- **Cost center planning**
  Cost center planning is done for operational, nonoperational, and capital expenditure costs the company will incur at various cost center levels.
- **Asset planning**
  Asset planning is concerned with planning for the purchase of new assets and the disposal and maintenance of existing assets, based on the company's objective to deliver products to its customers.
- **Profitability planning**
  Profitability planning determines profitability by identifying sales revenue and costs that will be incurred in the future.

Production

You should plan how much to produce and when and where to produce it, with the goal of producing products in the most optimal manner while satisfying customer requirements and maintaining sufficient capacity to fulfill demand. You should take into account variables such as seasonal behavior, geographical demand, and user behavior when planning for this area. Keep the financial implications of your production plan consistent with the company's overall financial plans, and keep the inventory costs and overhead costs incurred as a result of the production plan at optimal levels.

Human Resources

Every business should have the workforce required to run the business, and workforce requirements should be planned based on the company's overall growth objectives. When there is a shortage of employees in the industry, you'll have to make decisions about whether to hire new employees, train existing employees on new technologies, or bring people in from the outside as consultants to meet requirements. This area of planning is especially important in industries that require large numbers of employees and where specialized skills are required for realizing the objectives of the business. Costs associated with meeting labor requirements have a direct impact on the financial plan.

Marketing

The marketing department provides the necessary information to plan the products that can be sold and the price at which they can be sold—in short, a sales plan. (You should also take into account the distribution of products and services to customers.) The sales plan can be formulated based on actual sales in past years, revaluation based on future trends in the industry, the geopolitical environment in which your company has sales operations, and the company's strategic vision.

Maintenance

The maintenance department will typically develop a strategy for plant maintenance according to the equipment strategy and breakdown of the maintenance work into categories. These categories include in-program work, functional failures, and discretionary work. Based on the overall maintenance strategy, the maintenance plan can be developed to cover different types of maintenance activities within the plant such as turnarounds, preventive maintenance, breakdown maintenance, and so on.

Proper maintenance of plant equipment can significantly reduce the overall operational costs and boost the overall productivity of the plant. Costs associated with the maintenance costs have a direct impact on the financial plan.

The previously mentioned areas are critical for the survival of the business and have a direct or indirect effect on the company's financial plan. Planning for these areas is important to ensure the smooth functioning of business.

1.2.4 Common Scenarios for Planning in Business

You should plan for different scenarios depending on individual requirements at different times. This subsection delves into some of the common scenarios companies use for planning. These planning scenarios are based on the company's targeted objectives.

Gross Profit Margin Planning

The revenue obtained through the sale of a company's products, along with the costs that are incurred to produce or buy the product, is planned under this scenario. This usually starts with gathering external information on product demand. Often, prior-year revenue results can indicate a trend for future sales planning. This information,

along with management's strategic vision and demand estimates provided by the sales force, helps the company plan quantity and prices for its products.

You can arrive at expected manufacturing costs using standard costing procedures. This process should take into account any expected increase to material, labor, and overhead costs in the future. You use expected revenue and manufacturing costs to determine the expected gross profit margin.

Profit and Loss and Balance Sheet Planning

This type of planning is related to the financial area of business and refers to the process in which you prepare a profit and loss sheet, along with a balance sheet, to see whether it meets the returns expected by the business. This can also provide clues as to which areas may need corrective action. For example, it may become obvious from the projections that overhead costs are increasing every year; based on this, management may need to take corrective action.

Investment Planning

Investment planning is also associated with the financial aspect of business by focusing on new investments the company is planning in the future. The expected ROI for the investment is a key metric in this planning.

Labor Planning

Labor planning is the process of planning workforce requirements for your company. It should take both new projects and expected expansion into consideration, which may require additions to the workforce. The plan should take into account expected attrition from the workforce, both from retirement and labor turnover. Human capital is crucial to the business, and labor planning is vital to ensure that the company has the right people with the right skills to operate the business.

Sales Planning

With sales planning, you develop a plan to sell the company's products. The focus is on developing a marketing plan that will be most beneficial to the company and help sell the company's products at the optimal price. Recall that the marketing plan should reflect the company's overall strategic vision. For example, if a company decides to place a lot of emphasis on two specific products, the sales plan will

need to reflect this objective. The inputs for developing a sales plan are obtained based on market intelligence and opportunities data collected by sales managers.

Demand Planning

Demand planning presupposes that the process of sales planning is complete. Depending on the sales forecast, production planning is geared to fulfill the expected demand. Assuming that production capacity exists, demand planning uses the existing sales forecast to plan for material, labor, and machinery to produce the products.

1.2.5 Considerations in Planning

You should perform the planning process in a manner that is in tune with your company's strategic objectives. Consider the following key points.

- **Importance**
  Planning is an important component of every business that contributes to the company's success. There are many examples where good planning and execution have paved the way to the success of a business, as well as many examples where inadequate planning or no planning resulted in a company's failure. So management needs to understand the importance of planning in helping the company be successful in its business operations.
- **Skills**
  Planning is an art—and one that requires a lot of foresight. The key to developing a successful plan is the ability to take all of the factors (internal and external) into account, which requires people who have experience, maturity, and knowledge.
- **Internal**
  The person(s) involved in developing a new plan should understand how the new plan relates to other projects that are currently underway, as well as those the company is likely to undertake in the future. Also, you should take cross-functional aspects into account. For example, if you undertake a plan to improve production capacity, you should study the question of financing before getting started with the plan.
- **Data**
  The data that is used in planning should be reliable. For example, if market research and intelligence is used as the basis for planning, the data used for this

purpose should come from a trustworthy source. You must also take into account the possibility of errors in the data.

- **Governance**
  A good governance process should be established for planning. The levels of responsibility should be clearly defined so that there is no ambiguity, and the people involved in planning should clearly understand their role in the process.
- **Communication**
  Good communication among all company levels is of vital importance at all times during planning. This ensures that the various participants stay well informed of the plan's progress and take actions at the different plan implementation stages.
- **Monitoring**
  A sound review process should also be in place to monitor the execution of the planning process. The availability of such a system helps identify any deviations and aids in taking corrective action.
- **Planning method**
  There are two types of planning methods: top-down and bottom-up. With the *top-down planning process*, upper-level management decides what is to be done, and the lower levels of the organization implement the process. This type of process is relevant for strategic decision making, for example, when management decides on which areas of the business the company should focus on over the next 10 years. The *bottom-up planning process* is used more for operational planning, where planning starts at the lower levels of the organization and is approved by upper levels.
- **Participation**
  The planning process is more likely to succeed if people who are responsible for the execution of the planning process are consulted when the planning goals are initially set. This helps in participative decision making and promotes cooperation from everyone involved in implementing the plan. It also provides an opportunity for management to solicit employee feedback regarding the identified planning objectives.
- **Issue list**
  If issues come up during the planning process, they should be recorded so you can track progress toward their resolution. There should also be a good system in place for employees to address these issues.

- **IT department**
  The planning process involves collecting information from different sources. This information may come from cross-functional areas in the company (finance, production, marketing, or human resources), or sometimes from external sources. The IT department facilitates integrating this information, with the goal to build a process for planning that can meet business demands in a reliable and timely manner. The success of this process requires an IT department that can handle this responsibility effectively.
- **Flexibility**
  Avoid rigidity and pursue transparency. The planning process should be flexible so that it can absorb any necessary changes that may occur along the way.

You should now have an understanding of the basic concepts in planning. In the next section, we'll discuss the basic concepts in consolidation.

1.3 Consolidation

The consolidation process is used to accurately report the financial results of an organization either for legal or management reporting purposes. An organization may be composed of a number of entities and subsidiaries, some of which may or may not be fully owned. The financial statements that combine the financial results of all entities and subsidiaries are referred to as *consolidated financial statements*. The consolidated financial statements present the financial results of the operations (income statement), financial position (balance sheet), and cash flows (cash flow statement) of all entities and subsidiaries, as if the group(s) of companies is composed of a single entity. Such consolidated financial statements can also be generated at different levels of the entity hierarchy. (In other words, an organization may consist of several separate entities, but they operate as one centrally controlled economic entity.) Consolidated financial statements provide more useful information to shareholders than do separate financial statements of each entity. Consolidation of revenues, expenses, assets, and liabilities provide a clear picture of the operation and financial status of the consolidated single entity.

The process of producing consolidated results involves the following distinct tasks:

1. Gather financial data of all entities.
2. Translate data in local currency to one or more reporting currencies.

3. Run business rules to perform calculations/validation (e.g., cash flow).
4. Reconcile intercompany transactions (matching).
5. Eliminate intercompany transactions.
6. Consolidate investments.
7. Post top-side entries using journals.
8. Report the consolidated results.

To understand consolidated financial statements, let's walk through a few of these basic concepts.

1.3.1 Elimination of Intercompany Transactions

The consolidation of financial statements involves summing up the amounts across all separate entities and providing a unified view of the operations of the economic entity. But in this consolidation process, it's important to eliminate double-counting resulting from intercompany transactions. The objective of consolidated financial statements is to report the consolidated entity's transactions to outsiders, so if, for example, an entity sells products to another entity within the organization, then the consolidated financial statement should eliminate this sale because it is an intercompany transaction.

You must eliminate two kinds of intercompany transactions:

- **Intercompany sales and cost of goods sold**
  Any intercompany transaction between entities within an organization should be eliminated. Consider a case where an entity sells a product to another entity within an organization, and the second entity then sells the product in the same or a different form to an external customer. A consolidated financial statement of the organization should show transactions of the organization that relate to external companies. The transactions such as sales and cost of goods sold (COGS) that result from the exchange of goods or services within the organization should not be counted, so they should also be eliminated from consolidated statements.
- **Intercompany receivables/payables**
  When an entity sells a product to another entity within the organization, the amount associated with the sale is recorded as accounts receivable in the books of the selling entity. Similarly, the transaction is recorded as accounts payable in the books of the buying entity. This transaction does not result in the

consolidated company receiving or owing any amount to an external customer or supplier, so these transactions should be eliminated from the consolidated statements.

1.3.2 Consolidation of Investments

A corporation may have a number of subsidiaries that may or may not be fully owned. The consolidation of investment deals with elimination of ownership and booking of minority interest of these investments. Three methods are commonly used in consolidation of investments: the purchase, equity, and proportional methods.

The *purchase method* is generally used when the percentage of ownership is greater than 50%. The *equity method* is generally used when the percentage of ownership is less than 50%. The *proportional method* is similar to the purchase method and is generally used in Europe. In the chapters ahead, we'll discuss different methods of ownership and detail how the consolidation of investments is handled in each scenario.

In the purchase method, at the time of purchase, the consolidation of investment process reflects the elimination of ownership that is common between the parent and subsidiary, goodwill for the premium paid for the purchase, and minority stakeholders claim to the asset. After this first consolidation, assuming that the ownership percentage does not change, subsequent consolidations would post the parent and minority share of the subsidiary's earnings.

In the equity method, reported financial data of the equity unit is not taken into consideration in the consolidated financial report. The financial data for the equity unit is not entered into the consolidation system. Only the changes to the owner's equity are taken into consideration. This affects the investment value and goodwill stated in the consolidated balance sheet.

The proportional method of consolidation collects the units of the balance sheet and income statement into the consolidated statements based on the investor unit's proportion of ownership in the investee's unit. In other words, the percentage that is not owned is eliminated.

Let's consider an example of a parent company that acquires 70% ownership in a subsidiary using the purchase method of ownership. At the time of acquisition, the following transactions are performed:

- Consolidation of balances of parent and subsidiary
- Elimination of ownership that is common between the parent and subsidiary (the investment account of the parent that pertains to the subsidiary and common stock of subsidiary)
- Posting of any excess value paid that is above the book value of the acquired subsidiary to goodwill
- Posting of minority interest (the amount of minority interest appearing in the balance sheet is calculated by multiplying common shareholder equity of the subsidiary by the percentage of the minority interest—in this case, 30%)

After the first consolidation, assuming that the ownership percentage does not change, subsequent consolidations post the parent and minority share of the subsidiary's earnings.

In the following chapters, we'll discuss how you can use the SAP BPC application to perform the previously discussed tasks.

1.4 Summary

In this chapter, we discussed the importance of EPM software and introduced key concepts in the area of financial planning and consolidation. Chapter 2 offers an overview of the SAP BPC application and explores its use for planning and financial consolidation.

With a grasp of some basic concepts related to planning and consolidation, let's focus on the features of SAP BPC for NetWeaver to see how we can use the tool to perform planning and consolidation tasks.

2 Overview of SAP BPC

In this chapter, you'll find an overview of the SAP Business Planning and Consolidation (SAP BPC) application, SAP NetWeaver version 10.0, and an introduction to some of its key features. We'll discuss the flexibility, control, and ease this application offers in managing and analyzing data, and we'll explain how you can use the software to meet both the planning and consolidation needs of your organization. We'll also detail the multifaceted range of applications where you can use SAP BPC and expound on how you can use it to support a reliable and effective decision-making process. Finally, we'll discuss the architecture of SAP BPC for NetWeaver and introduce you to objects created in the environment.

Note

SAP has changed the title of SAP BPC from SAP BusinessObjects Planning and Consolidation to SAP Business Planning and Consolidation. We'll refer to this product as SAP BPC.

In Section 2.1, you'll learn about a key requirement most business users look for when using a software application: the ability to manage and analyze data with less dependence on an IT solutions team. We'll explain how SAP BPC satisfies this need.

In Section 2.2, we'll explain how you can use the application to meet planning and consolidation requirements, discuss its implementation in different areas of planning, and explain how you can derive competitive advantage by using it. We'll also cover how the application provides an environment for performing legal consolidation and management consolidation of an organization. This is especially important for multiple-entity organizations that are required to perform legal consolidation and to release financial results of the entire company per statutory requirements.

In Section 2.3, we'll discuss the architecture of the SAP BPC for NetWeaver and explain how it interfaces with the SAP NetWeaver Business Warehouse (BW) system.

In Section 2.4, we'll introduce you to the SAP NetWeaver BW system and explain the process of extracting, transforming, and loading data from a source system to an SAP NetWeaver BW system. The section also discusses the use of Business Content to expedite the development of data warehouse applications. We will also highlight some key features available in SAP NetWeaver BW 7.3 version.

In Section 2.5, you'll learn about the objects configured in SAP BPC and the relationship between these objects and those created in SAP NetWeaver BW.

2.1 SAP BPC: Business User-Owned and Managed

Sometimes business needs are predictable, but at other times, they are neither clearly defined nor easy to anticipate. This element of unpredictability has a direct bearing on the IT applications that are developed for business users. The model of an application should be free form and flexible enough to meet the needs of the business users and the business itself. When models are rigid or unwieldy, the application won't meet the objectives for which it was developed and therefore will go unused.

So you should consider the cost of retooling an application to support additional business requirements when designing the application. A clear process should be in place for how the data model fits the current scope and how it will support the business if the design requires changes during the life of the application. Organizations often find themselves spending increasing amounts of money to redesign their applications to meet user needs when some of this money could have been saved if good design principles had been followed in the first place.

In addition to the design element, the software that is used for building the application should be flexible enough to support changes. Because not every business change can be anticipated, a good software tool should be able to accommodate changes without requiring too much time or too high a cost.

2.1.1 Software Designed to Support Change

The following are some software tool aspects that ensure usability and flexibility to support changes:

- **Reporting**
  You must be able to design custom reports on your own to support any analyses that may not be supported by standard reports delivered by the IT solutions team.
- **Control of data**
  Users own data, and it is only right that you have the ability to control how this data is loaded into the system. The software tool should let you load data into the application so it can be used for analysis later. This is particularly important for users of planning applications, where you have the discretion to decide what data to use, and when and how to use it. This applies to consolidation applications as well, regarding how you want to use your data for statutory reporting. The options in the software tool, such as the ability to schedule the loading of data and to replace existing data, enable you to make more efficient use of the application.
- **Business rules**
  Business rules are used when you want to transform data that is loaded into an application. They are used to perform certain tasks in planning and consolidation applications such as currency conversion, account transformation, carry forward balance, and so on. You should be able to configure these rules and execute the processes.
- **Out-of-the-box functions**
  Depending on the application, the software should provide you with all of the functions that are used to meet the principal requirement of the application. These functions should be made available as part of the software tool to reduce the total cost of ownership (TCO) and help you use the application without relying on other tools to perform these functions.
- **Custom settings**
  The software should enable you to see report descriptions in your own language (e.g., if you are in France, you should be able to see the application user interface in French).
- **Customization**
  You should be able to customize the application without having to rely on the help of your IT solutions team. Customizations may involve a change to the data load process or reporting logic, for example.
- **Collaboration**
  Some applications require collaboration to ensure that all members of the

business team are in sync and have the latest information. For these applications, it's essential that the software provides features that allow collaboration. Collaboration may assume different forms. For example, you may post a document to get input from another user, or you may post a document for others to read and use in their decision making.

- **Distribution and collection**
  As a corollary to collaboration, it may be necessary to disseminate data to other business users who may in turn modify the data and may want to retract the modified data back to the application. For example, if you send a planning sheet in offline mode to a user to perform updates, the user performs the updates and returns the planning sheet to be loaded into SAP BPC. The software tool needs to allow you to distribute and modify information and retract modified information back to the application.
- **Programming interface**
  You may need to develop scripts that allow you to perform specific functions within an application. The software tool should provide an easy-to-use programming interface that enables this.

2.1.2 SAP BPC (Microsoft and SAP NetWeaver)

How does SAP BPC accommodate the need to support usability and flexibility to support change? The application provides an environment that satisfies all of these requirements in a manner that grants you sufficient flexibility in using your applications. Next, we'll discuss some of the features of the application and some of the disadvantages of relying solely on Excel for planning purposes.

Features of SAP BPC

SAP BPC lets you develop your own reports, enabling you to decide how to view your data. Although the data you can view is governed by your security access, you can customize your reports in a way that suits your decision-making requirements.

The application also allows you to load data yourself. This has two benefits: it eliminates your dependence on the IT solutions team and allows you to load data at any time, which in turn provides more control.

SAP BPC supports the creation of business rules that dictate how data is transformed in the application. For example, you can set up business rules for performing currency conversions.

The application also provides out-of-the-box functions for performing planning and consolidation. These functions can be used as-is or in combination with additional business rules to support specific application requirements. The following are some of the out-of-the-box functions:

- **Currency translation**
  You can use this function if your organization has its place of business in more than one country or does business in different currencies. Currency translation is used in both planning and financial consolidation applications. The currency translation process allows business users to translate transactions to one or more reporting currencies.
- **Intercompany elimination**
  This is applicable for an organization that has entities that do business with each other. An income statement or consolidated balance sheet of an organization shows transactions that relate to external companies, so the transactions (such as sales and COGS or receivables and payables) that result from exchange of goods or services within the organization should not be counted. In these cases, it's essential to eliminate the intercompany transactions. The intercompany eliminations process creates entries that eliminate intercompany transactions.
- **Allocations**
  Planning may take place at a high level, and you may later allocate the plan to lower levels. For example, you may need to allocate planned expenses incurred by the corporate office to each of its divisions based on some ratio, perhaps allocating corporate expenses to divisions based on the planned revenues of each division. The logic to perform this allocation is available in SAP BPC.
- **Account transformation**
  In some scenarios, accounts are consolidated or transformed into another account, an activity that is used extensively in cash flow applications. Account transformation business rules are used to aggregate values posted to specific combinations of source account, flow, category, and data source, and then post them to aggregated destination account, flow, category, and data source.
- **Control settings**
  The control settings or validation rules functionality in SAP BPC is a check mechanism that enables an organization to ensure accuracy of data. For example, using the control setting table, an organization can set up business rules that would compare balances in assets and liabilities/owner's equity accounts for a given period and report any variances for that period.

- **Carry forward opening balances**
  The carry forward of balances occurs when the closing balance of accounts for a fiscal period is transferred as the opening balance of a subsequent fiscal period. This transfer is an essential step in the creation of a balance sheet for an organization.
- **Eliminations and adjustments**
  The elimination and automatic adjustments logic supports the process of generating a consolidated financial statement. The automatic adjustment business rules are used for elimination of intercompany transactions and for consolidation of investments.

SAP BPC allows you to view and store information in multiple languages. It also includes a feature to augment collaboration by enabling you to enter comments about your data. This provides context information to the data (making it more understandable) and aids in decision making. For example, you can enter a comment to explain a reduction in overhead expenses from last year to the current year. This is clearly a better solution than storing this information in a different location that may or may not be accessible to the user making the decision.

Another feature of the application, Work Status, allows you to lock data so that it cannot be changed in the application. Using this feature, you can define the range of data that can be modified or locked. For example, if a sales representative creates a sales plan and sends it to his manager for approval, the data can be locked from being changed during the approval time frame.

SAP BPC also enables the distribution and collection of financial data. The application allows you to send plan information to other people, even people who do not use the application. The recipient can then make modifications to the data, and the changes can be retracted back to the system. This facilitates the exchange of data and truly supports collaboration between and participation of all people involved in fulfilling the organization's planning objectives.

Another important feature of this application is its easy-to-use programming interface, which allows you to create and execute logic that may be used for planning and consolidation. For example, you can develop a script to perform revaluation of a plan by a certain percentage. The application lets you implement logic using a variety of methods to manage the data, and it lets you include logic than can be

categorized into three broad areas. We'll discuss this at length in Chapter 6; for now, we'll only offer a brief description of each area:

- **Dimension logic**
  You configure this via simple mathematic formulas. For example, you could set a formula in the application to subtract the COGS from the gross sales to obtain the gross profit.
- **Business rules**
  You can develop rules to perform a certain function in planning or consolidation. For example, you can define custom business rules for currency translations, intercompany transactions elimination, account transformations, and so on to tailor the application to the needs of the business.
- **Logic script (or K2 script logic)**
  You can develop scripts that perform certain functions. This feature is generally used to apply detailed business logic to achieve a specific function, especially when out-of-the-box functionality cannot satisfy the business requirement. The logic script is a fully developed language for this application and can meet any special requirements not available in the standard application. For example, you may require logic script when you want to determine the depreciation of a certain group of assets based on some logic; if the logic is not readily available, you'll need to create it in SAP BPC.

Note

Script logic in SAP BPC 7.5 is referred to as logic script in SAP BPC 10.0.

The application frontend for planning, consolidation, and reporting runs on top of Excel, which is widely used by planners and accountants all over the world. This lets you perform planning and consolidation functions in the familiar Excel environment and helps you avoid spending time learning new software. Creating graphical data in Excel provides an intuitive display of the data that can be easily understood, interpreted, and analyzed.

Comparing Microsoft Excel and SAP BPC

Some organizations solely use Excel for planning, but there are many disadvantages to this approach:

- **Managing Excel spreadsheets**
  In Excel, planning data is stored in spreadsheets, which can be an issue from a security perspective because the only option to control access is to keep the files on a shared network drive and provide access only to select users. This in turn creates additional work for the IT solutions team. SAP BPC alleviates this issue by storing the data in a database and enforcing security on the data.
- **Versions**
  When a new version of data is required, SAP BPC can store the data in a separate version while maintaining the current data. In Excel, the data file has to be saved under a new name as the new version, which leaves room for user error when naming files.
- **Floating files**
  In Excel, it's possible for different users to maintain the same file on their respective computers, which makes it difficult to track whose file is the latest and most reliable version. SAP BPC is a central repository for financial data and can be accessed by several users at the same time.
- **Collaboration**
  Excel does not support true collaboration, whereas SAP BPC allows you to store comments that can inform other users of important information.

SAP BPC also lets you integrate data into Microsoft Word and Microsoft PowerPoint for reporting purposes.

2.1.3 SAP BPC for NetWeaver

SAP BPC for NetWeaver provides additional advantages. Because it is integrated into the SAP NetWeaver BW system, the data is stored in BW InfoCubes and enjoys all of the advantages of this system. These advantages apply to the SAP NetWeaver version:

- **Star schema**
  The technical architecture of an SAP NetWeaver BW InfoCube consists of a central fact table surrounded by several dimension tables, all grouped in the form of a star schema. SAP BPC for NetWeaver uses the extended star schema for storing data, which makes it a superior choice for reporting.

- **Database independence**
  SAP BPC for NetWeaver runs independently of the database and can run on major databases such as Oracle, DB2, SQL Server, and Informix.
- **SAP NetWeaver BW Accelerator (BWA)**
  SAP NetWeaver BW Accelerator (BWA) is an optional tool you can use in SAP BPC for NetWeaver to improve reporting performance. BWA provides enhanced reporting performance, even with a large data set. The investment in BWA not only helps with providing an enhanced reporting experience but also contributes to considerable cost savings by minimizing troubleshooting issues related to reporting performance.
- **ABAP language support**
  The logic script, which we mentioned earlier, is the programming language used for developing logic. SAP BPC for NetWeaver also allows you to use ABAP to develop this logic. You can develop a business add-in (BAdI) interface in the backend SAP system using ABAP, which can then be called from the logic script that is developed in SAP BPC. You can use scripts coded using ABAP to handle complex business requirements and simultaneously leverage the experience and expertise of the IT solutions team. This feature helps reduce development time for organizations that have been using an SAP system for a long time because developers will be familiar with the ABAP programming language and can use it to deliver customizations.

From the previous discussion, you now have an idea of the many features SAP BPC offers. Most of the features discussed require less assistance from the IT solutions team on an ongoing basis (although they will of course be involved in both the initial stages of the development of the applications and their ongoing support). SAP BPC's built-in features allow you to automatically assume ownership of the application and then partner with the IT solutions team in fulfilling specific business needs.

In the next section, you'll see how to use the application for performing both planning and consolidation functions.

2.2 Unified Planning and Consolidation

SAP BPC provides a full-fledged environment for organizations to create flexible and powerful applications for different types of planning. It can also be used

for consolidating and reporting financial data to satisfy your organization's legal requirements.

2.2.1 Planning in SAP BPC

In today's world of intense competition, it is necessary for organizations to plan ahead and stay ahead. Planning is the life-blood of business, and the ability to spot opportunities and prepare for various courses of action has become imperative because consumers' fast-changing needs require businesses to be vigilant and act with vision. To meet this objective, organizations are looking for software that can provide an enabling environment to handle planning needs.

As we have discussed, SAP BPC has many features that make it an ideal tool for planning because it is designed to maximize your ability to manage and take ownership of data. Its robust interface allows for customization and collaboration and makes it a compelling choice for planning. Unlike most software, this application reduces the dependence on the IT solutions team while providing more time for the team to handle the portions of application development and maintenance they best serve (i.e., system configuration and performance).

2.2.2 Consolidation in SAP BPC

The consolidation functionality of SAP BPC provides an environment to report the financial results of your entire organization. You can use it to determine the financial health of the company as a whole, after taking into account the currency translations, elimination of intercompany transactions, and other necessary adjustments. When the consolidation report is prepared, the application considers different types of investments among organizational units.

Consolidation provides the means for internal and external enterprise reporting of a company's financial data. When the reporting is external, it serves the objective of statutory compliance, also known as *legal consolidation*. This is classified as formal reporting and must satisfy the acceptable accounting practice that is applicable to the country where the consolidation is done.

Internal reporting is also called *management consolidation*. This provides more latitude and flexibility in defining the methods used for consolidating data. The objective of this type of consolidation is to provide you with the necessary tools for decision making. You can consolidate based on customer-defined consolidation units, which

can represent, for example, companies, plants, business areas, profit centers, or cost centers. You can also portray matrix organizations, for example, by using a combination of companies and profit centers.

The first step you perform in consolidation is to standardize the financial data reported by individual consolidation units to adhere to the group's accounting standards. You then translate the standardized financial data from the various local currencies into the group currencies (if the local and group currencies are different). Finally, you eliminate the effects of intercompany transactions. So you calculate the consolidated financial statements as if the group were a single entity. You can use the reporting functions of SAP BPC to analyze and report on your consolidated financial statement data. You can use SAP XBRL Publishing to create XBRL documents based on data in the SAP BPC application to exchange financial data with accounting jurisdictions such as banks, regulators, and agencies.

Let's discuss the features and advantages of performing consolidation in SAP BPC. SAP BPC allows you to use different charts of accounts for consolidation, enabling you to generate several consolidated financial statements in parallel to accommodate different accounting principles.

You can collect data for consolidation in SAP BPC for NetWeaver using the following methods:

- Online data entry
- Flexible upload from a data file
- Loading from an InfoProvider

The application allows you to execute the following tasks:

- Currency translation
- Account transformation
- Allocation
- Balance carry forward
- Intercompany reconciliation (matching)
- Intercompany booking
- Intercompany eliminations

- Consolidation of investments
- Validation

Performing consolidation with SAP BPC has the following advantages:

- **Legal and management consolidation**
  The application provides a systematic process to perform legal consolidation. You can also perform management consolidation using different sets of data to highlight different scenarios for management, as well as parallel consolidations with different categories of data and using different accounting principles.
- **Accuracy**
  The design of the application's system for consolidation enables the consolidation process to run in a controlled manner, with the ability to perform checks and balances along the way. This mitigates any errors during the process and ensures the completeness and accuracy of the data.
- **Multiple source systems**
  You can compile financial data from various source systems into a single system for the purpose of consolidation.
- **Reporting**
  You can generate reports based on consolidated data, which you can then use as a tool for measuring the organization's performance. The reports allow you to view key metrics such as return on equity, ROI, and so on.
- **Validation of data**
  You can analyze different sources of data used to generate the consolidated data to ensure that the data is valid and reliable. You can also reconcile the data in the application with the source system, facilitating reliability.
- **Collaboration**
  The application supports the extensive collaboration needed to gather data from multiple sources. This helps with the consolidation of data from diverse sources.

The main business benefits of SAP BPC include reduced planning and consolidation cycle times, accurate reporting that meets statutory requirements, and minimized compliance risk due to quality of data.

You should now have an understanding of how organizations can use SAP BPC to meet both their planning and consolidation requirements. In the next section, we'll discuss the architecture of SAP BPC for NetWeaver and explain how it interfaces with the SAP NetWeaver BW system.

2.3 SAP BPC for NetWeaver Architecture

The four-tiered architecture of SAP BPC for NetWeaver, which is shown in Figure 2.1, is designed to leverage the capabilities of SAP BPC and SAP NetWeaver BW. You can use the frontend tool to configure planning and consolidation features. The data that is entered or loaded through this frontend client interfaces is then stored in the SAP NetWeaver BW system, which is based on the extended star schema and is designed to provide high-performance reporting.

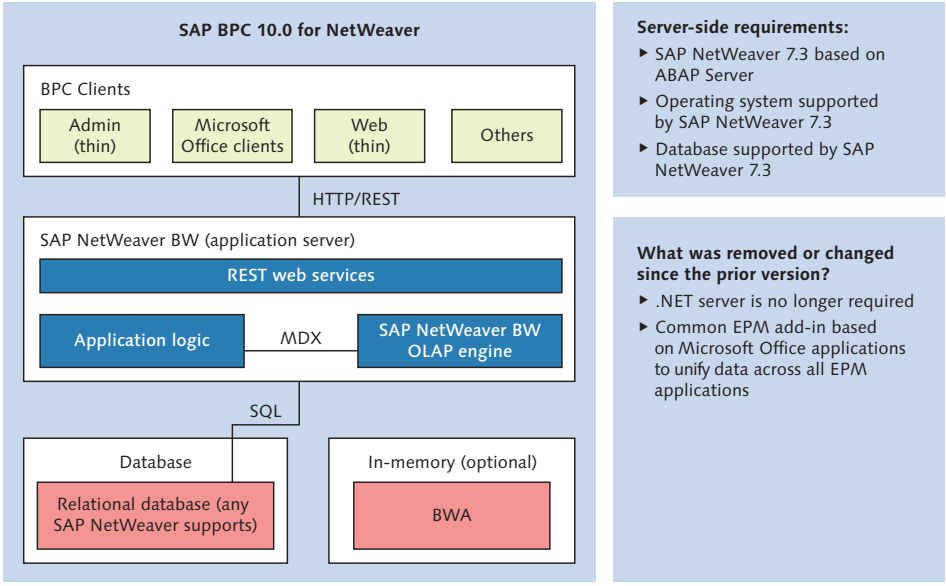


Figure 2.1 SAP BPC 10.0 for NetWeaver Architecture (Source: "SAP BusinessObjects Planning and Consolidation 10.0, Version for SAP NetWeaver" Presentation)

Let's deconstruct the SAP BPC for NetWeaver architecture from the top down to better understand how it is designed.

2.3.1 BPC Clients

The SAP BPC 7.5 client software is replaced with a new SAP EPM 10 Client that works across the SAP EPM suite of products. The client interface for SAP BPC 10.0 for NetWeaver, represented in Figure 2.1 by the BPC Clients layer, consists of

Admin, MS Office clients, Web, and Others. The Web Client and MS Office clients are primarily used for configuring and executing tasks in SAP BPC for NetWeaver.

The Web Client supports the configuration of the planning and consolidation applications. It is also available to report the data stored in SAP BPC. The Web Client replaces the Admin Client that was available for configuration in SAP BPC 7.5. The administrative activities in BPC 10.0 are handled using the Web Client. The new interface provides several improvements over the features that were available in SAP BPC 7.5. The web-based client is a zero footprint interface and does not require installation on the web. It is both the foundation on which applications are built and the tool to manage unstructured documents, develop business process workflows, monitor the status of consolidation and planning applications, and set custom parameters for an application.

The MS Office reporting client interface is based on the add-in client for Microsoft Excel, Word, and Powerpoint and reports data stored in SAP BPC and other SAP EPM applications. The EPM-based add-in client uses EPM functions and replaces the EvDRE functions that were used in SAP BPC 7.5. In addition, it provides sophisticated reporting capabilities and unifies data across EPM applications. It can also be used to report data residing in the SAP NetWeaver BW system.

The Web Client interface is supported in the following clients:

- Windows 7 and Vista Client (32-bit and 64-bit) and Windows XP (32-bit)
- Internet Explorer 7 and 8
- Microsoft Office 2003, 2007, 2010 (32-bit and 64-bit)

2.3.2 Connectivity

The connectivity between the client interfaces and the SAP NetWeaver system is enabled using Representational State Transfer (REST) web services. This is a significant departure in the architecture from the Web Server (IIS) and .NET application server that was used in SAP BPC for NetWeaver 7.0 and 7.5 for interfacing between the client and the SAP NetWeaver system.

The Simple Object Access Protocol (SOAP) web service is used for communicating with external EPM applications such as SAP Financial Information Management.

2.3.3 SAP NetWeaver Application Server

The SAP NetWeaver Application Server is also the ABAP Application Server. The Application Server includes the application logic and the engine to process requests from the client. It is also used to process MDX expressions. This is the tier where the requests received from the client interface are processed using REST web services and are then sent back to the client interface.

2.3.4 Database Server

The database server is the tier where data is stored. The SAP NetWeaver BW system is the backend system where objects are created and the data is stored for SAP BPC. The SAP BPC for NetWeaver version is database agnostic and supports all popular databases, including Oracle, Informix, SQL Server, and DB2.

The client interface interacts with the SAP NetWeaver BW system to manage objects and data. When you create, read, update, or delete objects or data using the client interface, the system performs these tasks on the SAP NetWeaver BW system.

Although a majority of the development tasks related to SAP BPC are accomplished using the client interface, a few tasks are performed directly in the SAP NetWeaver BW system. These include activities related to performing restore and backup (Transaction UJBR), viewing files in the file system (Transaction UJFS), improving performance, setting validation rules (Transaction UJ_VALIDATION), and creating custom ABAP programs. Custom process chains, which may be used to schedule data loads, are also created directly in the SAP NetWeaver BW system.

The data related to a planning or consolidation application may reside in a source system like SAP ECC. In this case, you may need to extract master, text, hierarchy, and transaction data from a source system into the SAP NetWeaver BW system before loading the data to an SAP BPC system. In this case, the SAP NetWeaver BW system is used as a staging system for data before being loaded to the planning and consolidation application. The objects related to setting up the staging environment are created directly in the SAP NetWeaver BW system.

Note

We caution you against performing any manual configuration for SAP BPC directly in the SAP NetWeaver BW system, except for the few tasks mentioned. You should configure the application using the frontend tools (Web Client and the Microsoft Office add-in client).

As you can see, the integration of SAP BPC with SAP NetWeaver provides many benefits to leverage the power of Excel and SAP NetWeaver BW systems.

In the next section, we'll introduce you to objects created in the SAP NetWeaver BW system.

2.4 Introducing SAP NetWeaver BW

When you create an object using the SAP BPC frontend, the system creates equivalent objects in the SAP NetWeaver BW system. In addition, the SAP NetWeaver BW system may sometimes be used to stage data coming from various source systems before being loaded to the planning and consolidation application. In the subsections that follow, we'll discuss the objects in SAP NetWeaver BW, the data flow process the system utilizes, process chains, and Business Content.

2.4.1 Objects Used in SAP NetWeaver BW

In this section, we'll introduce you to the objects used in SAP NetWeaver BW: InfoObjects and InfoProviders.

An *InfoObject* is the basic object for building a data model in SAP NetWeaver BW. There are several different types of InfoObjects:

- **Characteristics**
  A characteristic InfoObject represents an entity of a business or an attribute related to an entity, such as customers, materials, and employees. An InfoObject that provides additional information about an entity is called an attribute. Attributes are used in conjunction with the parent InfoObject. For example, customer address and phone number are InfoObjects included as attributes in the Customer InfoObject.

  You can configure an InfoObject to store master, text, and hierarchy data:

  - When you define an InfoObject to store master data, it contains the master data table with the InfoObject and the attributes associated with it. For example, the customer number, along with city, state, zip code, and country, can be stored as master data.

- When you design an InfoObject to store text data, it contains the text values of the InfoObject. For example, the names of customers can be stored as text values.
- When there is an inherent parent-child relationship in data, you can define the InfoObject as a hierarchy InfoObject. This is useful for analysis. For example, the reporting relationships between manager and employees can be defined as a hierarchy.

- **Time characteristics**
  The time characteristic InfoObject is used to set the value for the time-related characteristic of a transaction, such as calendar month, calendar year, fiscal year, and so on.
- **Unit characteristics**
  The unit characteristic InfoObject provides meaning to quantitative data. Currency and units of measure are examples of unit characteristics. For example, the sales amount for a transaction can be recorded as a currency. The quantity sold can be recorded as a unit (kilograms and pounds).
- **Key figures**
  The key figure InfoObject represents the quantitative measure associated with a transaction. Examples of key figures are the quantity and amount associated with a sales transaction.

An InfoProvider is the object provided for storing data in SAP NetWeaver BW; it is comprised of a collection of InfoObjects and can accordingly contain characteristic, time characteristic, unit characteristic, and key figure InfoObjects.

InfoProviders fall into one of two categories: physical InfoProviders and logical InfoProviders. Because they physically store data, *physical InfoProviders* are also known as *data targets* and can be InfoObjects, InfoCubes, and DataStore objects.

An *InfoCube* is a collection of InfoObjects and represents the extended star schema architecture in SAP NetWeaver BW. It is made up of a fact table and a number of dimension tables. Related characteristics in an InfoCube are grouped together under dimensions. When you load data to an InfoCube, a dimension ID is created by the system for each unique combination of characteristics in a dimension. A record in a fact table is made up of the individual key figures and the dimension IDs associated with a transaction. The individual characteristics in a dimension—and the corresponding attributes of a characteristic—are related using surrogate IDs (SIDs).

The InfoCube is the recommended approach for reporting in SAP NetWeaver BW because it provides the full advantage of the star schema architecture.

A *DataStore object* (DSO) is a transparent table that usually serves as the first layer of extracting data from a source system. The data loaded in a DataStore is often used for detailed analysis because it contains raw data extracted from the source system. The data extracted into the DataStore can be further loaded into another InfoProvider, depending on the particular requirements of analysis.

In contrast with physical InfoProviders, *logical InfoProviders* do not physically contain data but are used for providing views of data. They are InfoSets, MultiProviders, and virtual InfoProviders.

An *InfoSet* lets you join objects that store data. For example, two DSOs can be joined together to create an InfoSet. The joins created in an InfoSet are database joins. There is considerable flexibility when you define an InfoSet using inner and outer joins. Using an outer join, you can view all of the data defined in the left object of the InfoSet—for example, you can view the sales of all customers, including customers who did not have any sales reported in a particular period. InfoSets can also be used to combine data from an InfoCube and another InfoProvider.

A *MultiProvider* provides a powerful feature to view data contained in more than one InfoProvider. For example, you can use a MultiProvider to view data from an InfoCube and a DataStore. The MultiProvider provides a union of data in the underlying InfoProviders, which helps to combine the data available in more than one InfoProvider.

A *virtual InfoProvider* enables remote access of data from a source system connected to the SAP NetWeaver BW system.

2.4.2 Extracting, Transforming, and Loading Data in SAP NetWeaver BW

Let's tackle the process of extracting, transforming, and loading data into data targets in SAP NetWeaver BW. You'll extract data into SAP NetWeaver BW using the DataSource as the source of data. You can configure DataSources to extract data from a variety of source systems:

- **SAP systems**
  Data can be extracted from SAP systems (SAP R/3, SAP CRM, and other new dimension products offered by SAP) into SAP NetWeaver BW. There is a tight

connection between SAP systems and the SAP NetWeaver BW system. The service application programming interface (API) provides the interface to extract data from these systems.

- **Flat files**
  Data is extracted from a standard file into SAP NetWeaver BW (e.g., an Excel file). A DataSource is defined in the SAP NetWeaver BW system corresponding to the file layout for extracting data.
- **Web applications**
  An XML interface can be used to extract data from a web application using the Simple Object Access Protocol (SOAP).
- **DB Connect**
  A direct connection to a variety of external databases systems, such as Oracle, Informix, and so on, can be established using DB Connect.
- **Universal Data Connect**
  This connection provides access to external relational databases and multidimensional databases via the Java 2 Platform, Enterprise Edition (J2EE) server.
- **Third-party systems**
  A business application programming interface (BAPI) is used in conjunction with third-party tools to extract data into the SAP NetWeaver BW system. Some examples of third-party systems are Informatica and Ascential.

DataSources can be configured for each of these source systems.

The data transfer process (DTP) in SAP NetWeaver BW provides a flexible and improved process of extracting data from a source system, transforming the extracted data, and loading the transformed data into a data target.

The following are the components of the DTP in SAP NetWeaver BW:

- **Persistent staging area (PSA)**
  A PSA table is generated when a DataSource is activated in SAP NetWeaver BW. The transparent PSA table is the first layer for storing data. When data is requested for a DataSource from a source system, the data is initially loaded into the corresponding PSA table of the DataSource.
- **Transformations**
  A transformation process is defined between a source and target object in SAP NetWeaver BW. The objective of the transformation is to map data coming from

a source to a target using specific rules. It also allows you to apply changes to data coming from a DataSource.

You can use the DataSource (PSA), InfoSource, DSO, InfoCube, InfoObject, and InfoSet as source objects in the transformation; you can use the InfoSource, DSO, InfoCube, and InfoObject as target objects in the transformation.

The transformation rule allows you to map fields and InfoObjects to a target InfoObject, set a constant value to a target InfoObject, develop a routine in ABAP to perform the mapping, create a formula, look up master data attributes to determine the value of a target InfoObject, and perform time determinations.

An InfoSource acts as an additional layer of transformation before data reaches a data target. You can use it as an optional process when two or more transformations are required before data is transferred from a DataSource to a data target.

- **Data transfer process**
  The data transfer process (DTP) facilitates the flow of data in SAP NetWeaver BW from one persistent object to another; for example, after data is transferred from a source system into the PSA, the DTP is used to load from the PSA to a data target.

  PSA, InfoObject, InfoCube, and DSOs are examples of persistent objects in the SAP NetWeaver BW system.

  The DTP may encompass one or many transformations. A transformation process is defined to provide the rules to map data between a source object and a target object. Transformation rules provide the ability to apply changes to the data coming from a source to a target. The DTP moves data from the source to the target using the rules specified in the transformation.

- **InfoPackage**
  An InfoPackage is the object used to request data for a DataSource from the source system. After the InfoPackage requests data, the data for a DataSource is transferred from the source system to the PSA table.

Figure 2.2 shows the flow of data in SAP NetWeaver BW. Transformation rules are created between the source and target objects, and a DTP is used to load data from one persistent object to another.

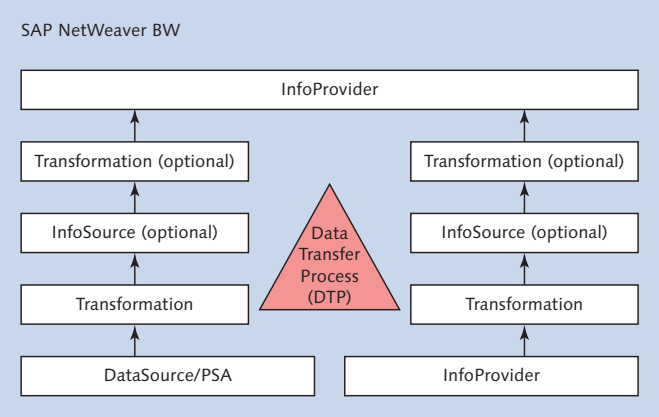


Figure 2.2 Data Transfer Process in SAP NetWeaver BW

2.4.3 Process Chains

A *process chain* provides different types of processes to manage data within the SAP NetWeaver BW system. You use process chains to automate the process of extracting, transforming, and loading data, as well as to more effectively administer data. For example, the InfoPackage that is executed to request data from the source system, as well as the DTP process that is used for moving data within SAP NetWeaver BW, can be automatically scheduled in a process chain.

2.4.4 Business Content

When discussing extraction, transformation, and loading in SAP NetWeaver BW, we mentioned that you can use BW as a staging environment before loading data into SAP BPC. *Business Content* is what allows you to build this staging environment; it consists of preconfigured objects that help accelerate the development process for an SAP NetWeaver BW application. Business Content is available in the following areas:

- **DataSources**
  SAP supplies DataSources to bring data from SAP R/3 to SAP NetWeaver BW systems. The logic for extracting the data is supplied with the DataSource, which greatly reduces the time and effort involved in extracting the data. Some of the DataSources have capabilities to extract delta changes to data since the last load.

A DataSource that is delta-enabled brings over only the data that was created or modified since the last extraction from the source system.

- **InfoObjects**
  SAP supplies standard InfoObjects used in business applications (for example, Customer, Plant, Material, and Material Group InfoObjects).
- **InfoProviders**
  Standard business content InfoProviders are available for a wide area of applications. These include InfoCubes, DataStores, and MultiProviders.

In addition, Business Content is available for objects that are used to load data (e.g., InfoSources, transformations, and DTPs). SAP also delivers the processes to control data loading using InfoPackages and process chains.

You can see that the process of setting up the SAP NetWeaver BW system is greatly facilitated by Business Content. Although Business Content may not be able to provide a solution for all of the analysis requirements of a business, it can be used as the starting point to prototype a solution. Business Content can subsequently be enhanced to meet the analysis requirements of specific users.

2.4.5 Key Features in SAP NetWeaver BW 7.3

SAP NetWeaver BW 7.3 includes significant improvements over the prior versions and is geared toward reducing the total cost of development (TCD) and total cost of ownership (TCO) for developing and maintaining the system. Some of these improvements are as follows:

- **Graphical data flow modeler**
  This feature provides an interface to design a data flow via drag and drop. In addition to providing the ability to create templates for data flows that can be used consistently across the system, it offers an intuitive process to design the objects inside the system. For example, a developer can create a data flow that starts with the DataSource InfoProvider (DSO or InfoCube) and then includes a MultiProvider used for reporting. In addition to designing a data flow, a transport can be created to collect objects based on the data flow.
- **Semantic partitioning**
  This feature helps to spread the data across multiple InfoCubes or DSOs by providing an interface to select the required InfoObject(s) and assign individual values to use as the basis for partitioning when designing an InfoProvider. This

helps improve the performance of the InfoProviders when retrieving data. In addition, it mitigates the issue of current reporting when a data load fails because it only affects the InfoProvider(s) loaded for the partitioned value. If the loads for other partitioned values are successful, the data in those InfoProviders can report the current data.

- **Improved performance**
  Several features are available in this version that improve the performance of data loads and offer faster access to the data when reporting:
  - The process of activating data in a DSO has been changed to significantly accelerate the time taken to activate the data.
  - The previous versions of BW supported the InfoCubes to be partitioned by the time InfoObjects. This helps to improve the performance when retrieving the data from the InfoCube. Starting with this version, DSOs can be physically partitioned.
  - A transformation rule can be defined to fill an InfoObject in the target to read data from a DSO.
- **Hybrid InfoProvider**
  The hybrid InfoProvider reduces the time to develop objects. This is useful when real-time data and use of optimal design are required for reporting. There are two options when configuring a hybrid InfoProvider:
  - Based on a DSO: The process takes data from a DSO and stores it in an InfoCube optimized for reporting. The DSO can then be connected to be a real-time enabled data source as the source of data.
  - Based on a virtual InfoProvider: The InfoProvider provides access to real-time data without having to store the data inside BW.
- **Analytic indexes**
  An analytic index can be defined inside the Analysis Process Designer (APD) and can include combining data using `JOIN/UNION` operations. The analytic index can be made available as an InfoProvider for reporting data.
- **SAP Data Services**
  A new source system type called *Data Services* is available in SAP NetWeaver BW to easily extract data from the SAP Data Services system. In addition, it is also integrated well into the data process flows inside the BW system. It can also be used to schedule data loads and monitor process chains in the BW system.

- **Process chain monitoring**
  This version includes additional features to monitor the status of process chain jobs in the BW system.
- **In-memory computing**
  SAP HANA is changing the way companies manage large volumes of data. SAP has introduced HANA with this version to counter the challenges with both managing large volumes of data and handling performance issues reporting the data. Figure 2.3 shows the evolution of in-memory computing and benefits that SAP HANA offers, as well as how it is significantly positioned to improve the management of data.

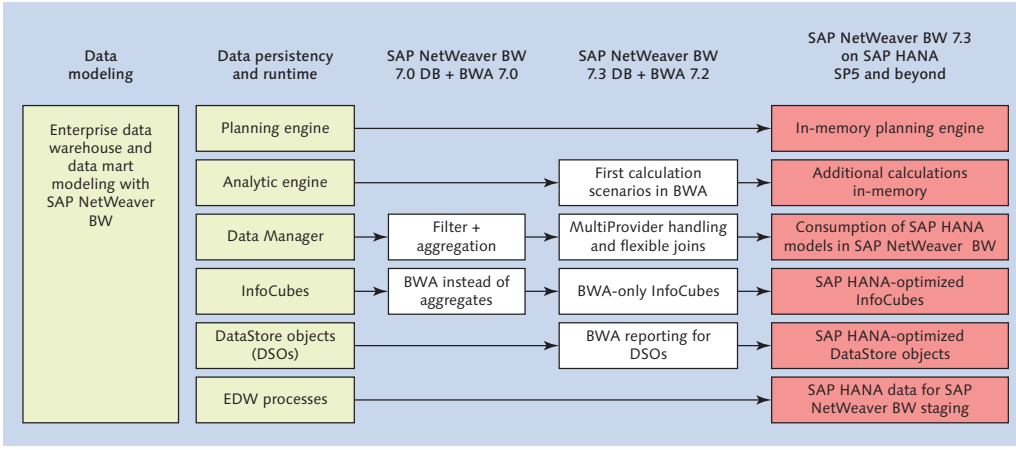


Figure 2.3 Evolution and Benefits of In-Memory Computing

In the next section, we'll introduce objects created in the SAP BPC system.

2.5 Terminology and Objects in SAP BPC

In this section, we'll discuss some of the terminology used and some of the objects created in SAP BPC for NetWeaver. We'll also explain how a request to create an object in the planning and consolidation application translates into creating an equivalent object in the BW system, as well as exploring objects that are configured in the SAP BPC system that would lay the foundation to carry out planning and consolidation functions.

Table 2.1 compares terminology used in SAP BPC 10.0 and the prior versions.

SAP BPC 7.5	SAP BPC 10.0
Appset	Environment
Dimensions	Dimensions
Applications	Models
Business rules	Business rules
Validation rules	Control settings
Script logic	Logic script
Work status settings	Work status settings
Journal templates	Journal templates
System report settings	Audit report settings

Table 2.1 Terminology Comparison between SAP BPC 7.5 and SAP BPC 10.0

Environment

An environment is the starting point to build models or applications in SAP BPC. Environments were called application sets or "appsets" in prior versions of SAP BPC.

The environment can be described as a functional area designed to include the applications or models associated with a segment of the business. For example, you can create an environment for sales applications and another environment for head-count planning.

Although the general objective of creating environments is to support a business requirement (such as sales planning, production planning, head-count planning, etc.), there are no rules about what models can be contained in an environment. You can decide how you want to develop models within an environment.

You create an environment via the Web Client. You can only create an environment using another environment as the basis. When the system is initially installed, an environment with the technical name ENVIRONMENTSHELL is available. This serves as a reference for creating other environments in the system. We caution you against making any changes to ENVIRONMENTSHELL; instead, use it to create a

new environment and customize that environment for the application. You should have system administrator access to create an environment.

Dimensions

Dimensions represent the master, text, and hierarchy data for each of the business entities (e.g., accounts, company codes, and categories). Dimensions belong to the environment in which they are configured. Note that dimensions cannot be used or shared across environments.

Secured Dimension

A dimension can be marked as secure for a model. When a dimension is marked as secure, specific read and write access has to be granted to users who use the model so they can access the data.

Properties

A dimension is designed to include *properties*, which provide additional meaning to the dimension. For example, an account dimension can include an *account type property* to indicate the type of account (whether it is an income account or an expense account). You can use property values as selection criteria for reporting.

Dimension Members

The data points associated with a dimension are referred to as *dimension members*. Each dimension member record represents master, text, and hierarchy data associated with the key of a dimension. A dimension member can store hierarchical relationships with another dimension member in the same dimension, which is very useful for reporting hierarchical relationships between data. There is no restriction on the number of hierarchies you can create.

Model

A model is built inside an environment. It is a collection of several dimensions and is designed to meet the planning and consolidation requirements of an organization. The model is where you interface to plan, consolidate, and report data. For instance, you might have a model for a cost center planning application to compare actuals versus plan based on cost centers or a model for legal consolidations that reports your organization's financial results.

Data Manager Package

SAP BPC provides standard out-of-the-box functionality to execute common tasks required for planning and consolidation such as currency translation, intercompany elimination, and so on. These tasks are executed via a Data Manager package.

User

A user is an individual who can use the models in an environment.

Team

A team is a group of users that is created based on the users' roles. For example, all of the developers can be grouped together and identified as a team.

Tasks

A number of activities that you can perform in SAP BPC are known as *tasks*. Examples of tasks are creating an environment, running a report, configuring security, and so on. Users are granted access to specific tasks based on their role in the organization.

Task Profile

Several tasks can be grouped together and included in a task profile. Task profiles can be assigned to a user or to a group of users in a team.

Data Access Profile

The member access profile identifies data-level access to a model and is relevant only for dimensions that were identified as secured dimensions in the model. For each model that includes dimensions marked as secured dimensions, the member access profile is configured to identify the read and write access available to users for each of the secured dimensions.

Business Rules

SAP BPC includes a rule-based table for defining certain common requirements for planning and consolidation. A company that has business operations in different countries may need to translate their data to one or more reporting currencies for financial reporting. The rules-based table provides a method for defining how local currencies should be translated to reporting currencies.

You can set up business rules for account transformations, carry forward balances, validating data, eliminating intercompany transactions, and setting up methods and rules for consolidation of investments.

Logic Script

Logic script is designed to provide greater flexibility in managing data for a model and includes SQL-based and MDX-based programming structures for custom programming. You should attempt to leverage business rules before writing logic script to meet a requirement. Business rules provide a convenient table-based approach to define the rules for different conversions. But there may be business scenarios that cannot be satisfied using business rules; in these cases, logic script provides the means to achieve that objective.

Relationship between Objects Created in SAP BPC and SAP NetWeaver BW

When objects are created in the SAP BPC system, the system creates equivalent objects in the SAP NetWeaver BW system. The corresponding objects created in the SAP NetWeaver BW system are shown in Table 2.2.

Object Created in SAP BPC	Object Created in SAP NetWeaver BW
Environment	*InfoArea* An environment in SAP BPC is equivalent to an InfoArea in SAP NetWeaver BW. When a request for creation of an environment is initiated in the SAP BPC frontend, an InfoArea is created in SAP NetWeaver BW with InfoObjects and InfoCubes under it. In SAP BPC, objects added under an environment are not shared across other environments. In SAP NetWeaver BW, objects created under an InfoArea can be shared by objects in other InfoAreas.
Model	*InfoCube* A model in SAP BPC is created with respect to an environment and is used for storing transaction data. When a request for creating a model is initiated in SAP BPC, an InfoCube is created in SAP NetWeaver BW.

Table 2.2 Relationship between Objects Created in SAP BPC and Objects Created in SAP NetWeaver BW

Object Created in SAP BPC	Object Created in SAP NetWeaver BW
Dimension	*InfoObject* A dimension in SAP BPC is equivalent to an InfoObject in SAP NetWeaver BW. It is created under an environment and is used for storing master data. Examples include account, product, and customer dimensions.
Property	*Attribute* A property is equivalent to an attribute included in an InfoObject and is dependent on the dimension under which it is created. For example, a customer dimension may include a property called "customer group."
Dimension member	*Master data record* The dimension member is equivalent to master, text, and hierarchy data in SAP NetWeaver BW.
Master data to store text and hierarchy data	*InfoObject* In SAP NetWeaver BW, an InfoObject should be enabled to store text and hierarchy data for an InfoObject. In SAP BPC, a property with the technical name `DESCRIPTION` is available to store text data. In SAP BPC, you can create hierarchies by editing the structure of the dimension. The system assigns technical name `PARENTH1`, `PARENTH2`, and so on for each hierarchy.
Unsigned data	*Key figure* Only one type of object is used for recording quantitative values for transaction data in SAP BPC: unsigned data. This object is equivalent to a key figure in SAP NetWeaver BW.

Table 2.2 Relationship between Objects Created in SAP BPC and Objects Created in SAP NetWeaver BW (Cont.)

2.6 Summary

In this chapter, we introduced the concepts and terminology used in SAP BPC, and you learned how organizations can use the SAP BPC application to meet both planning and consolidation requirements. You also studied the architecture of SAP BPC for NetWeaver and explored some of the advantages of staging an SAP BPC application on an SAP NetWeaver platform. In addition, you learned about certain objects that are configured in the SAP BPC and SAP NetWeaver BW systems.

In the next chapter, we'll discuss the details of configuring an application using SAP BPC.

This chapter shows you how to configure a financial planning model using SAP NetWeaver BW and SAP BPC for NetWeaver systems. We introduce you to a case study that will serve as the basis for creating this model.

3 Designing a Model with SAP BPC for NetWeaver

In this chapter, we'll introduce you to a case study of a company that has decided to plan its profitability by implementing SAP Business Planning and Consolidation (SAP BPC) for NetWeaver. The chapter explains the details of the company and how it wants to use the software for planning. The case study is also used as the basis for all planning examples presented in subsequent chapters.

Section 3.1 will introduce you to a case study that we use to illustrate the key features of SAP BPC for NetWeaver. This case study explains the details of an organization, Rich Bloom, Inc., which uses the software for planning its profitability.

Section 3.2 will explain the steps involved in modeling objects in SAP NetWeaver Business Warehouse (BW) to meet the requirements of the case study. The objects created in SAP NetWeaver BW are used in this case as a staging environment to store data coming from a transactional source system, before being loaded into SAP BPC.

Section 3.3 will explain the steps involved in modeling objects in SAP BPC for NetWeaver to meet the requirements of the case study. This section serves as a solid foundation to understand the features of the tool.

So let's explore Rich Bloom, Inc., which we'll follow as it uses SAP BPC for NetWeaver to positively impact its bottom line.

3.1 Rich Bloom Case Study

Throughout this book, we'll use a sample company called Rich Bloom to explain the features of SAP BPC for NetWeaver. In this section, we'll discuss the case study

to better understand the model company and the objectives the company is trying to meet using SAP BPC for NetWeaver.

3.1.1 Business

Rich Bloom is a clothing retailer headquartered in the United States with a presence in Germany and England. The company began its operations in 2000 and has seen its growth and profit increase rapidly due to its excellent management and the successful operation of its business. Rich Bloom has been able to vary its mix of products and consistently exceed customer expectations.

In a recent survey, the company determined that the ongoing worldwide recession has caused a reduction in the amount people spend on clothing. This has energized the company to further strengthen its planning process so it can continue to grow and thrive under difficult business conditions.

3.1.2 Offices

The company's business is incorporated as follows:

- 20—Rich Bloom, Inc., San Diego, CA, USA
- 22—Rich Bloom Corporate, Inc., Philadelphia, PA, USA
- 23—Rich Bloom New Markets, Inc., Houston, TX, USA
- 25—Rich Bloom Ltd., London, UK
- 30—Rich Bloom AG, Frankfurt, Germany

3.1.3 Products

The company sells clothing for teenagers and preteens, including the following Rich Bloom (RB) products:

- RB T-shirts
- RB shirts
- RB jackets
- RB designer jeans
- RB Apollo shirts

The company has partnerships with large wholesalers involved in the clothing business, through which it sells its products. Rich Bloom also has a few retail outlets in several key locations, through which it sells products directly to customers. It introduced the RB Apollo shirt in 2008, and this product has been very successful. It has found wide acceptance among teenagers and preteens, which has considerably increased profit forecasts.

The company is facing stiff competition to some of its products. Increased competition has stalled the sales growth of these products, and the company is looking to increase returns by improving the efficiency of its operations and by introducing product innovations to grow sales momentum.

3.1.4 Currency

The company operates in more than one country and has business transactions in multiple currencies. The currencies used for its operations are as follows:

- U.S. dollar
- British pound
- Euro

The company reports the financial transactions in one currency, U.S. dollars, because the company is headquartered in the United States.

3.1.5 Case Study

Rich Bloom has been using SAP ERP Central Component (ECC) for recording its business transactions and SAP NetWeaver BW as its data warehousing system to analyze information and make decisions. Using these systems has helped the company manage its business efficiently.

However, Rich Bloom wants to streamline the planning function of the business. The company currently does not have a systematic planning process. The dynamic fashion world and stiff competition have made management look into improving its operational efficiency and creating a sound planning system. Management recognizes that having a good planning tool will help the company face competition and retain leadership in the market.

Rich Bloom wants to be able to compare plan data with actual data to see if the company is able to meet what it set out to accomplish in the first place. The company has been using Microsoft Excel spreadsheets to manage its planning process, but this method has proved inadequate in several respects. First, the Excel spreadsheet does not provide a unified tool for entering plans and monitoring the planning process, which has resulted in multiple versions of plans created on user desktops, making it difficult for the company to distinguish between the latest and older versions. Second, the process of consolidating plan data from different Excel spreadsheets takes considerable time, making the entire process inefficient. Third, Excel lacks security controls. These factors have prompted Rich Bloom to look at other tools for planning. Although users have become accustomed to planning using Excel, Rich Bloom is grappling with its limitations and with using it exclusively for planning.

Rich Bloom has decided to use the recently introduced SAP BPC for NetWeaver for planning. First, this tool has an Excel-based frontend for planning, with which users are familiar. Second, Rich Bloom wants to leverage the SAP NetWeaver BW environment, which uses enhanced star schema architecture to provide a superior user experience. Rich Bloom also believes that SAP BPC for NetWeaver is an advanced tool for planning that will provide a much needed edge in the difficult and competitive marketplace.

The company will use SAP BPC for NetWeaver to plan its profitability for 2012, using the actual sales and cost data for 2011 as the basis to plan for 2012. It will maintain an InfoCube in SAP NetWeaver BW that contains actual sales and cost data, and the initial plan data for 2012 will be created by copying the actual sales and cost data for 2011 into the plan version for 2012. Subsequently, users will be able to manually change the sales and cost data to reflect current market conditions.

Management will consolidate, review, and approve the plan data. After the plan data is approved, no further changes will be allowed.

3.1.6 Decision to Use SAP BPC for NetWeaver

Rich Bloom has decided to use SAP BPC for NetWeaver for the following reasons:

- It wants to reduce the cycle time required to complete the planning process.
- It wants to overcome the limitations of Excel as a stand-alone tool for planning. With Excel, data cannot be stored in a database, and the tool does not provide a robust method of securing the data.

- Users are familiar with Excel for planning and analysis. The SAP BPC for NetWeaver frontend is based on Excel, so the company can tap this existing skill set to successfully manage its planning process.
- SAP BPC for NetWeaver provides users with role-based access to the data. This enables Rich Bloom to provide the appropriate level of access to users, based on their role in the planning process.
- Rich Bloom believes it can leverage existing actual sales and cost data currently available in SAP NetWeaver BW as the basis for planning. Actual data can be directly loaded into SAP BPC and can be used for planning.
- The company wants to leverage the already existing SAP NetWeaver BW environment to provide enhanced reporting to SAP BPC users.
- The SAP EPM add-in for Excel provides the ability to report data from the different software tools that are included in the SAP EPM suite of products. This enables Rich Bloom to unify the data from different sources—including SAP BPC, SAP NetWeaver BW systems, and so on—into one common report.

You should now have an understanding of Rich Bloom's requirement to develop a planning model to project profitability using SAP BPC for NetWeaver. In the next section, we'll develop objects in the SAP NetWeaver BW environment that will be used to stage actual sales and cost data coming from the SAP ERP system before being loaded into SAP BPC for NetWeaver.

3.2 Building the Data Model in SAP NetWeaver BW

As previously discussed, Rich Bloom wants to plan its profitability for 2012 using actual sales and cost data from 2011. In this section, we'll discuss the development of the staging environment in SAP NetWeaver BW that will house the actual sales and cost data. This data will eventually be loaded into SAP BPC for NetWeaver to be used for planning.

Figure 3.1 displays the flow of data from SAP ERP to SAP NetWeaver BW and subsequently to SAP BPC for NetWeaver. A Sales InfoCube that will be used as one of the sources of data for planning for the future will be created in SAP NetWeaver BW to store the actual sales and cost data for the past four years. Market research will also be used to forecast future demand and make the necessary adjustments for future sales and costs. The data in the Sales InfoCube will be loaded from the SAP ERP system on a daily basis.

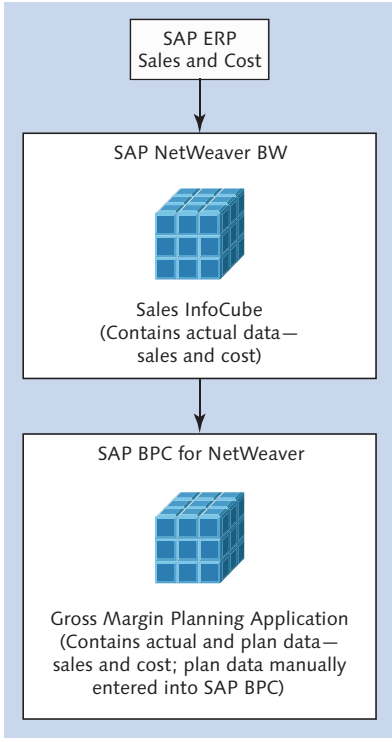


Figure 3.1 Flow of Data from SAP ERP to SAP BPC

We'll now build the objects necessary for storing the actual sales and cost data in SAP NetWeaver BW. The InfoObjects to be included in the Sales InfoCube are listed in Table 3.1, Table 3.2, and Table 3.3 and are SAP-delivered unless indicated by the words *custom InfoObject*. Table 3.4 also lists the key figures (quantitative measures) required for reporting.

Characteristics	Values
0COMP_CODE	20, 22, 23, 25, and 30 20—Rich Bloom, Inc., San Diego, CA, USA 22—Rich Bloom Corporate Inc., Philadelphia, PA, USA 23—Rich Bloom New Markets Inc., Houston, TX, USA 25—Rich Bloom Ltd., London, UK 30—Rich Bloom AG, Frankfurt, Germany

Table 3.1 Characteristics and Values

Characteristics	Values
0MATERIAL	CK2000, CK2001, CK2002, CK2003, CY7000, PRN CK2000—T-shirts CK2001—Shirts CK2002—Jackets CK2003—Apollo Shirt CY7000—Jeans PRN—None
0SALES_GRP	USE, UNE, UMW, USW, UWS, DE, UK, and SGN USE—US Southeast UNE—US Northeast UMW—US Midwest USW—US Southwest UWS—US West DE—Germany UK—UK SGN—None
ZACCOUNT (Custom Characteristic)	410000, 510000, 520000, 530000, 540000, 5500000, 560000, G4_GM, G4_NI, G4_PTI, and G5_OPEX 410000—Revenue 510000—Cost of Goods Sold 520000—Selling Expense 530000—General Expense 540000—Admin Expense 550000—Depreciation Expense 560000—Tax Expense G4_GM—Gross Margin G4_NI—Net Income G4_PTI—Pretax Income G5_OPEX—Operating Expense

Table 3.1 Characteristics and Values (Cont.)

Characteristics	Values
0CALMONTH	Calendar period for actual data
0CALYEAR	Calendar year for actual data

Table 3.2 Time Characteristics

Characteristics	Values
0CURRENCY	Used to store the currency associated with the amount value; used in key figure ZAMOUNT

Table 3.3 Unit Characteristics

Characteristics	Values
ZAMOUNT (Custom Key Figure)	Amount in local currency (key figure of type currency—associated with 0CURRENCY currency measure)

Table 3.4 Key Figures

3.2.1 Creating and Activating InfoObjects

Based on the requirements listed in the case study, we'll now start building the necessary objects. The InfoObjects listed in the previous tables are required for building the data model; most of them have a technical name that starts with 0. These are SAP-supplied InfoObjects. But before an SAP-delivered InfoObject can be used, it must be activated. We'll start by checking the status of InfoObjects and learn how to activate them if they are not yet active.

Checking the Status of an SAP-Supplied InfoObject

As illustrated in Figure 3.2, the following are the steps to check the status of an SAP-supplied InfoObject:

1. Open the Edit InfoObjects: Start window using Transaction RSD1.
2. Under Type, select Characteristic, and beside Version, select Active/Revised (Figure 3.2, ❶ and ❷).
3. In the text field next to InfoObject, enter the name of the InfoObject to check (Figure 3.2, ❸).
4. Click on the Display button (Figure 3.2, ❹).

If the InfoObject is not yet active, the following message is displayed in the status bar of the window: ENTER VALID CHARACTERISTIC ....

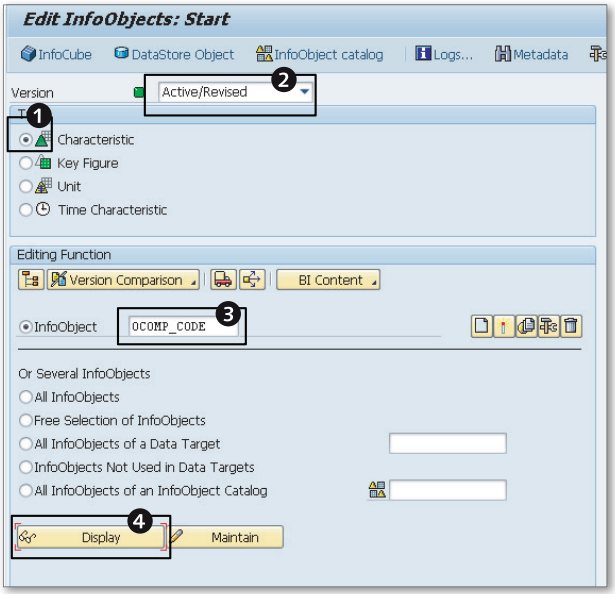


Figure 3.2 Checking the Status of an InfoObject

Activating an SAP-Supplied InfoObject

By default, none of the SAP-supplied InfoObjects are active, so we'll start by activating them. The process, illustrated in Figure 3.3, Figure 3.4, and Figure 3.5, is as follows:

1. Open the DATA WAREHOUSING WORKBENCH: BI CONTENT window using Transaction RSA1.
2. In the left pane, select OBJECT TYPES (Figure 3.3, ❶).
3. In the right pane, under GROUPING, select ONLY NECESSARY OBJECTS. For COLLECTION MODE, select COLLECT AUTOMATICALLY (Figure 3.3, ❷ and ❸).
4. Under INFOOBJECT in the middle pane, double-click on SELECT OBJECTS (Figure 3.3, ❹). This opens the INPUT HELP FOR METADATA dialog box.
5. In the INPUT HELP FOR METADATA dialog box, select all of the SAP-delivered InfoObjects that need to be activated for the case study, as outlined earlier in Table 3.1, Table 3.2, and Table 3.3.

6. When finished, click on Transfer Selections (Figure 3.4, ❺).
7. In the Data Warehousing Workbench: BI Content window, under Install, select Install to activate the InfoObject (Figure 3.5, ❻).

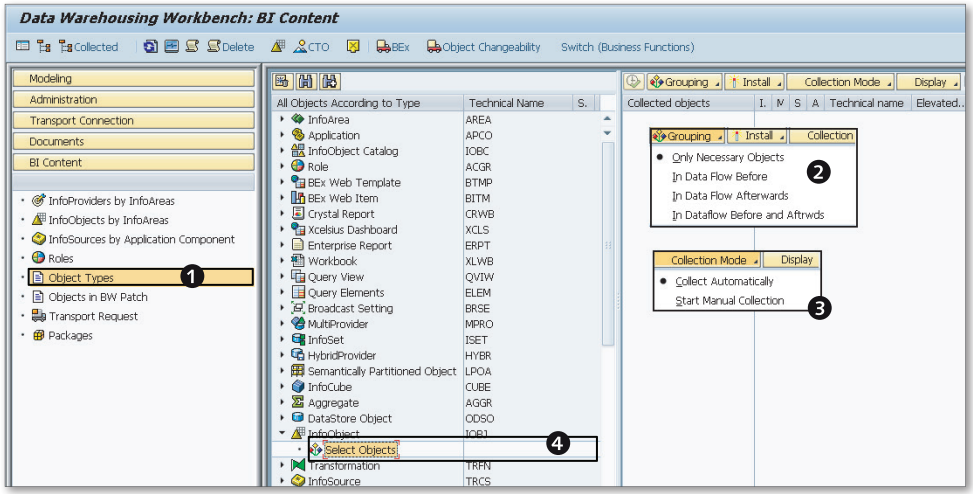


Figure 3.3 Settings to Activate an InfoObject

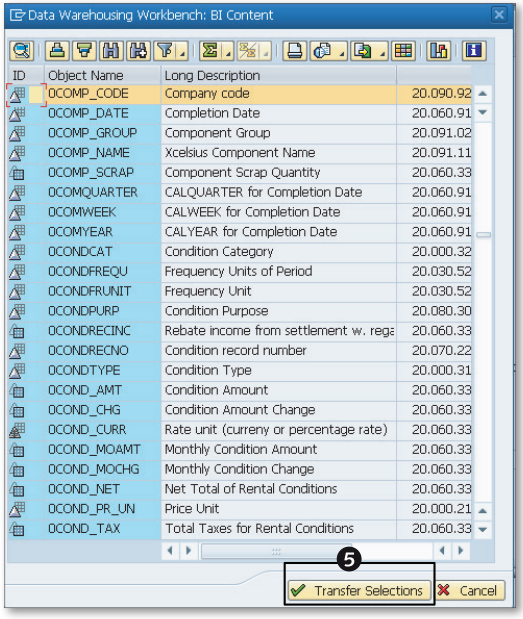


Figure 3.4 Selecting the InfoObject(s) to Activate

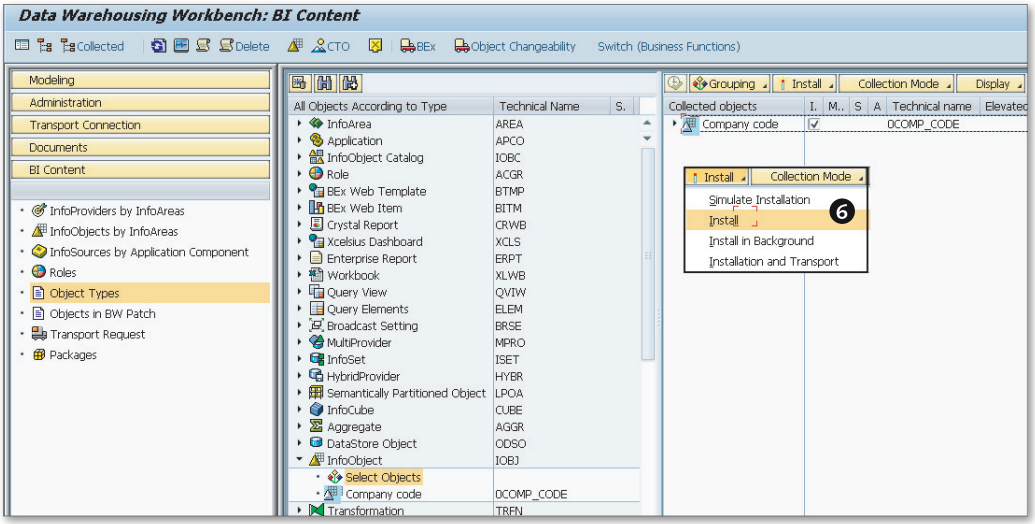


Figure 3.5 Activating an InfoObject

Note

Activating an InfoObject also activates all inactive InfoObjects that are dependent on this InfoObject.

After you're done activating the SAP-supplied InfoObjects, you'll need to create and activate the custom InfoObjects needed for the case study, as outlined in Table 3.1 and Table 3.4. Let's take a closer look at how to do this.

Creating and Activating a Custom Characteristic InfoObject

We'll start by creating the InfoObject ZACCOUNT (Account), as illustrated in Figure 3.6 and Figure 3.7. This InfoObject should be a characteristic InfoObject.

1. Open the EDIT INFOOBJECTS: START window using Transaction RSD1.
2. Under TYPE, select CHARACTERISTIC (Figure 3.6, ❶).
3. In the text field next to the INFOOBJECT option, enter the technical name of the InfoObject ("ZACCOUNT"), and click on the CREATE button (Figure 3.6, ❷ and ❸).

4. In the Create Characteristic dialog box, in the Long Description text field, enter a description ("Account"), and then click on the Enter button (Figure 3.6, ❹ and ❺).
5. On the General tab of the Create Characteristic <InfoObject_Name>: Detail window, enter the Short Description ("Account"), select the Data Type (CHAR - Character String), and enter the Length ("10") (Figure 3.7, ❻, ❼, and ❽).
6. No attributes are required for this InfoObject, so you can click on the Activate icon on the toolbar (Figure 3.7, ❾).

The ZACCOUNT InfoObject is now saved and activated.

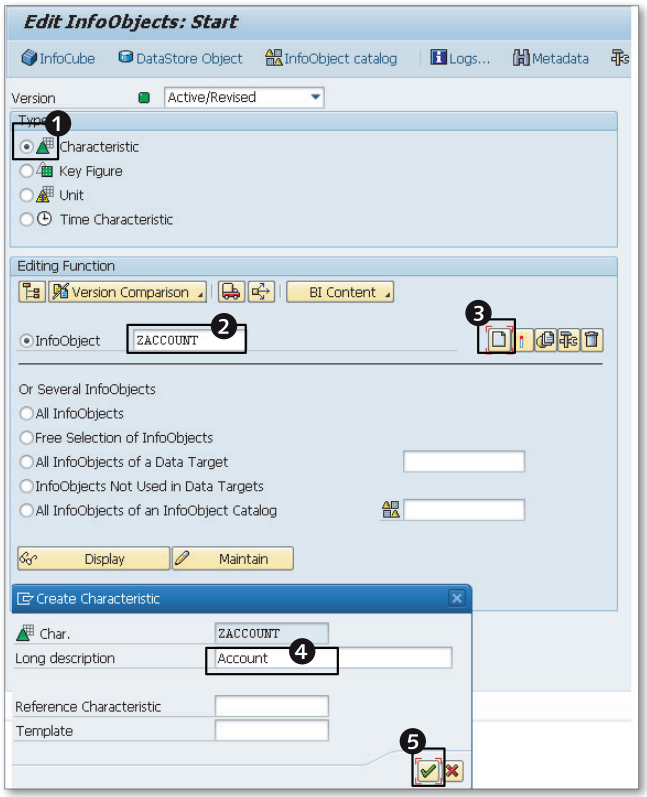


Figure 3.6 Settings to Create a New Characteristic InfoObject

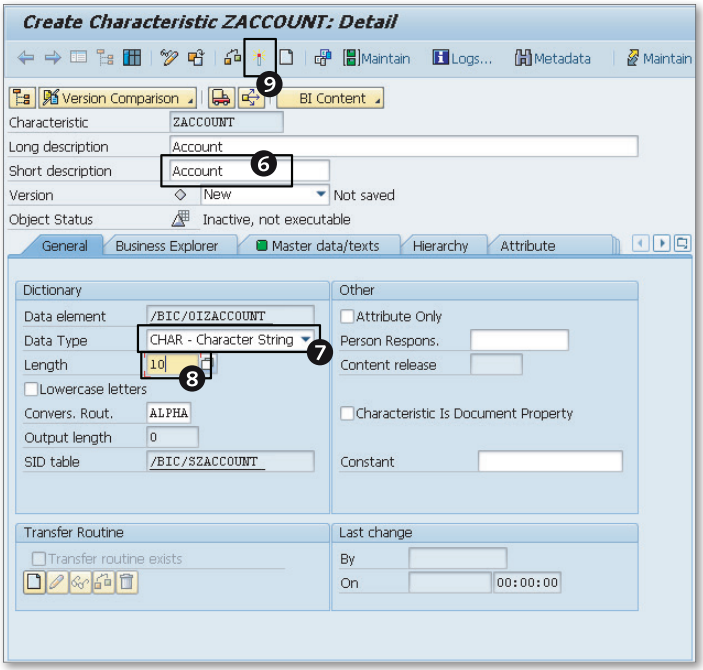


Figure 3.7 Activating a Custom InfoObject

Using a similar process, you'll now need to create and activate the remaining required custom InfoObject, ZAMOUNT. Let's get started.

Note

You should activate the SAP-delivered unit currency InfoObject, 0CURRENCY, before creating the custom key figure.

Creating a Custom Key Figure InfoObject

We'll now create a custom key figure InfoObject, as illustrated in Figure 3.8 and Figure 3.9. The technical name of the InfoObject is ZAMOUNT.

1. Open the Edit InfoObjects: Start window using Transaction RSD1.
2. Under Type, select Key Figure (Figure 3.8, ❶).
3. In the text field next to the InfoObject option, enter the technical name of the InfoObject ("ZAMOUNT"), and click on the Create button (Figure 3.8, ❷ and ❸).

4. In the Create Key Figure dialog box, in the Long Description text field, enter a description ("Amount"), and then click on the Enter button (Figure 3.8, ❹ and ❺).

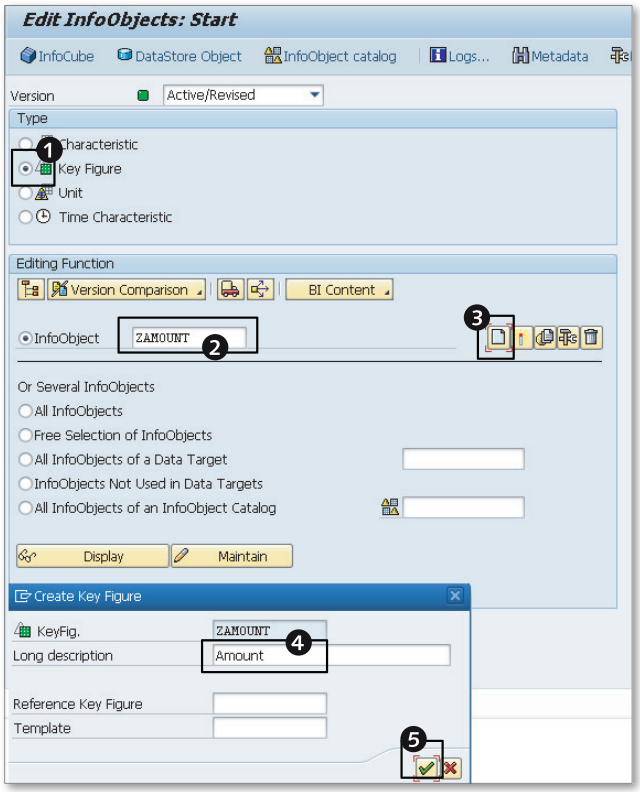


Figure 3.8 Settings to Create a New Custom Key Figure InfoObject

5. In the Create Key Figure <InfoObject_Name>: Details window, enter a Short Description ("Amount") (Figure 3.9, ❻).
6. On the Type/unit tab, select Amount, and then select CURR—Currency field, stored as DEC (Figure 3.9, ❼ and ❽).
7. Per the requirements, the ZAMOUNT InfoObject should be attached to the 0CURRENCY InfoObject. So under Currency/unit of measure, select 0CURRENCY (Figure 3.9, ❾).

8. Click on ACTIVATE (Figure 3.9, ⑩). The ZAMOUNT InfoObject is now saved and activated.

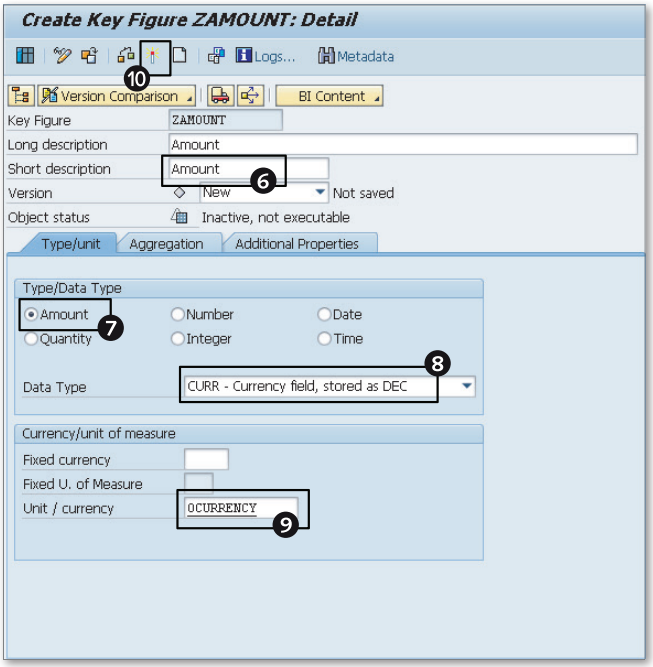


Figure 3.9 Settings to Create a New Custom Key Figure InfoObject (ZAMOUNT)

Populating Master Data

The next step in the process is to load master data for InfoObjects with the values listed in the reference table for the characteristic values (Table 3.1). As we have mentioned before, master data can be populated from a variety of sources: an SAP R/3 system, a flat-file system, an Oracle database system using DB Connect, and so on.

In this example, because there are only a few records in the master data table, we'll manually enter the values for the characteristics, but normally, the master data would be extracted and loaded from a source system.

We now manually enter the data for the Company Code InfoObject (0COMP_CODE), as illustrated in Figure 3.10 and Figure 3.11.

1. Open the Edit InfoObjects: Start window using Transaction RSD1.
2. Specify the InfoObject by entering its name ("0COMP_CODE") or selecting it from the dropdown list. Then click on the Display button.
3. In the Display Characteristic <InfoObject_Name>: Details window, click on Maintain to create or modify data for this InfoObject (Figure 3.10, ❶).
4. In the Characteristic <InfoObject_Name> – Maintain Master Data: Selection window, click on the Execute button (Figure 3.10, ❷).
5. In the Characteristic <InfoObject_Name> – Maintain Master Data: List window, click on Create to create a new record (Figure 3.11, ❸). Enter a value for the Company Code InfoObject ("20"), as listed in Table 3.1.
6. Click on the Enter button (Figure 3.11, ❹), and then click on the Save button.

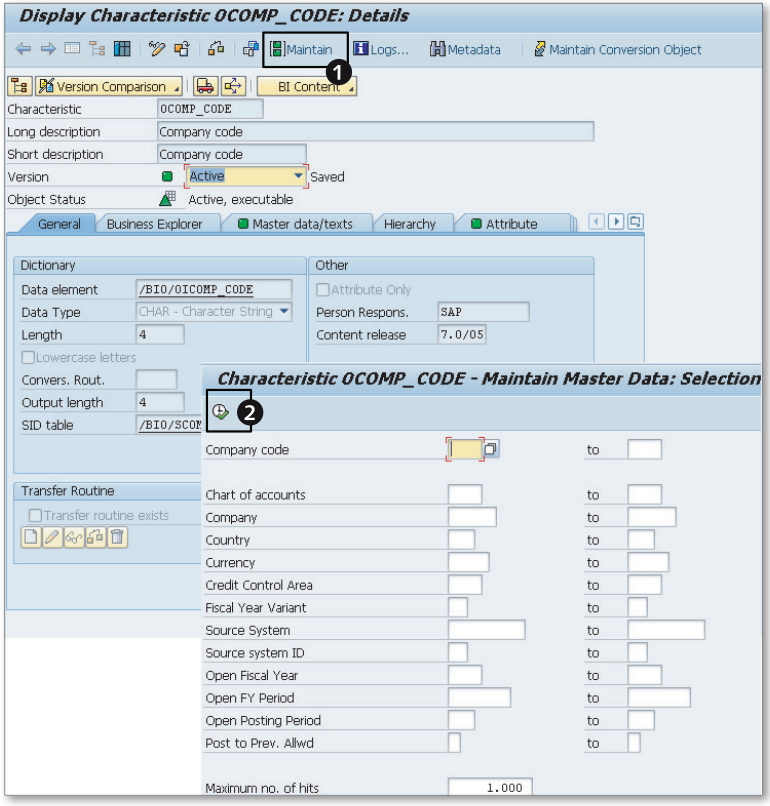


Figure 3.10 Manually Entering Data for a Characteristic InfoObject

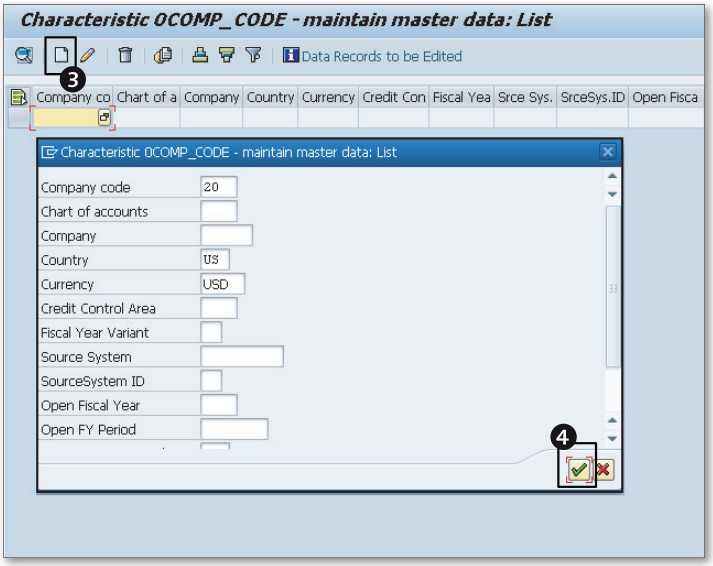


Figure 3.11 Manually Inserting a New Record for an InfoObject

Follow the steps outlined here to insert master data for the other characteristics listed in Table 3.1.

Applying the Hierarchy/Attribute Change Process

When you create a new record in the master data table, no action is necessary for the changes to become effective. But any changes to an existing record require that you run the apply hierarchy/attribute change process, as follows and as illustrated in Figure 3.12.

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Tools • Apply Hierarchy/Attribute Change from the menu.
3. In the Execute Hierarchy/Attribute Changes for Reporting window, click on InfoObject List to identify the characteristic InfoObjects that have undergone a change since the last change run (Figure 3.12, ❶).
4. Select the InfoObjects for which the changes to data should apply, and click on Save (Figure 3.12, ❷ and ❸). InfoObjects that are displayed here but are not selected for the change run will not reflect the latest changes made to the data when reporting.

5. After you click on Save, the system automatically assigns a job name, and you're taken back to the Execute Hierarchy/Attribute Changes for Reporting window.
6. Click on the Execute button (Figure 3.12, ❹).
7. Click on the Refresh button to confirm that the changes have been activated. The Change Status is displayed at the top of the table (Figure 3.12, ❺).

You can schedule the apply hierarchy/attribute change process from this location to run at selected times. You can also schedule it based on an event by clicking on the Selection button (Figure 3.12, ❻) in the Execute Hierarchy/Attribute Changes for Reporting window and then making the appropriate selections.

Subsequent events can be triggered following successful or unsuccessful completions of the change run.

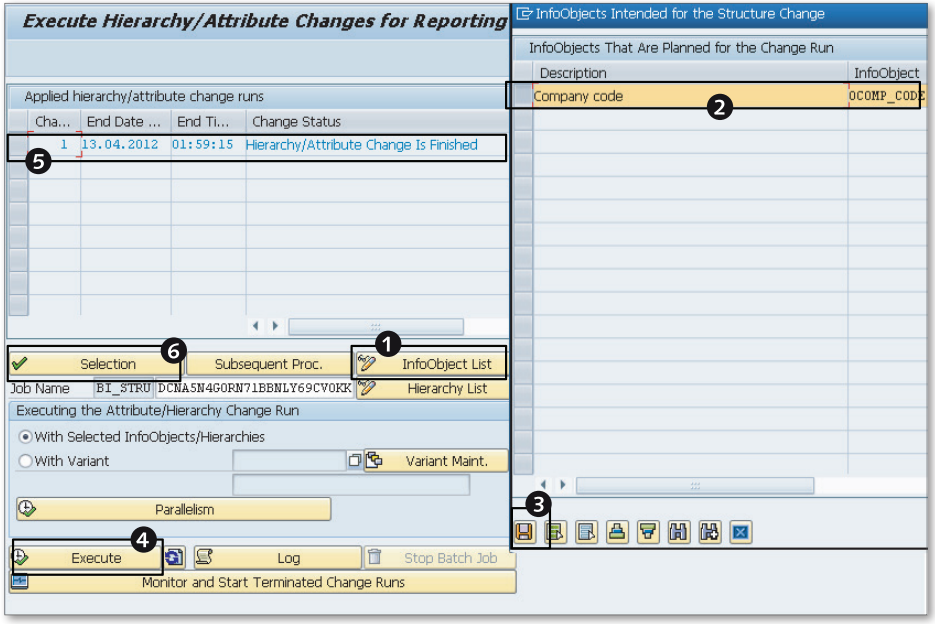


Figure 3.12 Manually Executing the Attribute Change Run Process

3.2.2 Creating an InfoArea

Now that you've created the InfoObjects, the next step is to create an InfoArea that will later contain InfoCubes. An InfoArea lets you group InfoProviders together.

The following are the steps to create an InfoArea, as illustrated in Figure 3.13:

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Right-click on InfoProvider, and select Create InfoArea (Figure 3.13, ❶).
4. In the Create InfoArea section, in the InfoArea text field, enter the technical name for the InfoArea ("ZSALES"). In the Long Description field, enter a description ("Sales Management") (Figure 3.13, ❷).
5. Click on Enter (Figure 3.13, ❸).

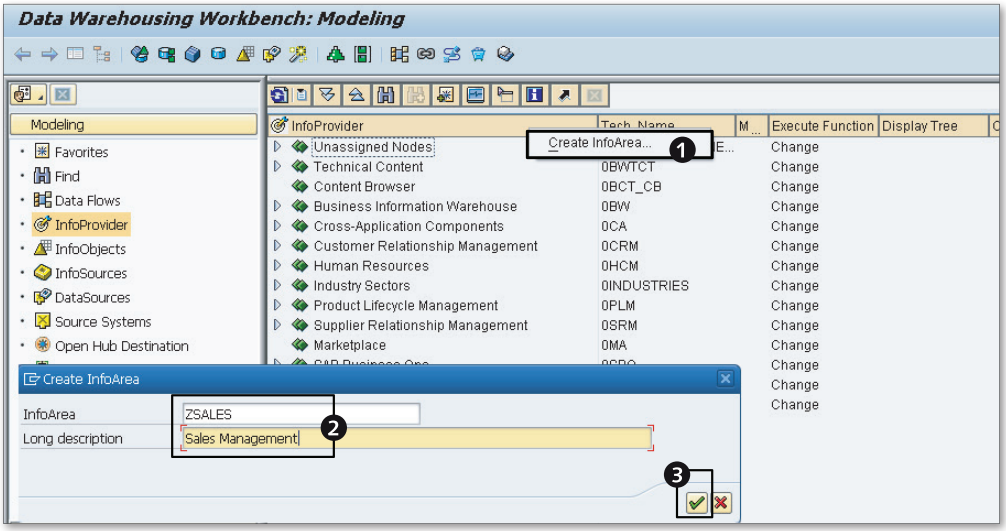


Figure 3.13 Create an InfoArea

3.2.3 Creating a Sales InfoCube

Now that you've created an InfoArea, the next step is to create the Sales InfoCubes under it, as shown in Table 3.5.

InfoCube	Description
ZSLSACT	Sales Actual InfoCube

Table 3.5 InfoCube for Storing Actual Sales Data

This section explains how to create the Sales InfoCube based on the requirements listed earlier in Table 3.1 through Table 3.4. Table 3.6 and Table 3.7 show the dimensions and key figures used in the Sales InfoCube.

Dimensions	Characteristics
Company	Company Code (0COMP_CODE)
Material	Material (0MATERIAL)
Sales Group	Sales Group (0SALES_GRP)
Account	Account (ZACCOUNT)
Time	Calendar Month (0CALMONTH) Calendar Year (0CALYEAR)
Unit	Currency (0CURRENCY)

Table 3.6 Dimensions for the Sales InfoCube

Key Figures	Description
ZAMOUNT	Amount in Local Currency

Table 3.7 Key Figures for the Sales InfoCube

The following are the steps and concepts you should follow to create the Sales InfoCube. The discussion includes creating dimensions, assigning characteristic InfoObjects to dimensions, and assigning key figures to the key figure folder. This process is illustrated in Figure 3.14 through Figure 3.18. Let's get started.

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Right-click on the Sales Management InfoArea you created earlier, and select Create InfoCube from the context menu (see Figure 3.14, ❶).
4. The Edit InfoCube window appears (Figure 3.15). Enter the technical name of the InfoCube ("ZSLSACT"), and the long description ("Sales actual") into the text fields next to InfoCube. Make sure you've selected Standard InfoCube in the InfoProvider Type area. Then click on Enter (Figure 3.15, ❷, ❸, and ❹).

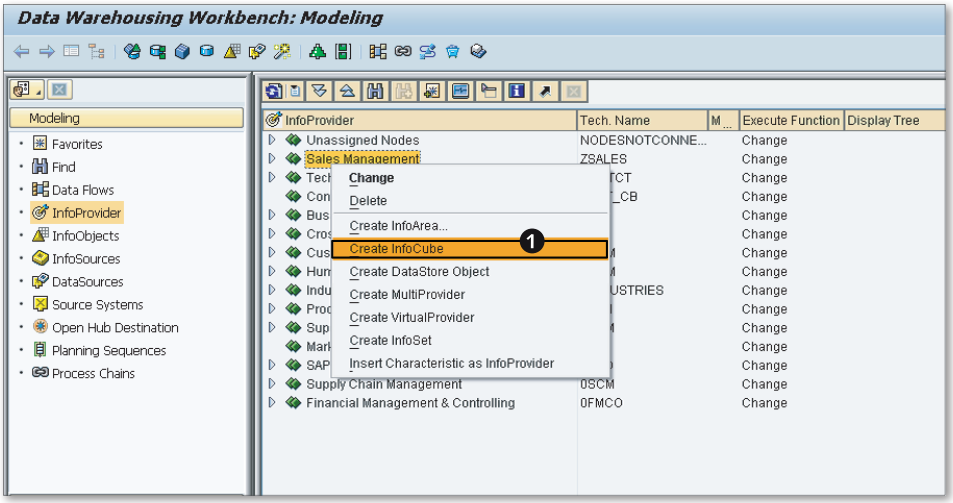


Figure 3.14 Creating an InfoCube—Part A

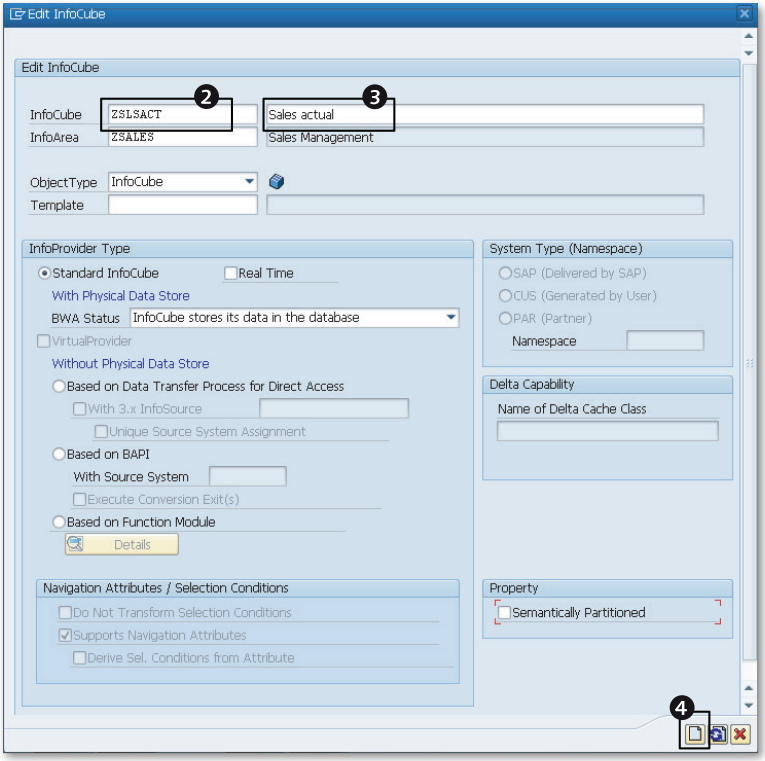


Figure 3.15 Create InfoCube—Part B

5. The Edit InfoCube window appears (Figure 3.16). Notice that the system automatically creates the data package, time, and unit dimensions.
6. Create a new dimension by right-clicking on Dimensions and selecting Create New Dimensions (Figure 3.16, ❺). The technical ID of the dimension is automatically assigned and cannot be modified, but the description of the dimension *can* be changed.
7. In the Create Dimensions window, enter a description for the dimension ("Account") (see Figure 3.16, ❻).
8. Click on the Create button (Figure 3.16, ❼). This creates the dimension and then opens the Create Dimensions window again.
9. Enter the description for the next dimension ("Company").

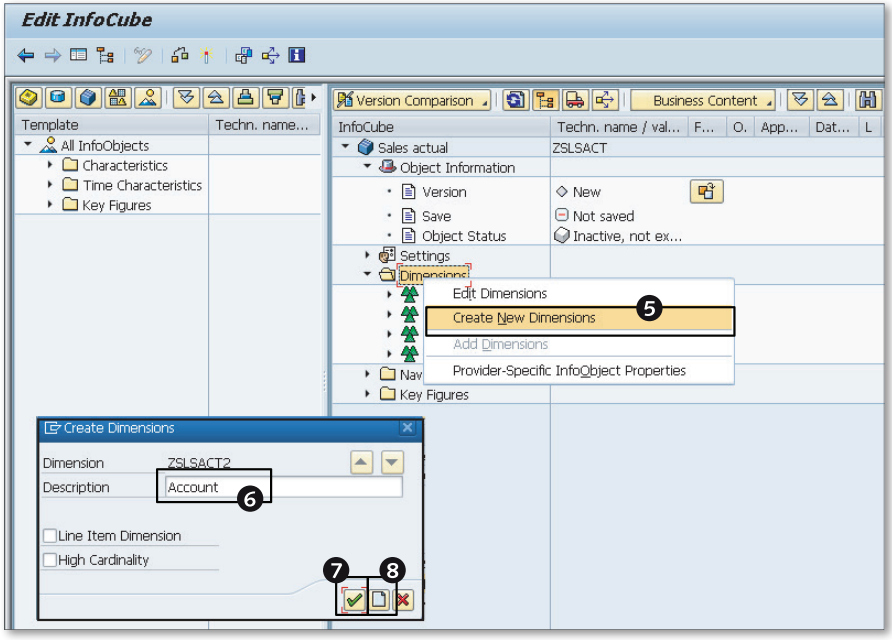


Figure 3.16 Create InfoCube—Part C

10. Repeat steps 8 and 9 to create the remaining dimensions for the Sales InfoCube that are specified in Table 3.6—Material, Sales Group, and Account.
11. Click on Enter when you're finished (Figure 3.16, ❽).

Note

You can create up to 13 user-defined dimensions in an InfoCube.

You've now created the dimensions needed for the Sales InfoCube. The next task is to assign characteristic InfoObjects to dimensions and key figure InfoObjects to the KEY FIGURES folder. Similar characteristics are grouped under one dimension. You can directly assign an InfoObject to a dimension or the KEY FIGURES folder (using the INFOOBJECT DIRECT INPUT option), as we'll show in the following steps:

1. Right-click on the DIMENSIONS or KEY FIGURES folder, and select INFOOBJECT DIRECT INPUT (Figure 3.17, ❾).

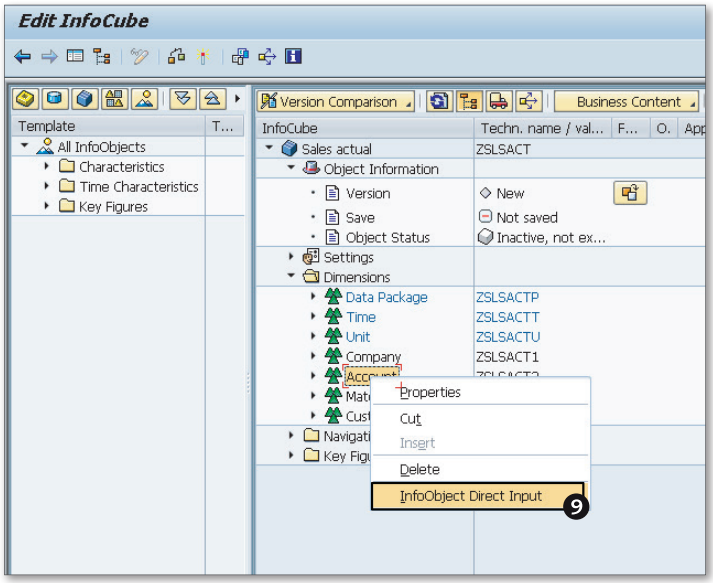


Figure 3.17 Create InfoCube—Part D

2. In the INSERT INFOOBJECTS window, select the InfoObjects you want to include in the dimension (Figure 3.18, ❿ and ⓫) or the key figures you want to include in the KEY FIGURES folder (not illustrated), and click on ENTER.

   Assign the following InfoObjects to the dimensions:

   - Dimension 1 (Company): 0COMP_CODE
   - Dimension 2 (Account): ZACCOUNT

- Dimension 3 (Material): 0MATERIAL
- Dimension 4 (Sales Group): 0SALESGRP
- Time Dimension: 0CALYEAR and 0CALMONTH

Assign the following key figure to the KEY FIGURES folder:

- ZAMOUNT

For key figures that are of the data types Quantity or Currency, the corresponding unit/currency measure associated with the key figure is automatically added to the unit dimension.

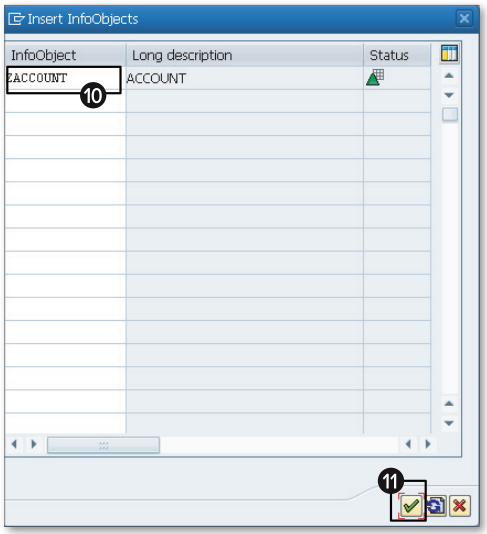


Figure 3.18 Create InfoCube—Part E

Note

There is also another way to include InfoObjects in a DIMENSIONS or KEY FIGURES folder. If the InfoObjects required for the InfoCube are in an InfoSource, DataStore, InfoCube, or InfoObject catalog, select the corresponding template in the left pane of the EDIT INFOCUBE window. Here, you can also select the INFOOBJECTS ALL template. This option lets you display all characteristic and key figure InfoObjects. Drag an InfoObject from the template to the preferred DIMENSIONS or KEY FIGURES folder.

The InfoCube is now ready to be saved and activated. Click on ACTIVATE to activate the InfoCube (Figure 3.19, ⓬).

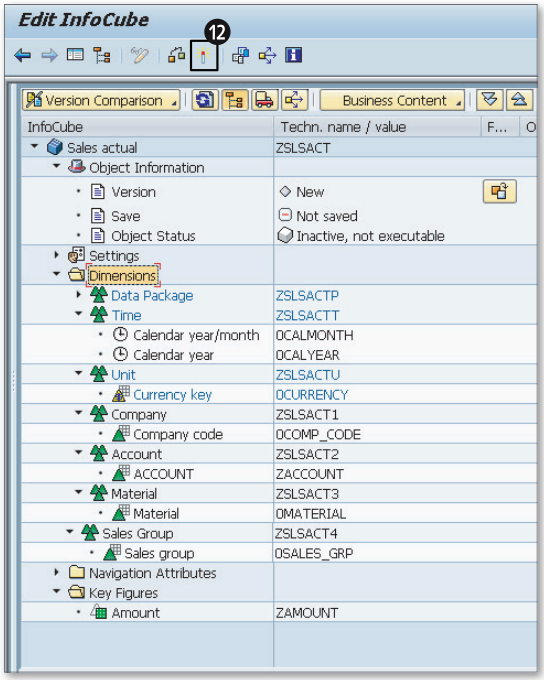


Figure 3.19 Create InfoCube—Part F

We have now created the InfoCube to store actual sales and cost data in the SAP NetWeaver BW system. We'll discuss how to load data into this InfoCube in Chapter 4. In the next section, we'll configure the objects required for this case study in the SAP BPC system.

3.3 Building Models in SAP BPC

In this section, we'll teach you how to configure a model in SAP BPC. This model will contain actual and plan data related to sales and costs. Rich Bloom requires the following pieces of data in the model to be able to plan effectively:

- Account
  - Account ID
  - Description
  - Account type (income account, expense account, etc.)

  - Rate type (average, year end, etc., to indicate the rate to use for currency translation)
- Entity
  - Entity ID (the company ID)
  - Description
  - Currency (the currency the company uses)
- Category
  - Category ID (denotes the nature of data—actual, budget, etc.)
  - Description
  - Year
- Time
  - Time ID
  - Description
- Currency
  - Currency ID (the currency associated with the transaction)
  - Description
  - Reporting (indicator used for currency translation)
- Product
  - Product ID
  - Description
- Sales Group
  - Sales Group ID
  - Description
- Audit
- Quantitative measure (amount/quantity)—also called SIGNEDDATA

We must create dimensions for all of the characteristics listed. A *dimension* is a type of object that is used to build a model. It represents the master data associated with any characteristic. For example, the Entity dimension represents the different company codes and their descriptions. The key to any dimension is referred to as an ID, and each record that the ID identifies in a dimension is called a *dimension*

member. In addition, *properties* can be described to provide meaning to the dimension member.

The SIGNEDDATA field is automatically available in all models to store quantitative information. This field contains either quantity or amount information. A model has only this field to store quantitative data; the Account dimension qualifies the data in the SIGNEDDATA field. So you can see how SAP BPC uses the account-based data model, where there is only one key figure in a model that is qualified by the information in the Account dimension.

For this case study, we'll use two models: one to store actual and plan data, and the other to maintain currency exchange rates. Because Rich Bloom does business in more than one country and uses different currencies, we'll use a rate model that can convert amounts from the local currency to the reporting currency.

Let's review the steps necessary to create a model in SAP BPC.

3.3.1 Accessing the SAP BPC System

The discussion about the architecture of SAP BPC for NetWeaver in Chapter 2 should have given you an idea about the different components that are included in this software and how they communicate with each other. A significant advantage in using this version of BPC is that a Web Client can be used for configuring the models. This replaces the Admin Console that was required to be installed in client desktops in the previous versions of the software. The Web Client is a thin client and can be accessed using a URL from Internet Explorer (IE). Developers use this tool to configure and maintain models.

Let's see how to access the SAP BPC system from the Web Client. The following steps explain the process for logging in to the SAP BPC via the Web Client:

1. You can directly enter the URL in the IE browser to access the SAP BPC system configured for your organization. The SAP BPC administrator or Basis administrator in your organization can provide the URL. You will have to request a user ID and password to access the system. The user ID is created in the SAP NetWeaver BW system with the necessary rights to access the Web Client and environments in SAP BPC. Consider this example of a URL to access SAP BPC: *http://rbus.richbloom.com:8000/sap/bpc/web*.

2. After entering the URL in the IE browser, press the Enter button. You will be presented with a dialog box to enter your credentials to access the system (Figure 3.20).

Figure 3.20 Accessing the SAP BPC Web Client—Part A

3. After entering your user ID and password, click on the Log On button. Your login information will be authenticated, and based on the access granted to your user ID, you will be presented with the screen shown in Figure 3.21. The Home workspace includes the components that you are authorized to access. The navigation pane in the left includes the components to configure the system in SAP BPC (Figure 3.21, ❶). Some of the components include views to enable detailed configuration for that area (Figure 3.21, ❷, ❸, and ❹). The following is a brief description of the components and views shown in the navigation pane:

   Start Page: This component is opened by default when you log in to BPC. It provides the options to display favorites, create a web report, input a form, create a new workspace, launch the EPM add-ins, and open the Planning and Consolidation administrator in a new workspace inside the Web Client.

   Activities: This component provides the interface to maintain the business process flow (BPF) instances. In general, BPF enables users to follow a sequence of steps toward fulfilling a planning or consolidation process. We will discuss BPFs in detail in Chapter 8.

Process Monitor: This component provides the interface to monitor the progress of BPF instances and display the status.

Library: This component enables you to share workspaces, web reports and input forms, Dashboards, and Books, as well as maintain your favorites.

Documents: This component provides the options to manage documents.

Consolidation Central: This component contains four different views:

- Consolidation Monitor: The Consolidation Monitor view displays the status of the consolidated financial data of an organization and provides an efficient process for consolidating data.
- Controls Monitor: The Controls view is available to enforce rules and validate data inside SAP BPC. This is a new feature available in EPM 10.0 and replaces the validation business rules functionality that was available in prior versions.
- Journals: Journals provide the interface to configure and manage adjustments. This interface is used to create journal entries.
- Ownership Manager: The Ownership Manager is available for an organization that is required to consolidate the data of subsidiaries with different ownership structures. It provides the ability to create a hierarchy of the organization with ownership data and aids in reporting the financial results of a company accurately and in compliance with the statutory requirements of reporting financial data.

Audit: This view provides the interface to configure and view system activity.

4. The name of the environment to which you are logged in is shown in Figure 3.21 (❺). Click on the Planning And Consolidation Administration link in the Launch area (Figure 3.21, ❻).
5. A new Administration workspace opens and includes the options to administer tasks (Figure 3.22). The navigation pane in the left consists of distinct domains. Under each domain, there are distinct views to perform configuration. The following domains are available:
   - Dimensions and Models: The views inside this domain provide the ability to maintain dimensions and models.
   - Security: Some scenarios require restricted access based on the role of the user in the organization. The views listed under this domain enable defining the security at a detailed level.

- Rules: Different types of rules can be configured under this domain. Member formulas are used to dynamically calculate values for dimension members when executing reports. Logic script allows you to develop program logic to meet specific business requirement. Business Rules supports functionality for various standard functions that include account transformation, carry forward balance, currency translation, consolidation, and so on. The Controls view is designed to validate the data stored in SAP BPC and to provide for reliable analysis of data.
- Business Process Flows: The views listed under this domain enable maintenance of BPF templates and instances.
- Features: The options available here provide the ability to define work status to control the update of data, maintain journal templates, set up context defaults for models inside the environment, define drill-through to pass parameters from SAP BPC reports to SAP NetWeaver BW reports and URL links, maintain the document types that can be maintained inside the environment, and define the audit settings for environments and models.

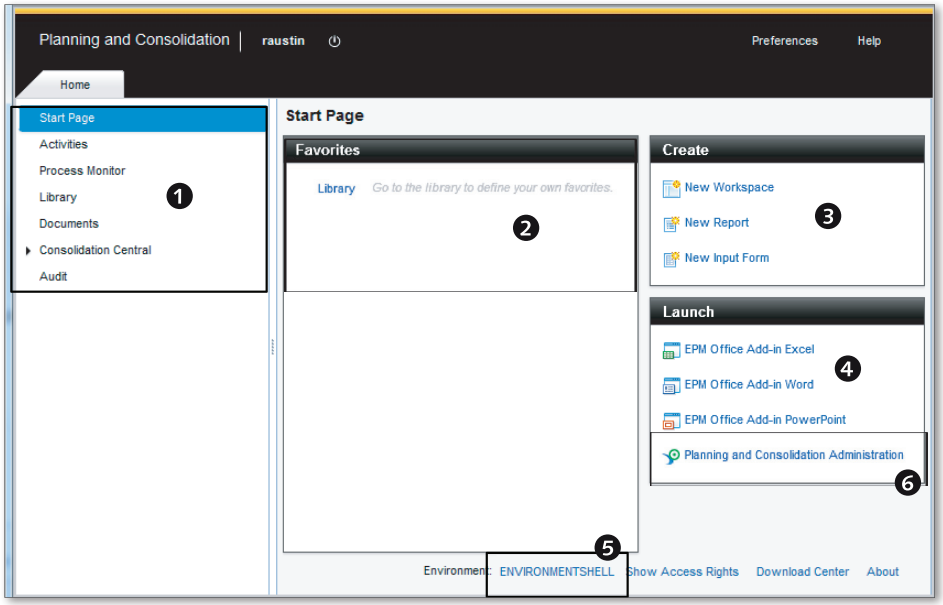


Figure 3.21 Accessing the SAP BPC Web Client—Part B

6. To see the list of environments to which you have been provided access, click on Manage All Environments (Figure 3.22, ❼). If you do not see a particular environment, you do not have authorization to access it.
7. To change from the current environment into a different environment, click on Change Environment (Figure 3.22, ❽).

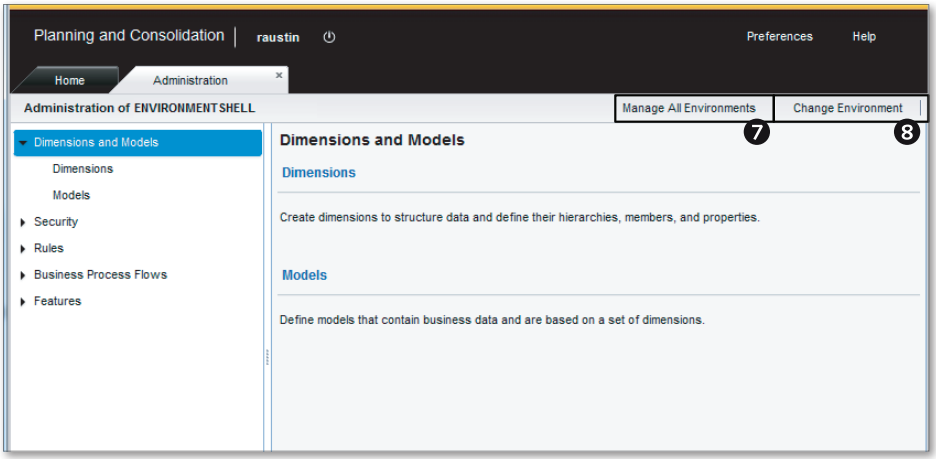


Figure 3.22 Accessing the SAP BPC Web Client—Part C

You've now seen how to log in to the SAP BPC system from the Web Client and how to access an environment. Before we create a new environment, we'll discuss the key features of an environment and how to use them.

3.3.2 Environment

The first step in building a model in SAP BPC is creating an environment. In SAP BPC 7.x and 5.x versions, the environment was called an application set, and the model was referred to as the application. An environment is designed to contain the models for a given functional area; for example, one environment can be created exclusively for sales planning models, and another can be created for head-count planning models.

You cannot build a new environment from scratch; you always copy it from an existing environment. In this context, it is important to note that dimensions and dimensions members within an environment are not shared with other environments.

One of the steps in the installation of an SAP BPC system is the installation of an environment with the technical name ENVIRONMENTSHELL. This is the only environment that is available when the system is first created. This special environment comes with the standard dimensions and includes sample models for planning, consolidation, and currency exchange translation. We do not recommend making any changes to the ENVIRONMENTSHELL environment; it should only be used as a reference or basis to create a new environment. The dimensions and models included in ENVIRONMENTSHELL are loaded with sample data, so it serves as a good reference point to understand how models are built in SAP BPC.

When you create a new environment, the model definitions are copied from the source environment to the new environment. When you copy one environment to another, a checkbox is available that provides the option to copy model data from the source environment to the new environment. If this flag is not selected, only the model definition is copied. If this checkbox is selected, the data from the models in the source environment are also copied to the new environment. The following options are used to indicate the type of data you want to copy from the source environment to the new environment:

- Data in model
- Web documents
- Documents
- Journal templates

Note

When a new environment is created, the dimension members are automatically copied from the source environment.

Only a user with system administrator access can create and maintain environments. When a request for the creation of a new environment is initiated in the SAP BPC system, a background job is scheduled in the SAP NetWeaver BW system. The background job creates all of the necessary objects, including InfoObjects and InfoCubes for the new environments, based on the options selected in SAP BPC when initiating this request. You can also choose to have the data copied from the source environment to the new environment. We'll discuss the parameters that can be set for an environment in Chapter 7 when we discuss the web interface for SAP BPC.

So far, you've seen how to access the Web Client and log into the SAP BPC system to access an environment. We'll now discuss the process of creating a new environment.

Creating an Environment

Our case study requires us to build a model for profitability planning, which we do by starting to build a new environment using the ENVIRONMENTSHELL environment as the source. To create a new environment, proceed as follows:

1. Follow the steps listed in Section 3.3.1 to log in to the SAP BPC Web Client. Click on MANAGE ALL ENVIRONMENTS as shown earlier in Figure 3.22 (❼) to administer environments. Figure 3.23 lists the various options for managing the environments. Only the environments you have access to are displayed in the screen. Also, some of the buttons may be grayed out depending on your access level. To create a new environment that is a copy of the ENVIRONMENTSHELL environment, select the ENVIRONMENT SHELL environment as shown in Figure 3.23 (❶). Then click on COPY as shown in Figure 3.23 (❷).

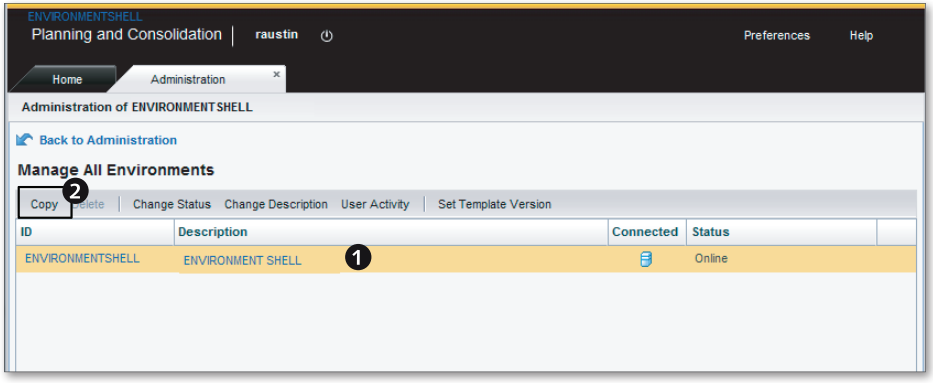


Figure 3.23 Creating an Environment—Part A

2. In the COPY AN ENVIRONMENT dialog box, enter the technical name ("ZRB_PR_PLAN") and description ("Profitability Planning") for the new environment (Figure 3.24, ❸ and ❹). Select the options you want to use for copying data from the source environment (Figure 3.24, ❺). When DATA IN MODELS is selected, the model data is also copied to the new environment. For our case study, we'll select this option. Then click on COPY (Figure 3.24, ❻).

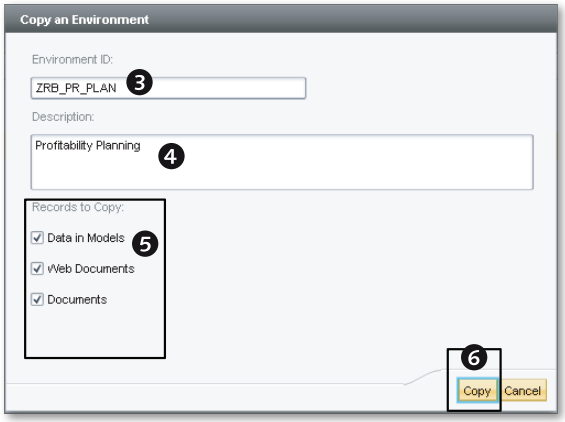


Figure 3.24 Creating an Environment—Part B

3. The SAP BPC system schedules a background job in the SAP NetWeaver BW system to create the new environment. The new screen indicates that the copy is in progress. The detailed information on the progress is viewed by clicking on SEE DETAILS (Figure 3.25).

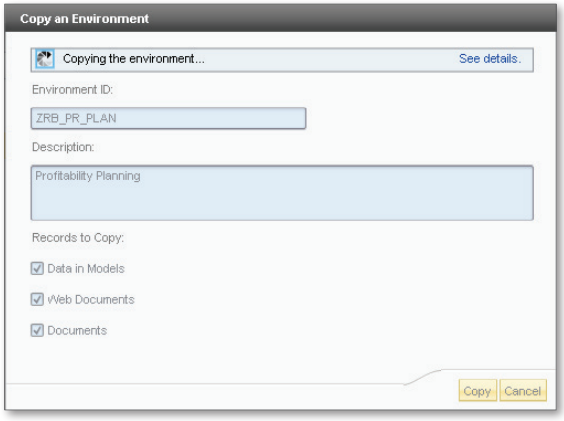


Figure 3.25 Creating an Environment—Part C

4. After the background job for creating the environment completes successfully, the new environment is listed in the environments available for access (Figure 3.26).

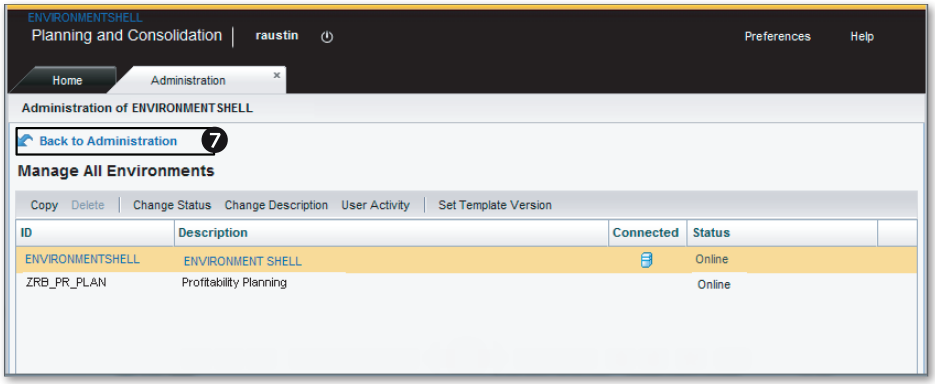


Figure 3.26 Creating an Environment—Part D

Even though you've created a new environment, the session information continues to show the ENVIRONMENTSHELL environment because you're connected to this environment. To access the new environment, ZRB_PR_PLAN, click on BACK TO ADMINISTRATION, as shown in Figure 3.26 (❼). This takes you back where you can now click on CHANGE ENVIRONMENT as shown earlier in Figure 3.22 to change the environment you want to connect to. This opens the screen shown in Figure 3.27, where you'll select an environment to connect to.

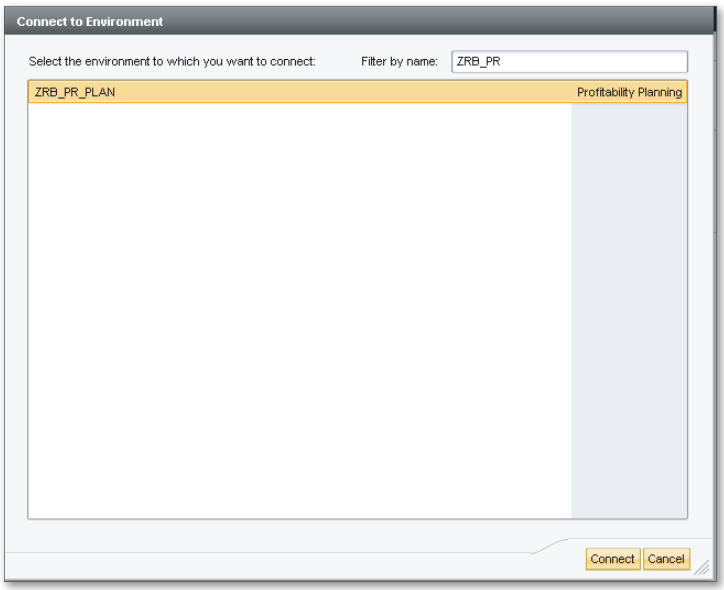


Figure 3.27 Connect to Environment—Part E

You should now understand how to create a new environment. We'll use the new environment we created for building the dimensions and models for our case study.

Managing the Environment

The following options are available for managing an environment:

- **Set availability of environment**
  You can set an environment to Not Available by selecting the Change Status option in the action pane. A user who is not a system administrator in SAP BPC cannot access any models in the environment when the status of an environment is set to Not Available. An information message can be added to provide more information to users about this status, which is particularly useful when maintenance on the environment is in progress. The message can provide an estimate of when the environment will be back and when users can log back on.
- **Set template version**
  SAP BPC lets you create and maintain standard templates specific to environments for reporting. When changes are made to the standard templates, you can refresh them immediately. To do this, update the client cache and set a higher version by selecting Set Template Version in the action pane.
- **Change description**
  The description associated with an environment can be changed by clicking the Change Description button.
- **Delete environment**
  You can delete an environment by selecting the Delete menu option. When an environment is deleted, all of the objects under the environment are also deleted. Exercise caution when deleting an environment—it cannot be recovered.
- **Monitor user activity**
  You can monitor the activity associated with an environment by selecting the User Activity option in the action pane.

You should now have an understanding of how to create an environment and of the options associated with managing the environment. Next, we'll discuss how to use dimensions in SAP BPC.

3.3.3 Dimensions

Dimensions are the building blocks used in creating an SAP BPC model; they describe the elements of the business. Examples of dimensions are Account, Entity, Time, Sales Group, and Product. Note the following points related to dimensions:

- Dimensions are always created within an environment and can only be used in that environment. A dimension created in an environment cannot be shared or used in another environment.
- The name of a dimension can be a maximum of 16 characters.
- Dimensions can include properties. For example, the Account dimension can include a property called Account Type. Properties can be used as filter criteria for selections in reports. The length of a property can be set to any number. When the length of a property exceeds 60 characters, it is not available for selection when creating reports.
- The properties for a dimension can be classified under three distinct types:
  - User-defined: These are properties the user defines for a dimension. For example, the Product dimension can include a property called Brand that indicates the product brand.
  - Reserved: These are system-defined properties that are delivered with dimensions. For example, the Scaling and Format properties are system-supplied properties for the Account dimension.
  - Hidden: These are properties that are not visible but that the SAP BPC system uses internally.
- SAP BPC provides standard dimensions for planning and consolidation functions. Table 3.8 lists the types of dimensions available in SAP BPC.

Dimension Type	Examples
Category	Plan version, actual
Account	GL accounts or codes that define data (account-based model)
Time	Calendar months or fiscal periods
Entity	Company code

Table 3.8 Dimension Types

Dimension Type	Examples
Currency	Local or reporting currency
Audit	Manual, automatic
Inter Company	Trading partner
Sub Tables	Flow dimension—opening balance, movements, and closing balance
User Defined	Sales group, product

Table 3.8 Dimension Types (Cont.)

- The Account, Category, Entity, and Time dimensions are required in all models. A dimension type cannot be used more than once in a model; for example, a model cannot include more than one Entity type dimension.
- When you create a new dimension, the dimension type is associated with it. The default properties of the dimension type are available in the new dimension. You can then add custom properties to the dimension.
- You can copy a new dimension from an existing dimension. When you copy a dimension, all of the dimension members from the source dimension, along with its properties, are also copied to the new dimension.
- You can modify a dimension to either include new properties or delete existing properties.
- You can delete a dimension, provided it is not included in any of the models in the environment.
- In SAP BPC, when you create a dimension, the system creates an equivalent InfoObject with corresponding attributes in the SAP NetWeaver BW system.
- You can set security to restrict users' read and write access, based on specific values of a dimension. The security is set for each model. We'll explain how this is done when we discuss the security features in SAP BPC.
- Data in dimensions is referred to as *dimension members*. The key for the dimension is called the ID. The maximum length of the ID is 20 characters, and it is case-sensitive. You cannot rename the ID member value of a dimension once it is saved.
- You can maintain (add, delete, and change) dimension data in a dimension via the Web Client. An Excel-like interface is available to maintain data for a dimension.

- When you maintain dimension members, you must process the dimension to make the modified data available. When you request the Process dimension option, the SAP BPC system schedules an *attribute change run* for the corresponding InfoObject in the SAP NetWeaver BW system. This process activates the data and makes the modified data available for the SAP BPC models.
- A set of selected dimensions constitutes a model. When creating a model using another model as the source, you can select the required dimensions for the model.

Reference Dimensions

The purpose of *reference dimensions* is to support the validation of member values that you can enter in the property of a dimension. Table 3.9 details the properties of dimensions belonging to a particular type that are validated based on values of members in the reference dimension.

Dimension Type	Property Name in Dimension	Reference Dimension	Reference Attribute Name
A—Account	Rate Type	A—ACCOUNT	ID
C—Category	Year	T—Time	Year
E—Entity	Currency	R—Currency	ID
I—Inter Company	Entity	E—Entity	ID

Table 3.9 Reference Dimension

Hierarchies in Dimensions

Certain models require the use of a hierarchy. The data in dimensions such as Entity, Account, and Time are structured in hierarchies. For example, you may want to set up separate legal and management trees for entities. You can do this using a hierarchy, and SAP BPC lets you create any number of hierarchies for a dimension. You have the option to add a hierarchy when editing the structure of the dimension.

The rules for using hierarchies are as follows:

- The values in the hierarchy column should be valid dimension members as well.
- A parent in SAP BPC must have the same children if it is reused across multiple hierarchies.

Note

Currently, SAP BPC for NetWeaver does not support time-dependent hierarchies.

Dimension Formulas

Dimension formulas automate the process of calculations in SAP BPC. You may want to add transaction values in two accounts and report under a third account. You can create a dimension formula for this purpose. When a formula is created, the calculation is automatically affected for all models in the environment that contain the dimension. When this formula is used in reports, conflicts in the calculation of the formula could occur, for instance, when conflicting formulas exist in two different dimensions. In this case, SOLVEORDER determines the sequence of how the formula should be calculated. We will study dimension formulas in detail when discussing the development of business logic in Chapter 6.

Displaying Dimension Properties and Member Data

We'll now explain how to review the properties of a dimension and display the members of a dimension. First, let's review the properties and members for the ACCOUNT dimension.

1. In the Administration workspace, select Dimensions in the left pane, and then click on the ACCOUNT dimension (Figure 3.28, ❶). Then click on Edit Structure (Figure 3.28, ❷) to view the properties.
2. The properties for the ACCOUNT dimension are displayed (Figure 3.29). To add a new property, click on Add (Figure 3.29, ❸). To use formulas inside a dimension, the Allow Member Formulas box is checked (Figure 3.29, ❹). When this box is checked, a dimension member can be created as a formula in the Member Formulas view under the Rules domain. After reviewing the properties, click on the Close dimension (Figure 3.29, ❺) to return to the previous screen.
3. Click on Edit Members (Figure 3.28, ❻) to edit/display the Dimension Members. The member IDs and their descriptions are displayed (Figure 3.30). Member IDs are case-sensitive. If you have defined hierarchies when editing the structure, you can make assignments to the hierarchies here. After editing dimension members, click on Save and Process (Figure 3.30, 7) to activate the changes.

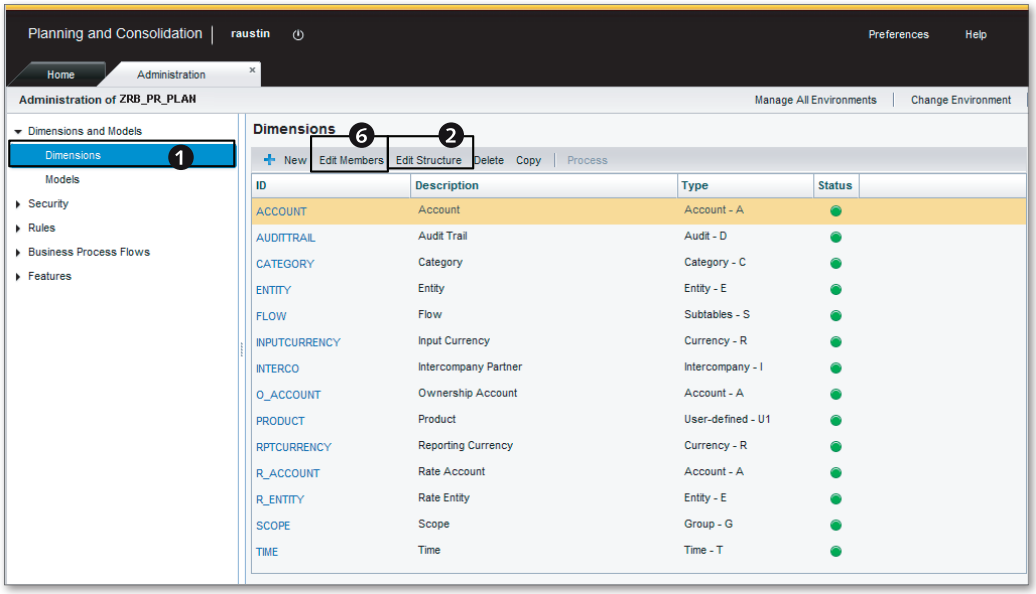


Figure 3.28 Displaying Dimension Properties and Member Data—Part A

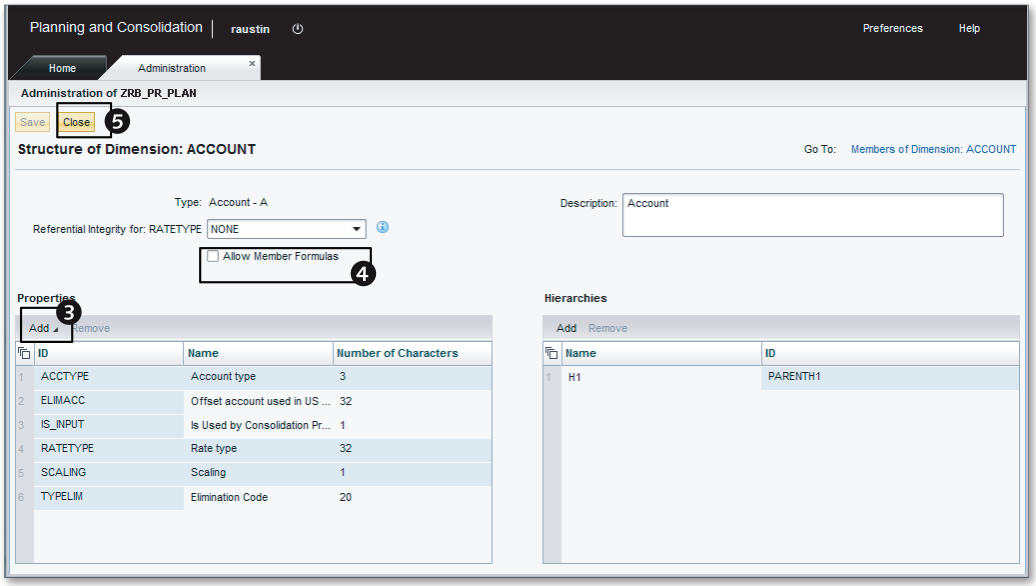
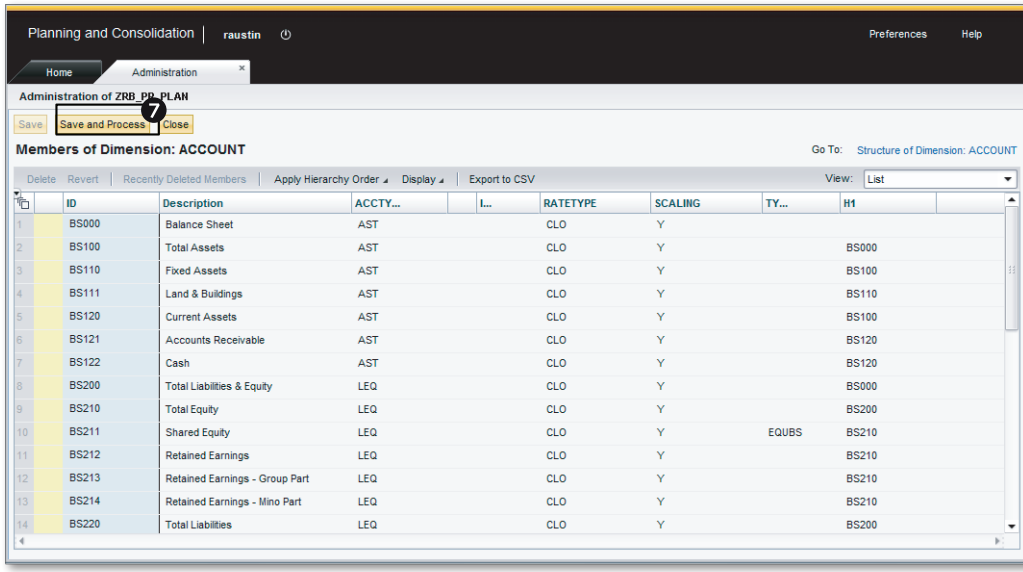


Figure 3.29 Displaying Dimension Properties and Member Data—Part B



	ID	Description	ACCTY...	I...	RATETYPE	SCALING	TY...	H1
1	BS000	Balance Sheet	AST		CLO	Y		
2	BS100	Total Assets	AST		CLO	Y		BS000
3	BS110	Fixed Assets	AST		CLO	Y		BS100
4	BS111	Land & Buildings	AST		CLO	Y		BS110
5	BS120	Current Assets	AST		CLO	Y		BS100
6	BS121	Accounts Receivable	AST		CLO	Y		BS120
7	BS122	Cash	AST		CLO	Y		BS120
8	BS200	Total Liabilities & Equity	LEQ		CLO	Y		BS000
9	BS210	Total Equity	LEQ		CLO	Y		BS200
10	BS211	Shared Equity	LEQ		CLO	Y	EQUBS	BS210
11	BS212	Retained Earnings	LEQ		CLO	Y		BS210
12	BS213	Retained Earnings - Group Part	LEQ		CLO	Y		BS210
13	BS214	Retained Earnings - Mino Part	LEQ		CLO	Y		BS210
14	BS220	Total Liabilities	LEQ		CLO	Y		BS200

Figure 3.30 Displaying Dimension Properties and Member Data—Part C

Copying a Dimension

We'll now explain how to create a new dimension by copying from an existing dimension. Our case study requires us to maintain the following dimensions for the planning model:

- RB_ACCOUNT
- RB_CATEGORY
- RB_ENTITY
- RB_PRODUCT
- RB_SALESGRP
- RB_AUDITTRAIL
- RPTCURRENCY
- TIME

We'll start by creating the RB_ACCOUNT dimension by copying it from the ACCOUNT dimension. When you copy a dimension, the members of the source dimension are copied to the new target dimension.

1. To copy a dimension, click on Dimensions in the left pane (refer to Figure 3.28, ❶). Select the ACCOUNT dimension, and click on the Copy option.
2. Select the dimension you want to use to copy to the new dimension. Enter "RB_ACCOUNT" as the technical name of the dimension (Figure 3.31, ❶). Enter "RB Accounts" as the description for the dimension (Figure 3.31, ❷). Click on Copy (Figure 3.31, ❸).
3. A status message box would be displayed that indicates that the dimension was successfully created, and the newly created dimension is shown under the list of dimensions in the center pane (Figure 3.32).

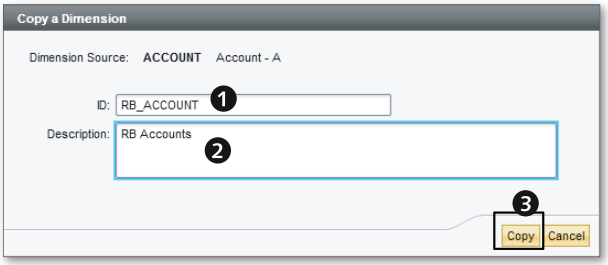


Figure 3.31 Copying a Dimension—Part A

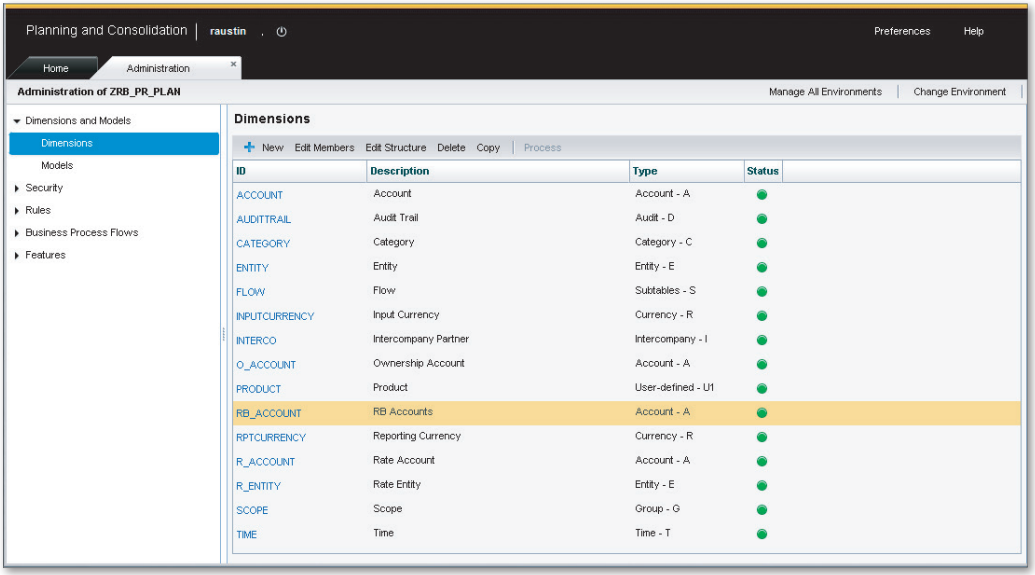


Figure 3.32 Copying a Dimension—Part B

Similarly, create the RB_CATEGORY, RB_AUDITTRAIL, and RB_ENTITY dimensions by copying from the Category, Audit Trail, and Entity dimensions. Because we're using the copy task, all of the dimension members from the source dimension are also copied to the new dimensions. If you want to create a new dimension without copying from an existing dimension, you can select the NEW option.

Creating a Custom Dimension

Our sample company requires product and sales group data for planning. These dimensions are not standard dimensions provided by SAP BPC; instead, we must create them as custom dimensions. We'll now create the product dimension as a custom dimension using the following steps:

1. To create a new dimension, click on DIMENSIONS in the left pane (refer to Figure 3.28, ❶). Select NEW from the list of menu options.
2. You're now prompted to enter the technical name and description of the dimension. Enter "RB_PRODUCT" for the technical name and "RB PRODUCTS" as the description (Figure 3.33, ❶ and ❷). Select USER-DEFINED – U as the dimension type from the dropdown list (Figure 3.33, ❸). Select CREATE (Figure 3.33, ❹).
3. The standard properties are automatically added for the dimension type (Figure 3.34, ❺). They are protected and cannot be changed or deleted. You can now add new properties as needed, although you do not need to add any new properties to this dimension for our case study. To create a new property, click on ADD (Figure 3.34, ❻). To add a new hierarchy, click on ADD in the HIERARCHIES area (Figure 3.34, ❼).
4. In the NEW HIERARCHY dialog box enter "PRODHIER1" as the hierarchy name, and click OK (Figure 3.34, ❽ and ❾). Click on SAVE to create the dimension (Figure 3.34, ❿).

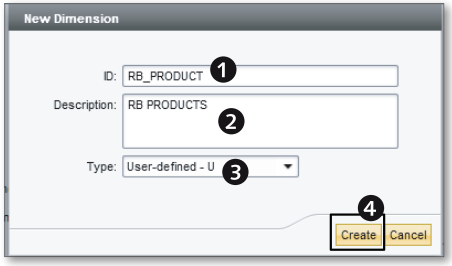


Figure 3.33 Creating a Custom Dimension—Part A

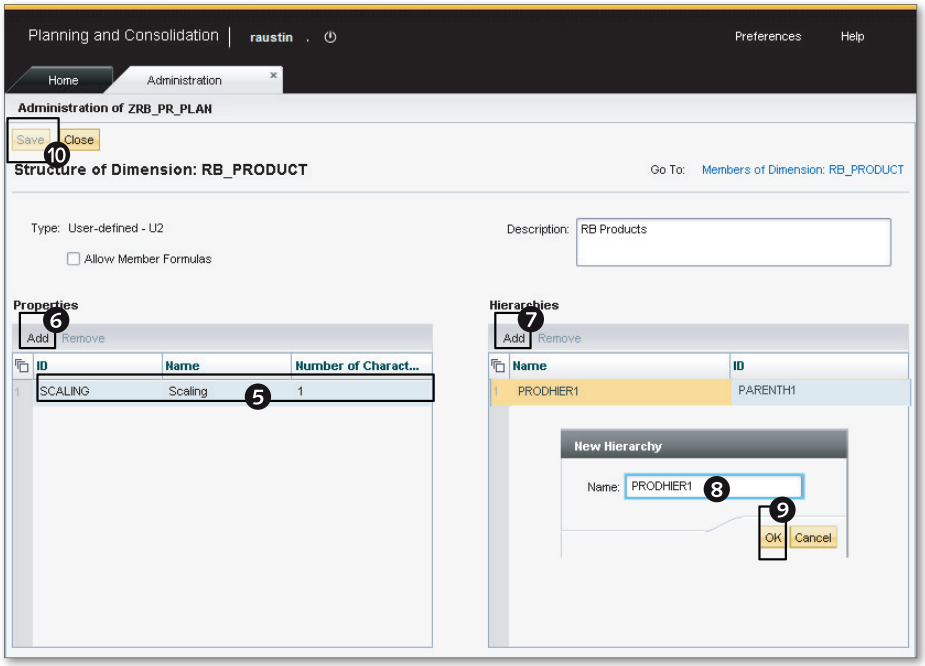


Figure 3.34 Creating a Custom Dimension—Part B

Similarly, create another sales group dimension with the technical name "RB_SALESGRP" and description "RB SALES GROUPS". We'll discuss the process of manually entering data into a dimension in the next section.

Manually Entering Data into a Dimension

Three options are available when entering data into a dimension in SAP BPC. You can load it from a flat file, load from an SAP NetWeaver BW system, or enter it manually. We'll discuss the option of loading data from a flat file and from an SAP NetWeaver BW system in Chapter 4, when we discuss the process of loading master and transaction data. You can use the option to enter data manually into a dimension when there are limited data records in the dimension.

Next, we'll explain how to manually enter data into the RB_ACCOUNT dimension for our case study.

1. To add data manually, click on DIMENSIONS in the left pane (refer to Figure 3.28, ❶). Select the RB_ACCOUNT dimension, and click on EDIT MEMBERS. This displays the member, hierarchy, and property data of the dimension. Delete the

existing data in the dimension because it was originally copied from the ACCOUNT dimension, and enter the member data as shown in Figure 3.35. You can delete a member ID only if it is not already used in a model. If any model contains the member ID, you cannot delete it.

2. The data entered into the dimension has to be processed in SAP BPC to take effect. Select SAVE AND PROCESS (Figure 3.35, ❶).

3. This opens the PROCESS DIMENSION dialog box. Processing a dimension causes a corresponding attribute change run for the InfoObject in SAP NetWeaver BW. Click on NO to the question "Do you want to take the environment offline?" (Figure 3.35, ❷). The changes made to the dimension members are processed. You can see the dimension has a status of green to indicate that all of the updates to the dimension are processed (Figure 3.36).

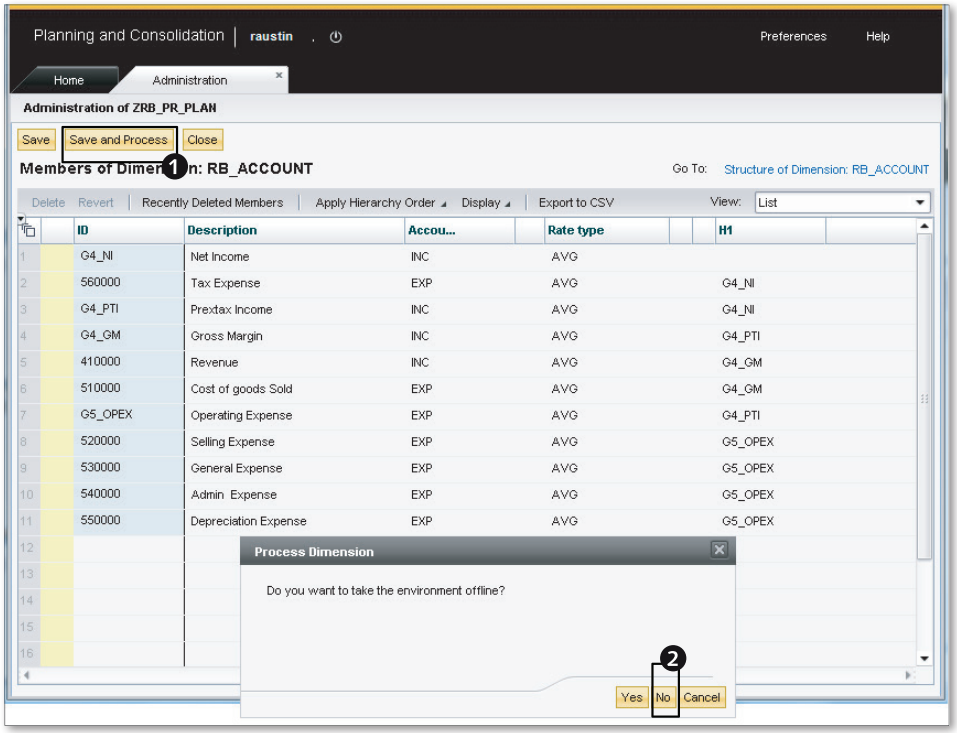


Figure 3.35 Manually Entering Data into a Dimension—Part A

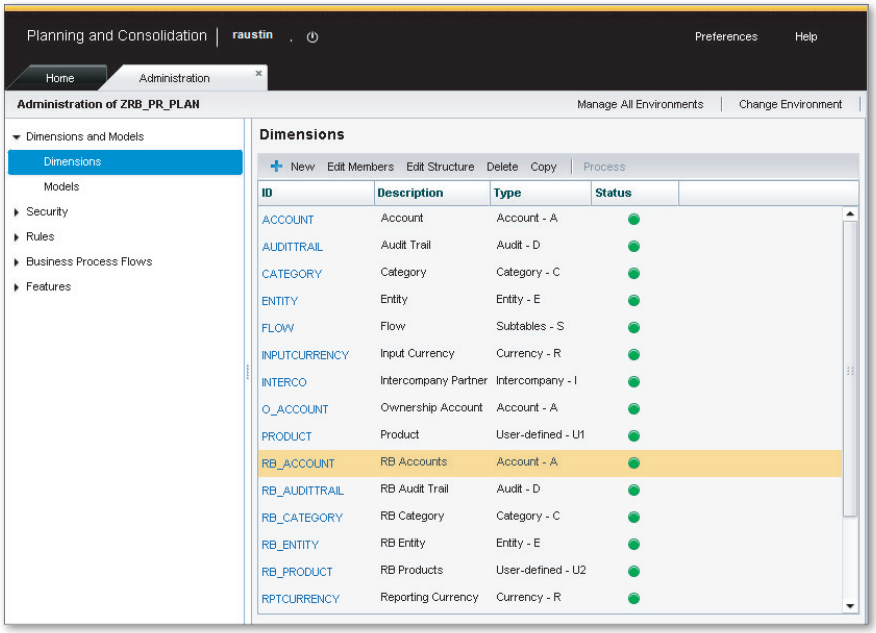


Figure 3.36 Manually Entering Data into a Dimension—Part B

You have now seen how to create dimensions and maintain them in SAP BPC. We displayed properties associated with dimensions and discussed how to use hierarchies in dimensions. We also created a dimension by copying from an existing dimension and created a custom dimension.

The master data for dimensions as it appears in SAP BPC is available for download at this book's website at *www.sap-press.com.* The name of the file is *Dimension Members.xls*.

We'll now discuss models and how to create them.

3.3.4 Model

A model in SAP BPC meets the particular functional business requirement of storing transaction data and is also available for reporting. Examples of models are sales planning, gross profit planning, and head-count planning. The following are a few basic but important facts about models:

- A model is created with respect to an environment and is a collection of dimensions. The dimensions included for a model are based on business requirements.

- A model can use any dimensions included in an environment.
- There are no restrictions on the number of dimensions included in a model, but we recommend that you limit the number to 15 to maintain good performance. The system automatically creates one dimension called the Data Package dimension for every BPC model. By limiting the number of dimensions to 15, all dimensions will be set as line item dimensions, which improve performance.
- Several objects and tables are created in the background, depending on the type of model created. Some of these include comment tables, journals, work status tables, audit tables, logic script files, file service folders, and Data Manager packages to load and manage data for the model.
- Two options are available when you create a model:
  - NEW: Select this option when you need to create a model from scratch or based on another model. If you have selected the option to create the new model based on an existing model, you can change the dimension assignments. For example, you can remove a dimension from what the template provided or add a new dimension to the model.
  - COPY: Select this option when you need to copy the dimension and record from a source model to a target a model. When this option is selected, you are not able to change the dimension assignments.

There are two main types of models:

- **Reporting**
  Examples of reporting models are sales planning and consolidation models. The following subtypes are available under reporting models:
  - Financial: This type of model is used primarily for planning.
  - Consolidation: This type of model is used for financial consolidation. Consolidation is the process of reporting financial results of an organization either for legal or management reporting purposes.
  - Standard: A reporting model that is neither a financial nor a consolidation model is categorized as a standard model.
- **Drivers and Rates**
  This type of model is used to support a reporting model. The following subtypes are available in the DRIVERS AND RATES area in the NEW MODEL dialog box:
  - EXCHANGE RATES: This model is used to store exchange rates and to enable currency translation for models used in reporting. When a company does

business in more than one currency, it may need to translate the business transactions into one or more reporting currencies. In these cases, the rate model is used to perform currency translations to one or more reporting currencies. In SAP BPC, you can report transactions in more than one reporting currency. The rate model (called RATE) is delivered as part of ENVIRONMENTSHELL. The exchange rate data in the RATE model needs to be maintained either manually using input templates or automatically loaded into the model.

- Ownership: The ownership model is used for consolidating financial information. An organization may be composed of a group of companies with a different ownership mix; for example, it may hold a 50% stake in one company and have also made a significant investment in another company. In such cases, it may be necessary for legal reporting to consolidate the financial data of all of the companies in the group as a whole. The ownership model stores the percentage ownership in different companies and helps perform legal consolidation.
- Generic: This model type is used to support a reporting model. A model containing the price of products is an example of this subtype of a Drivers and Rates model.

All models, whether of type Reporting or Drivers and Rates, must include the four basic dimensions: Category, Account, Entity, and Time.

When a request for creating a model is initiated in SAP BPC, a corresponding InfoCube and MultiProvider are created in the SAP NetWeaver BW system. Table UJA_APPL stores the technical name of the InfoCube and MultiProvider. The naming convention when creating the InfoCube and MultiProvider is as follows:

- **InfoCube**
  //CPMB/<Environment Prefix>I<Model Prefix><1GeneratedChar>
- **MultiProvider**
  //CPMB/<Environment Prefix>M<Model Prefix><1GeneratedChar>

It is easy to delete or add dimensions to a model in SAP BPC. When dimension data is deleted from a model, the SAP NetWeaver BW system creates a shadow InfoCube in the background and moves the data from the original InfoCube to the shadow InfoCube. After successfully loading the data to the shadow InfoCube, the system updates the database for the model to point to the shadow InfoCube and deletes the original InfoCube because it is no longer required.

Note

Do not change any model objects directly in the SAP NetWeaver BW system. Always use the SAP BPC frontend to make changes to a model.

Model Optimization

SAP BPC lets you optimize models with the following two options:

- Light Optimize: When data is loaded into a model, an open request is created for the InfoCube associated with the model in the SAP NetWeaver BW system. This request is open (shown with a yellow status) until a particular threshold of records is updated. When the threshold is reached, the request is closed (shown with a green status). When Light Optimize is selected, all open requests are closed and set to green. The cube is compressed, and all of the indexes are built. The statistics for the InfoCube are also updated during this process.
- Full Optimize: Full Optimize works the same as Light Optimize, but it also reviews the data model for scope for additional optimization. If, during the course of the check, it is determined that the data model of the InfoCube requires changing, the necessary changes to the data model will be made. This process can take some time to complete, depending on the volume of data in the InfoCube.

Parameters for a Model

You can set several parameters for a model. One of these is YTDINPUT, which determines how quantitative data is stored in a model. By default, the value of this parameter is 0. This implies that the quantitative data for each record in the model is stored for the period for which it is entered. When the value of this parameter is set to 1, the quantitative data is year to date (YTD) and up to the period for which it is entered.

We'll use an example to illustrate the functionality of this parameter. Consider a situation where YTDINPUT is set to 1, with two records entered into a model. The first is for an amount of $200—entered for January 2012—and the second is for an amount of $500—entered for February 2009. In this case, the entries for these two months represent YTD values. So the periodic value for January 2012 is $200, and the periodic value for February 2012 is $300 ($500 – $200).

We'll discuss other application-specific parameters in Chapter 7.

Creating a Model

We'll now explain how to create a model for Rich Bloom to use for planning profitability. These dimensions are required for our planning model:

- RB_ACCOUNT
- RB_CATEGORY
- RB_ENTITY
- RB_SALESGRP
- RB_PRODUCT
- RB_AUDITTRAIL
- RPTCURRENCY
- TIME

We'll create a new model to meet the requirements for planning that includes these dimensions. Proceed as follows:

1. Connect to the ZRB_PR_PLAN environment.
2. Click on MODELS in the left pane, and select NEW (Figure 3.37, ❶ and ❷).

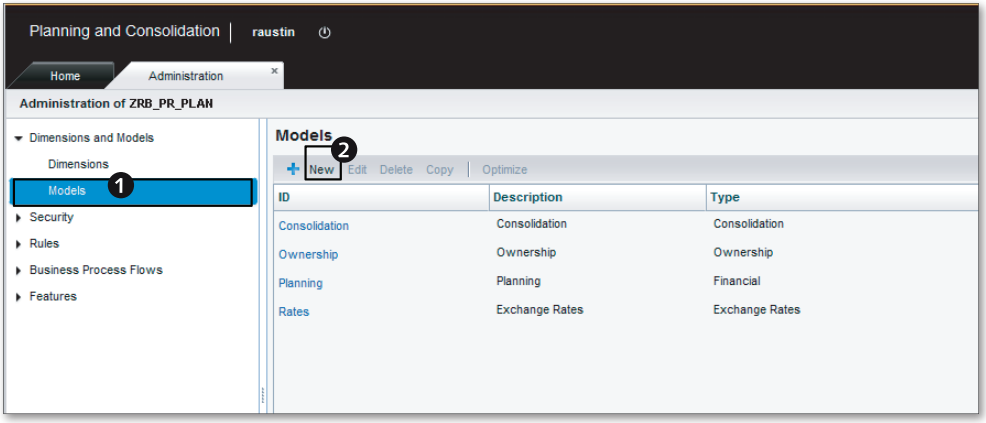


Figure 3.37 Creating a Model—Part A

3. You're now guided through a wizard to create the model. Enter "ZRB_SALES_CMB" as the technical name and "Actual and Sales Plan Data" as the description for the model (Figure 3.38, ❸ and ❹). Select NEXT (Figure 3.38, ❺).

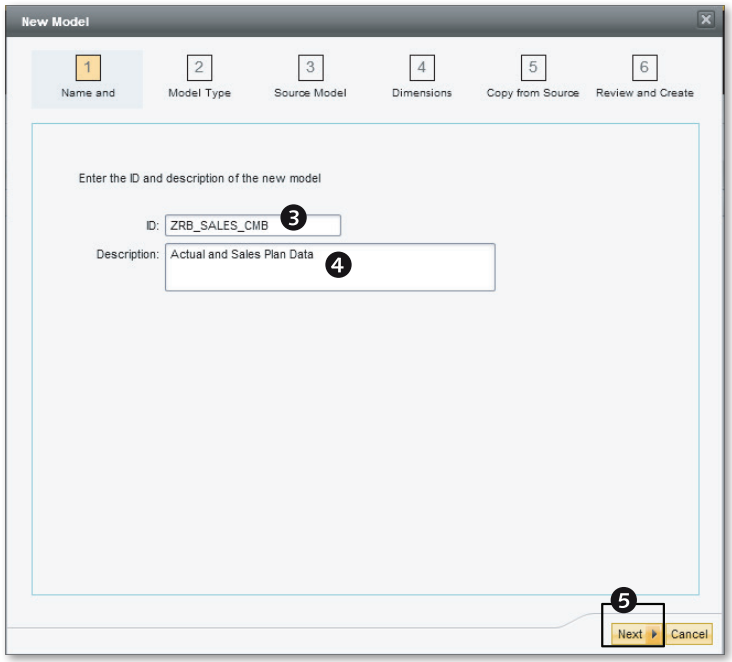


Figure 3.38 Creating a Model—Part B

4. You're prompted to select the model type. We want to create a financial model that can be reported, so under REPORTING, select FINANCIAL (Figure 3.39, ❻). The default data model of type FINANCIAL is PER - PERIODIC (Figure 3.39, ❼). A Financial model should always be associated with an Exchange Rates model. A model called RATES is delivered as part of ENVIRONMENTSHELL and has been copied into the current environment. Select this model as the Exchange Rate Model (Figure 3.39, ❽). Select NEXT (Figure 3.39, ❾).
5. Select whether you want to start using a blank model or use a source model as the template for creating the new model. Select AN EXISTING MODEL (Figure 3.40, ❿), and choose PLANNING, which is a model that is delivered as part of ENVIRONMENTSHELL and has been copied into the current environment as the basis for creating the new model (Figure 3.40, ⓫). Select NEXT (Figure 3.40, ⓬).

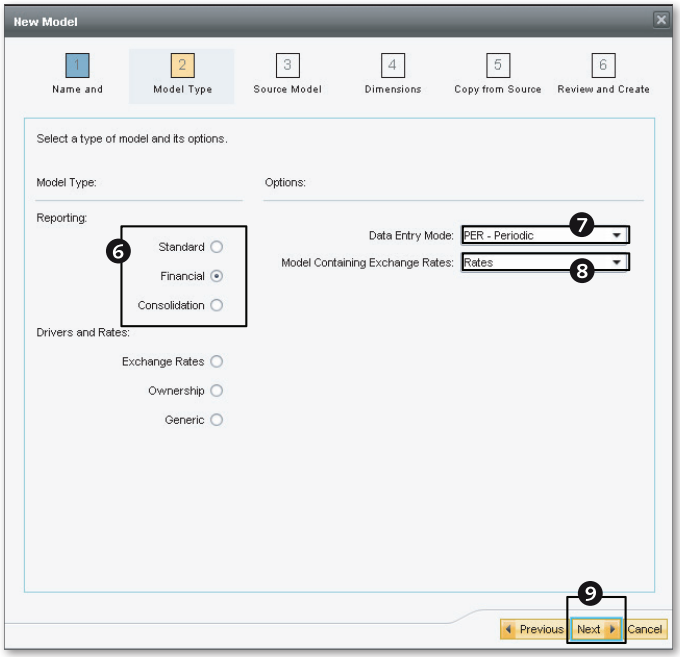


Figure 3.39 Creating a Model—Part C

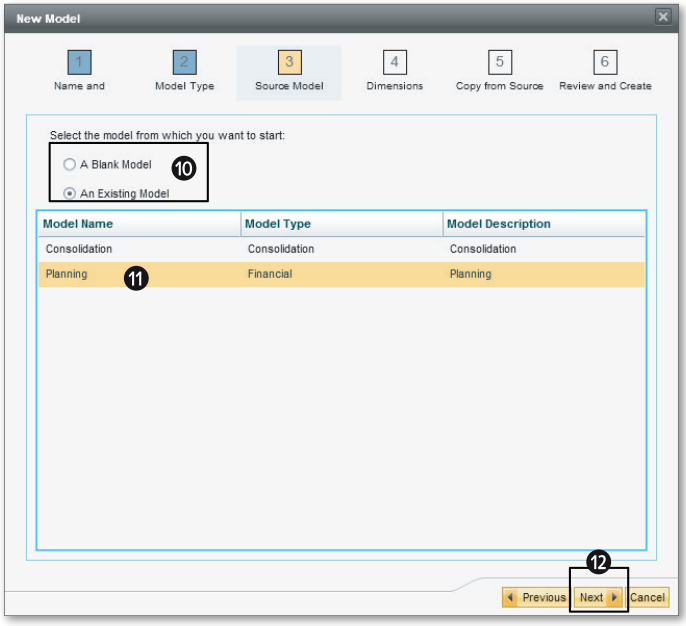


Figure 3.40 Creating a Model—Part D

6. In the next screen, use the arrows to remove and add dimensions to your model. Our model will not need the dimensions shown in Figure 3.41 (⓭). Click on the arrow as shown in Figure 3.41 (⓮) to remove them from the model.

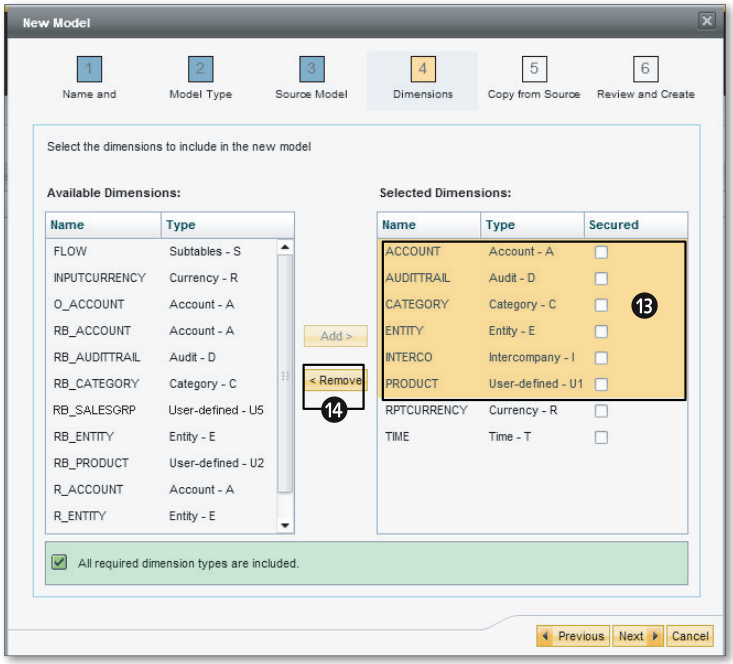


Figure 3.41 Creating a Model—Part E

7. Add the RB_ACCOUNT, RB_CATEGORY, RB_SALESGRP, RB_AUDITTRAIL, RB_ENTITY, RB_PRODUCT, RPTCURRENCY, and TIME dimensions from the list of Available Dimensions in the left area to the new model in the right area using the Add button to the Selected Dimensions (Figure 3.42, ⓯).
8. A dimension can be marked as a secured dimension. Marking a dimension assists with restricting data access for users based on values in that dimension. To do this, select the dimension, and click on Secured. Select Next (Figure 3.42, ⓰).
9. This screen provides information on the objects that will be copied from the source model. This is applicable only if an existing model was selected as the basis for creating the new model. Select Next (Figure 3.43, ⓱).

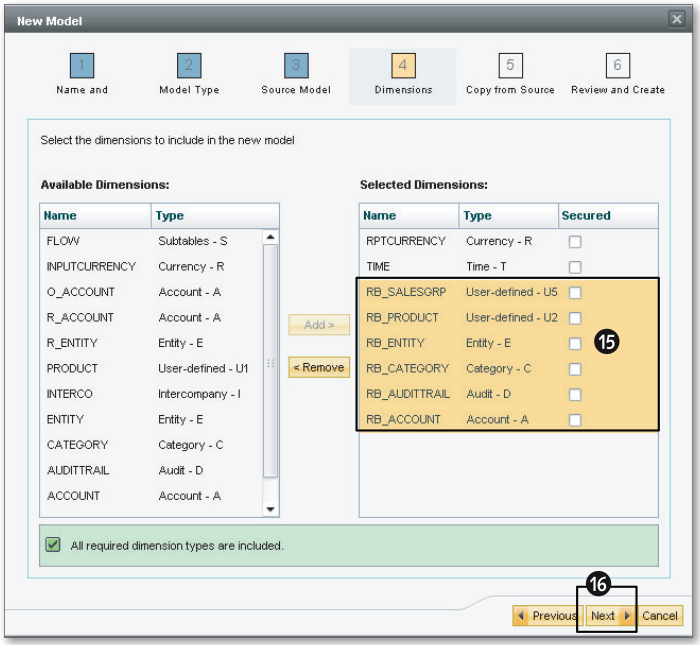


Figure 3.42 Creating a Model—Part F

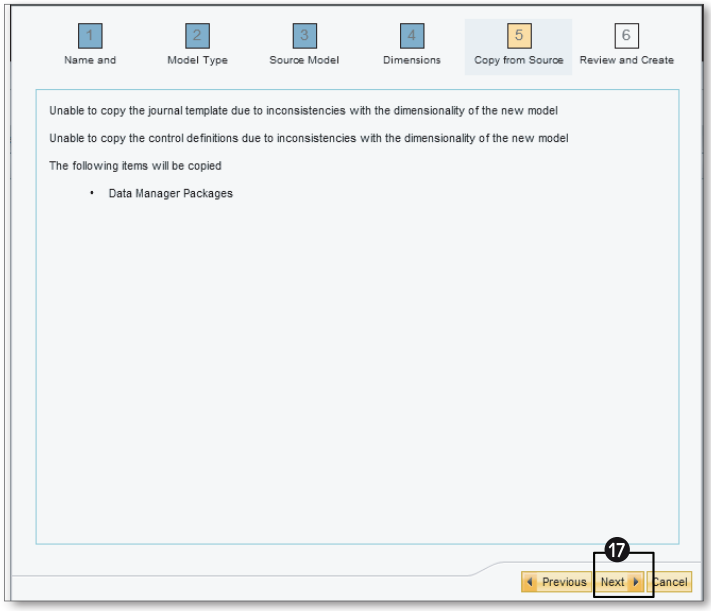


Figure 3.43 Creating a Model—Part G

10. The last screen provides the opportunity to review the technical name of the model, the dimensions included in the model, and the model type. The items that will be copied from a source model are displayed when using an existing model to create the new model. Click on CREATE to create the new model (Figure 3.44, ⓲). A status message indicates that the new model has been created (Figure 3.45). The PROFITABILITY PLANNING model is created and displayed under the MODELS available for the environment.

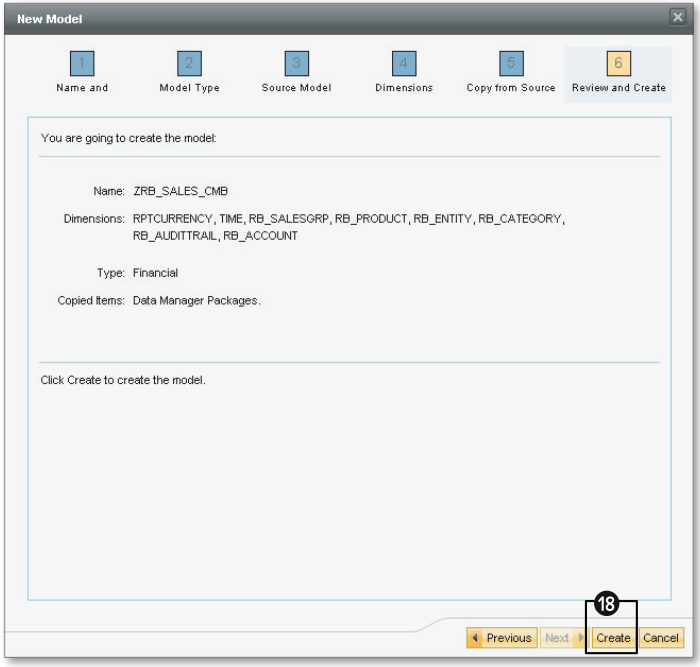


Figure 3.44 Creating a Model—Part H

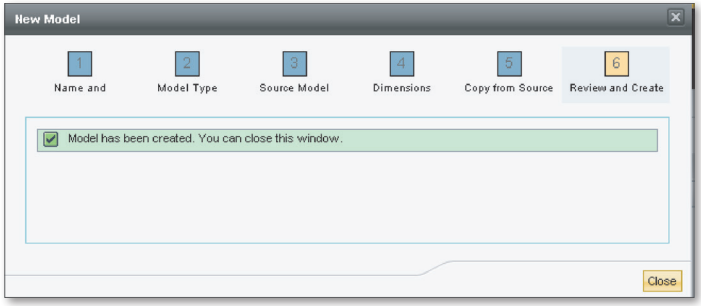


Figure 3.45 Creating a Model—Part I

We have now configured a planning model that Rich Bloom will use for planning profitability. We included all of the necessary dimensions in the model that will be used to develop a plan and compute the profitability.

3.4 Summary

In this chapter, we introduced a case study of a sample company, Rich Bloom, which uses SAP BPC for profitability planning. We developed the staging environment in the SAP NetWeaver BW system that will be used to store actual sales and cost data. We also discussed the building blocks of creating the planning model in SAP BPC. We discussed environments, dimensions, and models, and we saw how the objects created in SAP BPC translated to corresponding objects in the SAP NetWeaver BW system.

In the next chapter, you'll learn the steps for loading data into SAP NetWeaver BW and SAP BPC for NetWeaver.

Typically, historical data available in a source system is loaded into SAP BPC and used as basis to plan for the future. Data from different source systems may also be gathered to produce consolidated financial statements. To perform planning and consolidation tasks, data is extracted from a source system, transformed to meet operational needs, and then loaded into data targets in SAP NetWeaver BW and SAP BPC systems.

4 Loading, Scheduling, and Managing Data in SAP BPC for NetWeaver

In the previous chapter, we introduced our model company, Rich Bloom, and discussed a case study that we'll carry out throughout the book. To meet the requirements of the case study, we built an InfoCube in SAP NetWeaver Business Warehouse (BW) and a model in SAP Business Planning and Consolidation (SAP BPC) to store actual sales and cost data. In this chapter, we explain the extraction, transformation, and loading (ETL) processes used in SAP NetWeaver BW and SAP BPC systems. We also demonstrate how to use the SAP BPC tool to create baseline plan based on actual data.

Section 4.1 discusses the steps of loading data from a flat file to an InfoCube in SAP NetWeaver BW. If you're new to SAP NetWeaver BW, this section will help you understand the process of loading transaction data into an InfoCube.

Section 4.2 explains the options available to load master and transaction data into SAP BPC. You'll see how to use transformations and conversions when loading data to dimensions and models. This section also explains the importance of Data Manager packages and discusses how to create and maintain them. Data manager packages enable you to perform tasks relating to loading and managing data in SAP BPC. In this section, we'll discuss the use of process chains in a Data Manager package and introduce the process types available for use in a process chain inside an SAP BPC model.

Section 4.3 discusses how to use 2011 actual sales and cost data to create baseline plan data for 2012. We'll copy actual sales and cost data for 2011 to a plan version for 2012.

4.1 Loading Data into an InfoCube in SAP NetWeaver BW

In this section, we'll discuss the steps of loading data from a flat file into an InfoCube in SAP NetWeaver BW.

4.1.1 Creating a DataSource

A *DataSource* is an object that provides data to load into an InfoProvider in SAP NetWeaver BW. The actual source of the sales and cost data for our sample company, Rich Bloom, is in an SAP ECC system. The daily sales and cost data created in the SAP ECC system is brought into the SAP NetWeaver BW system on a nightly basis. But for the purposes of our case study, we'll assume that the sales and cost data for 2011 exists in a flat file. We'll load the sales and cost data from this flat file into the Sales InfoCube in the SAP NetWeaver BW system.

Before you start creating the DataSource, you must create a flat file that contains the sales and cost data for 2011. We use a comma-separated value (CSV) format containing the following columns:

- Company Code
- Account
- Calendar Month
- Calendar Year
- Product
- Sales Group
- Amount
- Currency

Then enter the data as shown in Figure 4.1. Name the file "us_salesdata.csv", and save it to a directory called *C:\Planning*. (The US_SALESDATA.CSV file is available for download at this book's website at *www.sap-press.com*.)

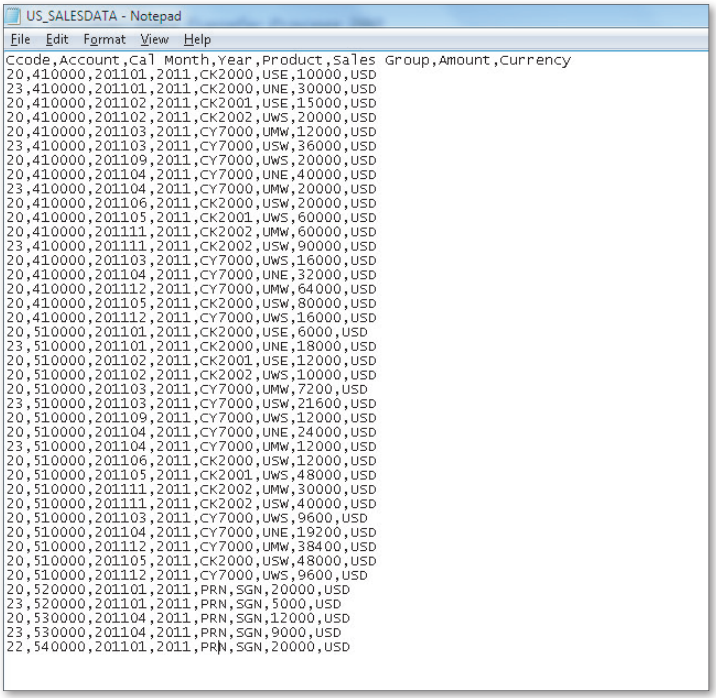


Figure 4.1 Sales and Cost Data for 2011 in a Flat File

Next, perform the following steps to create the DataSource, as illustrated in Figure 4.2, Figure 4.3, and Figure 4.4:

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • Source Systems.
3. Select and double-click on a flat file source system for creating the DataSource. We used the PC File System (Figure 4.2, ❶).
4. The DataSources for the <File System> dialog box appears (in this case, PC_FILE PC File System). Select any application component area (e.g., Non-SAP_Sources), right-click, and select Create DataSource from the context menu (Figure 4.2, ❷).
5. In the Create DataSource window, enter the name of the DataSource ("ZSLS_FILE") (Figure 4.2, ❸), and then select Transaction Data from the Data Type DataSource dropdown list (Figure 4.2, ❹).
6. Click on Enter (Figure 4.2, ❺).

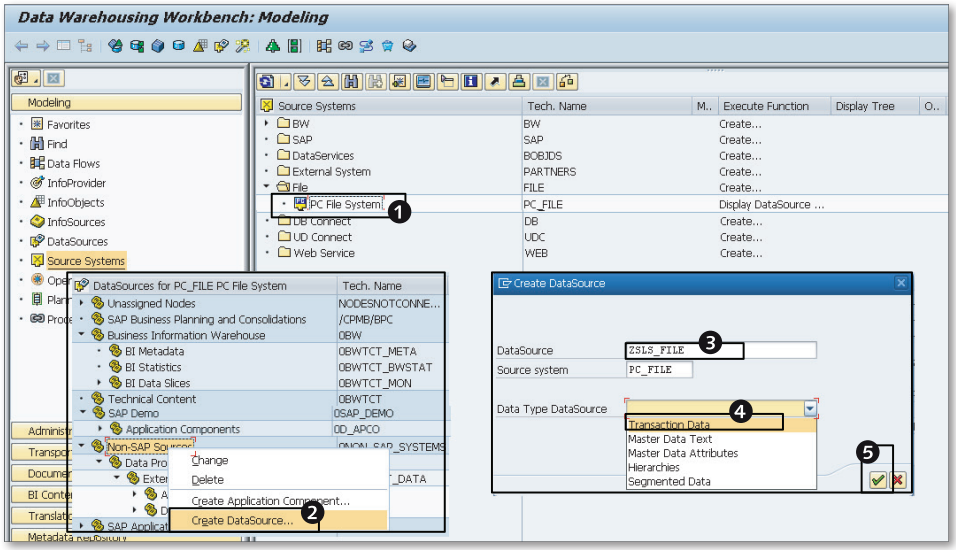


Figure 4.2 Create a Flat File DataSource for Sales Data—Part A

7. On the General Info. tab in the Change DataSource <File Name> window, enter the short ("Sales actual file"), medium ("Sales actual file"), and long ("Sales actual file") descriptions for the DataSource. We'll be using this DataSource to load sales and cost data for 2011 from a flat file.

8. On the Fields tab, enter the list of fields for which data exists in the flat file. The sequence you use is the same as the one used for the data in the flat file. Instead of entering a field name, you can enter the name of an InfoObject in the first row of the InfoObject Template column (Figure 4.3, ❻) that represents the field.

9. After you enter the first InfoObject ("0COMP_CODE") and click on Enter, the Default from InfoObjects dialog box appears (Figure 4.3). Select Do Not Show This Question Again in This Session, and click on Copy (Figure 4.3, ❼ and ❽). This transfers all of the properties from the InfoObject to the corresponding field in the Field column (Figure 4.4, ❾).

10. Continue to enter InfoObjects into the InfoObject Template column (Figure 4.4, ❿) by clicking on the Insert button (Figure 4.4, ⓫), following the sequence of data coming from the flat file that contains the sales and cost data for 2011.

11. Click on the Activate button to save and activate the DataSource (Figure 4.4, ⓬).

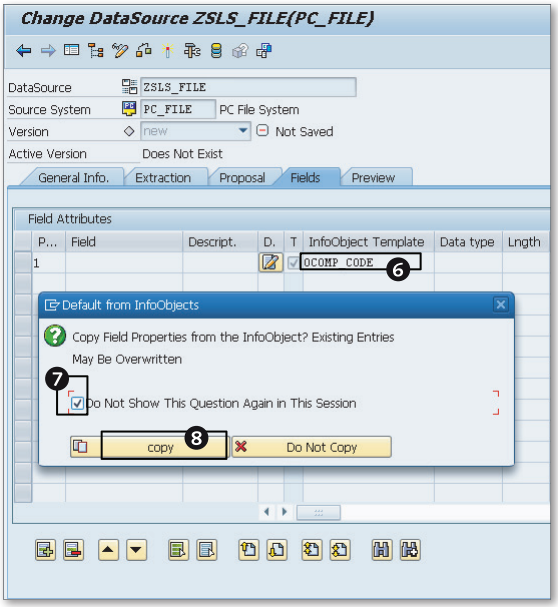


Figure 4.3 Create a Flat File DataSource for Sales Data—Part B

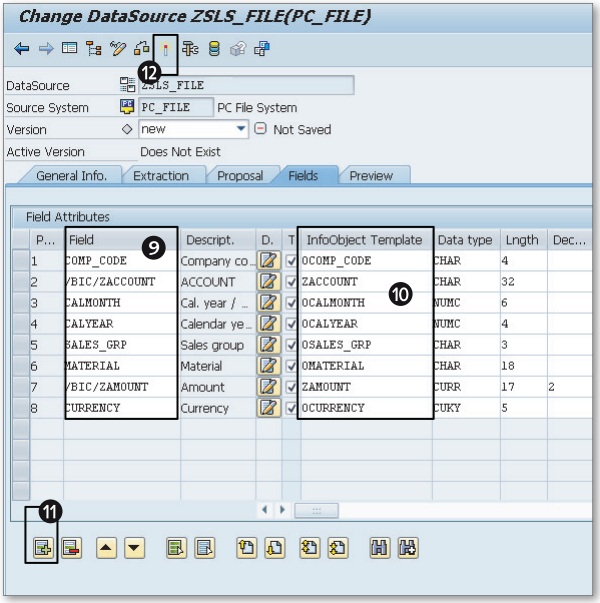


Figure 4.4 Create a Flat File DataSource for Sales Data—Part C

When the DataSource is successfully activated, the system automatically creates an equivalent persistent staging area (PSA) table. The structure of the PSA table mirrors the structure of the file layout of the source data.

4.1.2 Creating Transformations

In SAP NetWeaver BW, the data coming from a source system is loaded into the DataSource or PSA in its original form as sent from the source system. The *transformation* process, however, lets you apply rules to modify the data coming from a DataSource before it is loaded into an InfoProvider.

At least one transformation process is required before data reaches the data target. For our case study, we will create a transformation process to transform the data from the PSA associated with the Sales DataSource to the Sales InfoCube.

Perform the following steps to create a transformation that loads data from the Sales DataSource created earlier into the Sales InfoCube, as illustrated in Figure 4.5 through Figure 4.8:

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Under the Sales Management InfoArea, right-click on the Sales Actual InfoCube, and select Create Transformation from the context menu (Figure 4.5, ❶).
4. In the Create Transformation window, select the source of the transformation. Select the Object Type (DataSource), and enter the Source System ("PC_FILE"), and the DataSource ("ZSLS_FILE") (Figure 4.5, ❷, ❸, and ❹).
5. Click on Enter (Figure 4.5, ❺).
6. The Create Transformation window appears (Figure 4.6). The system automatically proposes a mapping for the fields from the DataSource to the InfoObjects in the Sales Actual InfoCube. The proposal is based on the mapping of the InfoObjects in the InfoObject Template column of the DataSource to the same InfoObject that exists as a characteristic or key figure in the Sales Actual InfoCube.

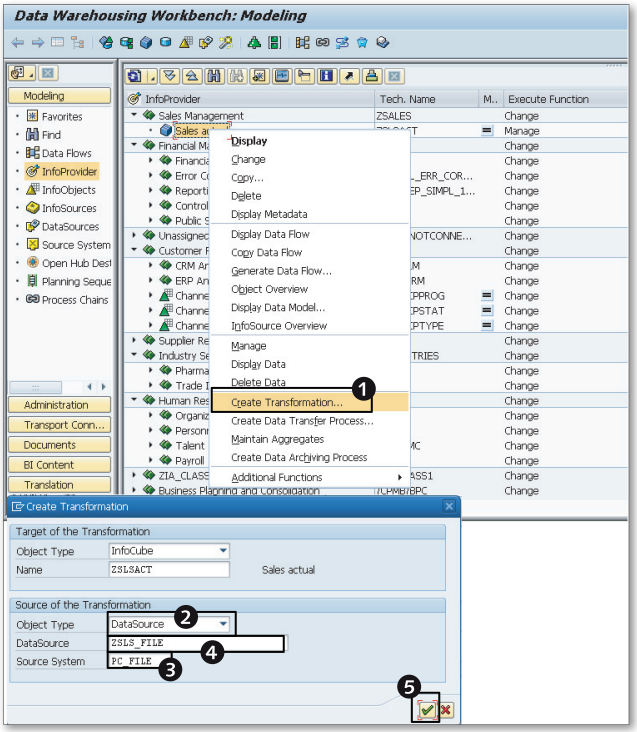


Figure 4.5 Create a Transformation—Part A

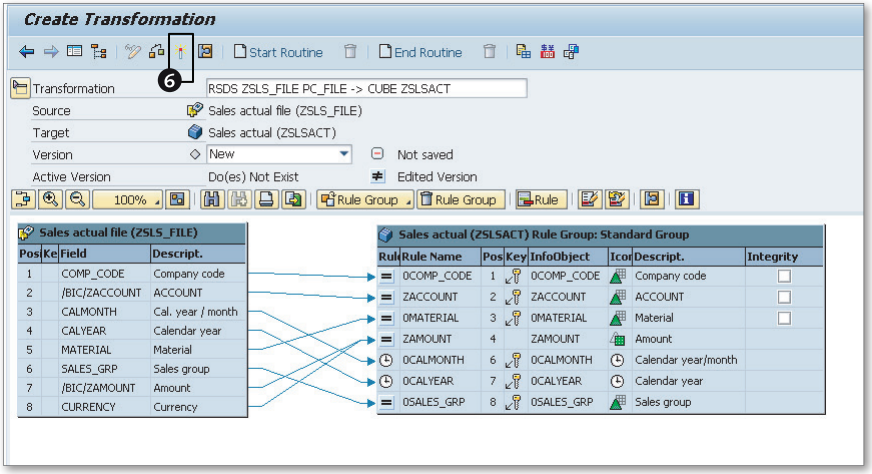


Figure 4.6 Create Transformation—Part B

You may at times notice that some of the InfoObjects are not mapped. This is because data for these InfoObjects is not available in the DataSource, so these values have to be directly set in the transformation. In these cases, select the appropriate InfoObject in the Rule Group: Standard Group table, and double-click on it. There you'll have additional options to map the value for those InfoObjects.

7. After you've assigned all of the mappings, click on the Activate button to save and activate the transformation object (Figure 4.6, ❻).

4.1.3 Creating the Data Transfer Process

The data transfer process (DTP) provides the ability to extract and load data from one persistent object to another.

Note

Note that the transformation process we created earlier only provides the rules for the extraction and loading of data. The DTP then uses the transformation to actually load the data from one persistent area to another.

For our case study, we want to define a DTP that extracts data from the Sales Actual File DataSource, uses the transformation process we created earlier, and loads the data into the Sales Actual InfoCube.

We'll create a DTP by performing the following steps, as illustrated in Figure 4.7 and Figure 4.8:

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Under the Sales Management InfoArea, right-click on the Sales Actual InfoCube and select Create Data Transfer Process from the context menu (Figure 4.7, ❶).
4. We are creating a DTP that loads data from the Sales Actual File DataSource, which we created earlier, to the Sales InfoCube. In the Creation of Data Transfer Process window, select the source object for the DTP. Select the Object Type source (DataSource), and enter the Source System ("PC_FILE") and the

DataSource we created earlier ("ZSLS_FILE") (Figure 4.7, ❷, ❸, and ❹) and click on Enter (Figure 4.7, ❺).

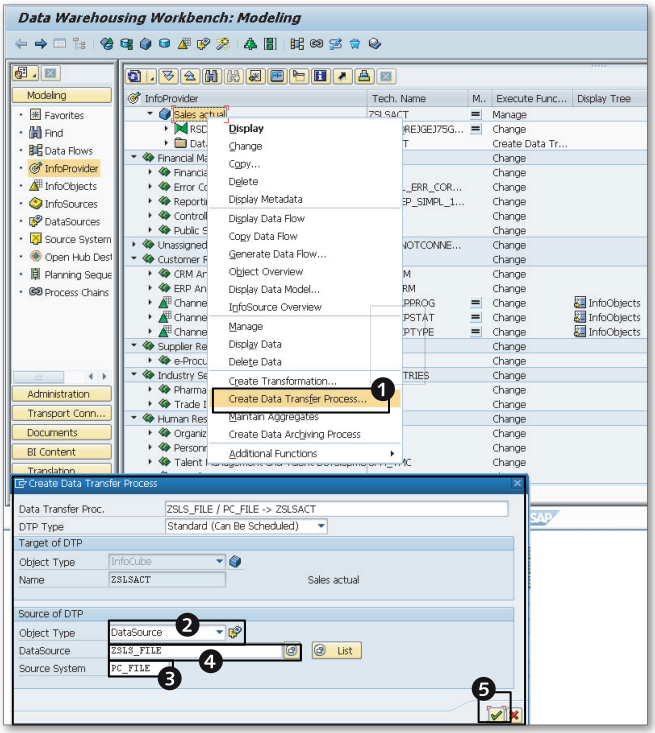


Figure 4.7 Create Data Transfer Process—Part A

Note

A DTP can be created between a source and a target object only if a transformation process exists between the two objects.

5. In the Change Data Transfer Process window (Figure 4.8), select the extraction mode. If you select Full, all of the data from the source is loaded into the target. If you select Delta, only the changes made since the last time the data was extracted from this source are extracted into the target. We will use the Delta extraction mode.
6. Click on the Activate button to save and activate the DTP object (Figure 4.8, ❻).

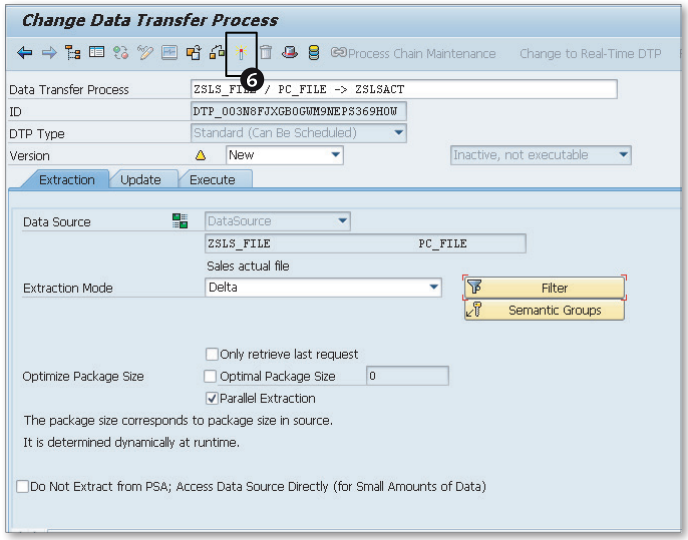


Figure 4.8 Create Data Transfer Process—Part B

4.1.4 Creating InfoPackages

In the previous steps, we configured objects such as the DataSource, transformation, and DTP that will be used to load sales actual data to the Sales InfoCube. But the configuration of the objects by itself does not load the data; to do this, you must perform the following steps:

1. Schedule the loading of data from a DataSource to the PSA.
2. Schedule the load data from the PSA to a data target, which could be an InfoCube, DataStore, or InfoObject.

The InfoPackage is the object that is used to schedule the load of data from a DataSource to a PSA. Create an InfoPackage for loading the sales data using the following steps, as illustrated in Figure 4.9, Figure 4.10, and Figure 4.11:

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Right-click on the Sales Actual File DataSource, and select Create InfoPackage from the context menu (Figure 4.9, ❶).
4. In the Create InfoPackage window, enter the name of the InfoPackage ("Sales actual data"), and click on Enter (Figure 4.9, ❷ and ❸).

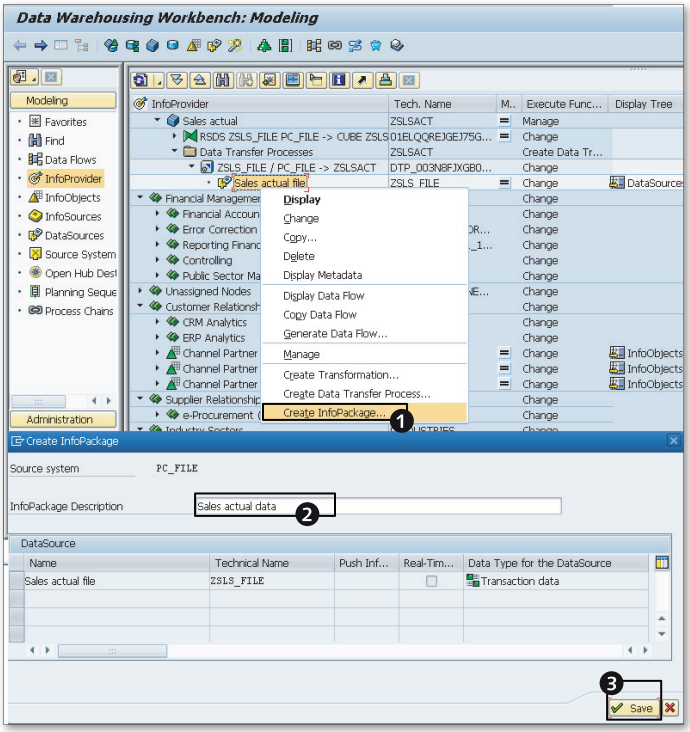


Figure 4.9 Create InfoPackage—Part A

5. In the SCHEDULER (MAINTAIN INFOPACKAGE) window, select the EXTRACTION tab (Figure 4.10, ❹).
6. Choose the ADAPTER, which determines whether the file you want to load exists on a local workstation or an SAP application server. Because this file exists on your PC, select LOAD TEXT-TYPE FILE FROM LOCAL WORKSTATION from the dropdown list (Figure 4.10, ❺).
7. Enter the name of the file (C:\PLANNING\US_SALESDATA.CSV) that contains the sales and cost data for 2011 (Figure 4.10, ❻).
8. Specify the header rows to be ignored if you want the system to ignore header rows. (Refer to Figure 4.1 for the layout of the file.) Because the first row of data contains the header, enter "1" in the HEADER ROWS TO BE IGNORED field (Figure 4.10, ❼).
9. Next to DATA FORMAT (Figure 4.10, ❽), select SEPARATED WITH SEPARATOR (FOR EXAMPLE, CSV) from the dropdown list, and, next to DATA SEPARATOR, enter a comma (Figure 4.10, ❾).

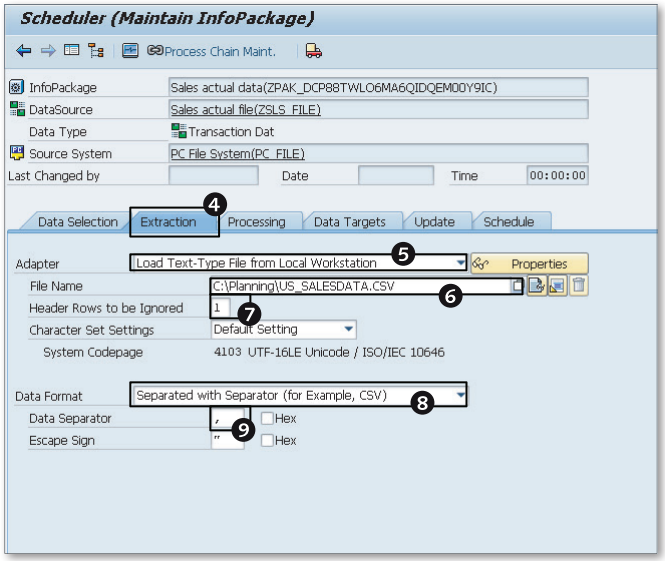


Figure 4.10 Create InfoPackage—Part B

10. Select the SCHEDULE tab (Figure 4.11, ❿). Ensure that START DATA LOAD IMMEDIATELY is selected (Figure 4.11, ⓫). Click on START (Figure 4.11, ⓬). The data will now be extracted from the DataSource into the PSA. Click on the MONITOR button to check the status of the extraction process (Figure 4.11, ⓭).

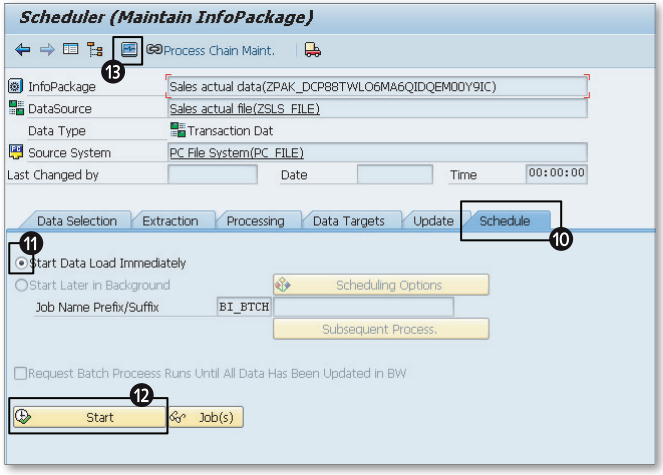


Figure 4.11 Execute the InfoPackage

11. The Monitor Administrator Workbench window, which is a pane in the Monitor InfoPackage screen in Figure 4.12, shows the status of the data extracted from the flat file DataSource into the PSA. If the data load is successful, the status of the load is displayed with a green traffic light.

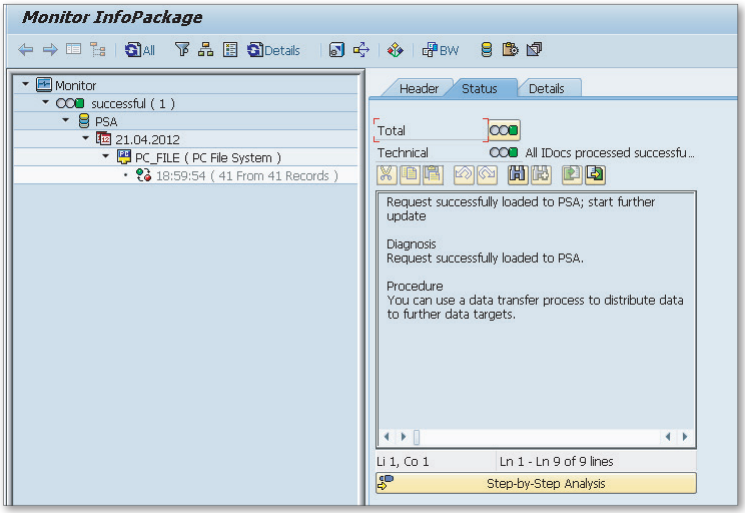


Figure 4.12 Monitor Status of InfoPackage

4.1.5 Loading Data from the PSA to the Data Target

Now that we have loaded the data from the DataSource to the PSA, we're ready to schedule the load of data from the PSA to the Sales InfoCube using the DTP. The process applies the transformation when loading the data from the PSA to the data target.

1. Open the Data Warehousing Workbench using Transaction RSA1.
2. Select Modeling • InfoProvider.
3. Locate the DTP created earlier under the Sales InfoCube, and double-click on it. The pane on the right displays the details of the DTP object (Figure 4.13).
4. Select the Execute tab, and then click on the Execute button (Figure 4.13, ❶).
5. Click on the Monitor button to check the status of the load (Figure 4.13, ❷).
6. Return to the Data Warehousing Workbench by clicking on the back arrow button. Select the Manage option of the Sales InfoCube.

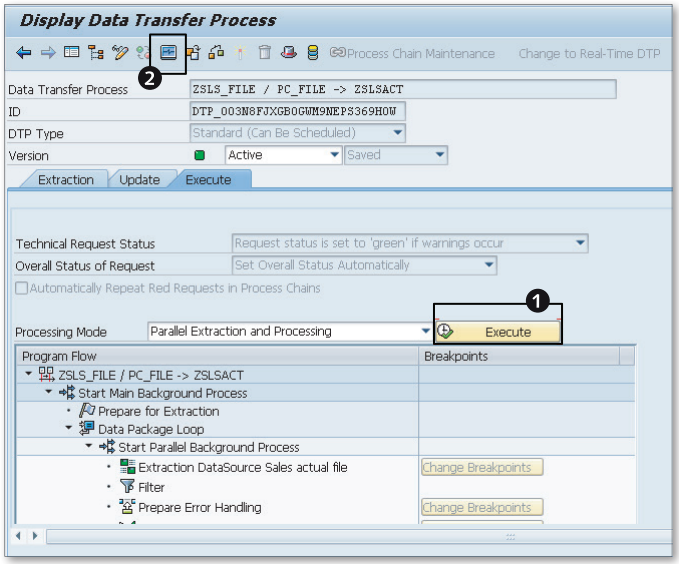


Figure 4.13 Execute Data Transfer Process

7. In the InfoProvider Administration window, select the Requests tab to see the requests that were loaded into the InfoCube. Click on the monitor icon associated with a request to view the details. Figure 4.14 shows the details of the request loaded into the Sales InfoCube.

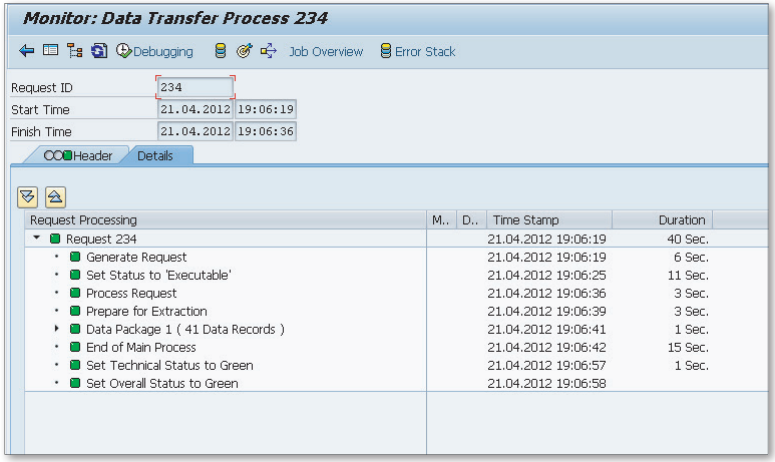


Figure 4.14 Monitor Data Transfer Execution Process

After reading this section, you should understand how to load data into an InfoCube using a flat file as the DataSource. Rich Bloom can now use the sales data for 2011 in this InfoCube as the basis for planning in 2012.

In the next section, we'll discuss the options for loading master and transaction data into SAP BPC. We'll demonstrate examples of how to load master and transaction data as it relates to our case study.

4.2 Loading Data into SAP BPC

Two types of data exist in SAP BPC: master data and transaction data. Master data is stored in a dimension, and transaction data is stored in a model. Although both the master data and transaction data can be manually entered into SAP BPC, it may not be feasible to adopt this approach for large volumes of data. To automate the process of loading master data and transaction data, SAP BPC provides the following options:

- **Flat file**
  When this option is selected, master or transaction data can be loaded into a dimension or a model from a flat file. The flat file is the most flexible method to load data into an SAP BPC model. In this case, the flat file is staged in the BPC server system folder before it is loaded into the model.
- **InfoProvider in SAP NetWeaver BW**
  In SAP BPC for NetWeaver, you can leverage the data in an InfoProvider in the SAP NetWeaver BW system. You can use an InfoObject as the source to load master data and hierarchy data into a dimension. You can use an InfoCube or DataStore object (DSO) as source to load data into an SAP BPC model.

To begin, we'll discuss the transformation and loading process in SAP BPC. When you load data, extended features are available to transform and convert the data coming from a source file. The transformation definition interprets the source of the data; it specifies the layout of the file, the columns in the file, and how the data should be loaded into the model. The conversion definitions are used when the source data should be interpreted differently before it is loaded into SAP BPC.

4.2.1 Transformation and Conversion

The transformation file specifies the definitions used to interpret data coming from an external source before loading it into an SAP BPC dimension or model. SAP BPC provides the interface to create and maintain transformations in an Excel file. The transformation file includes three sections:

- Options
- Mapping
- Conversion

Options

The Options section contains the options you want to use when interpreting the data from the external source. You can specify the options shown in Table 4.1 in the Options section.

Option	Default Value	Description
`AmountDecimal Point=<value>`	Period	This option specifies the character to represent a decimal point. The default value to represent a decimal point is a "." (period). This is useful in countries where a value other than a period is used to denote a decimal point.
`ConvertAmount Dim=<dim_name>`	Account	This option can specify a dimension for amount calculations. The default value for this option is the Account dimension. A formula should exist in the conversion file for the value specified here for the calculation to work.
`CREDITPOSITIVE= YES|NO`	YES	The default value for this option is YES. If this option is set to NO, all credit accounts (of type Liability/Owner's Equity and Income) will have a negative sign in the source file.

Table 4.1 Options

Option	Default Value	Description
FORMAT=DELIMITED/ FIXED/VARIANT		This is a required parameter. It specifies the format of the data in the input file.
DELIMITER	Comma	This option is applicable when the format option is set as DELIMITED. This option is relevant when loading data directly from a file. Specify a value that matches the format of the input file.
HEADER=YES/NO	YES	If the file includes a header row with column names, set the value of this option to YES. When a header is included, the column names in the header can be referred in the MAPPING section of the transformation file to map the column names to the technical name of the target dimension of the model.
MAXREJECTCOUNT	Empty string	This option is used for validating the data from the external source and indicates the number of rejected records that can be tolerated before the validation process ends. An empty string represents 500 records. A value of -1 indicates that processing should continue no matter how many rejected records are encountered. A positive value indicates the number of rejected records allowed before processing will end.
SUPPRESSCHARACTER	<TEXT>	This option will remove the value specified by <text> from the records. This is useful when certain pieces of text are not required when loading the data from an external source.

Table 4.1 Options (Cont.)

Option	Default Value	Description
ROUNDAMOUNT = <INTEGER_VALUE>	Integer	This option rounds amount fields to the value specified in this option. The value for this option should be an integer number.
SKIP=<INTEGER_VALUE>	Integer	This option skips the specified number of records from the top of the file. The value should be an integer number.
SKIPIF=<TEXT_VALUE1>|<TEXT_VALUE2>	Empty string	When text is specified for this option, a line will be skipped from the data file if it begins with the specified value. More than one text value can be specified when using this option, separated by the pipe delimiter.
SELECTION=<Dimension1_techname>,<Dimension1_value>;<Dimension2_techname>,<Dimension2_value>	Empty	This option is relevant only when extracting data from an InfoProvider in SAP NetWeaver BW. The dimension name is the name of the InfoObject used in the cube. The dimension value is the value for the InfoObject to be used when extracting the data. This option is useful when you need to selectively extract data from an InfoCube. Assuming a particular InfoCube includes two InfoObjects, Account and Material, the following selection will restrict the data into SAP BPC for the COGS account and the M1 material: SELECTION=0ACCOUNT, COGS; 0MATERIAL, M1.
FORMULA=		A dimension can include a formula property to use in calculations. SAP BPC provides functions that can be used in formulas. They are referred to as K2 functions. When functions are used in dimension formulas, the library file that includes these functions is specified here.

Table 4.1 Options (Cont.)

Option	Default Value	Description
CONVERT_ INTERNAL=YES|NO		A conversion sheet can be used for each dimension to convert values coming from an external DataSource. The conversion sheet can have an internal value and an external value for a dimension member. The external value is the data coming from the external system. The internal value is how it would be stored in SAP BPC. When this is set, the external value will be converted to the internal value when loading data. This option indicates whether the dimension member names should be compared with the internal names. This should be always set to No.

Table 4.1 Options (Cont.)

The following is an example of the OPTIONS section in the transformation file:

```
*OPTIONS
FORMAT=DELIMITED
HEADER=YES
DELIMITER=','
AMOUNTDECIMALPOINT=
SKIP=0
SKIPIF=
VALIDATERECORDS=YES
MAXREJECTCOUNT=
ROUNDAMOUNT=
```

Mapping

The MAPPING section specifies how external data is mapped to an SAP BPC dimension or model. The mapping is defined by specifying a dimension to the left and assigning it to the name of a column or the column number from an external file to the right. The name of the InfoObject is specified when data is extracted from an InfoProvider like an InfoCube.

The example that follows specifies the format to use when mapping a dimension to a column value from an external file. In this case, `Entity` is the name of the dimension in the model, and `Company_Code` is the column name for the data coming from the flat file. For this to work, the OPTIONS section should have the value for header parameter set to YES, and the file should have a column named `Company_Code`.

```
Entity=Company_Code
```

The following is an example of a mapping when data is loaded from an InfoCube in SAP NetWeaver BW. In this case, `Entity` is the name of the dimension in the model, and `0COMP_CODE` is the name of the InfoObject in the InfoCube.

```
Entity=0COMP_CODE
```

All dimensions in an SAP BPC model should be mapped in the MAPPING section of the transformation. It is possible that the flat file or data coming from an InfoProvider does not contain the values for all dimensions in a model. These dimensions should be assigned with the `*NEWCOL(<text_value>)` parameter in the MAPPING section, where `<text_value>` is the value for the dimensions. This ensures that a value is assigned to all of the dimensions during the load process.

Another example of a transformation that is often required when you're loading data from an external DataSource is to prefix the data coming from the source with a specific text string. For example, you may want to prefix all of the entity data coming from a flat file with the string `"RB"`. This can be accomplished by specifying the following in the MAPPING section:

```
Entity=*Str(RB) + Company_Code
```

An example of a MAPPING section is shown here:

```
*MAPPING
Account=Acct
DataSrc=New(FILELOAD)
Entity=*Str(RB) + Company_Code
Category=Category
Time=Period
```

You've now seen a few examples of how to use the MAPPING section. It supports more advanced features when you're assigning data. Table 4.2 describes the various mapping options.

Mapping Parameter	Description	Example
*COL(<NUMBER>)	This parameter can be set to assign the value of a dimension based on its position in the file. The <NUMBER> indicates the position of the column in the file.	Entity = *Col(4) This assigns the value for the Entity dimension from the fourth column in the data file.
*COL(<NUMBER>, <POS1>: <POS2>)	This parameter works the same way as the *COL(<NUMBER>) parameter and provides a feature to select only a subset of the column value. <POS1> and <POS2> indicate the start and end positions within the column.	Entity = *Col(4,1,3) This assigns the value for the Entity dimension from the fourth column in the data file. Only the value from the first three characters of this column will be used in the assignment.
*FCOL(A:B)	You can use this parameter when using fixed format data files to define starting and ending positions within a data row.	Entity = *FCOL(2:5) In this example, the values from character columns two to five in the data file will be assigned to the Entity dimension.
*MVAL	This parameter is especially useful when you're loading data from a BW InfoCube to an SAP BPC model. If a source record in the InfoCube has multiple key figures, you can use this parameter to map multiple key figures to different accounts.	ACCOUNT = *MVAL(0AMOUNT |*NEWCOL(AMOUNT) | |0QUANTITY | *NEWCOL(QUANTITY)) Let's say we're loading data from a BW InfoCube to an SAP BPC model. In this example, the source record in a BW InfoCube is translated to two records in the SAP BPC model. The data in key figure 0AMOUNT is mapped to ACCOUNT=AMOUNT, and the data in the 0QUANTITY key figure is mapped to ACCOUNT=QUANTITY.

Table 4.2 Mapping

Mapping Parameter	Description	Example
*NEWCOL(<TEXT_VALUE>)	In some cases, the data file may not contain the value for a dimension, and it may be necessary to set a constant value for the dimension.	AuditTrail=NEWCOL (FILELOAD) The value for the AuditTrail dimension is set to the constant value FILELOAD in this assignment.
*str(<TEXT_VALUE>)	This parameter can be set when it is necessary to concatenate the data coming from a data file with other values. The text string can be used to prefix or suffix the column value.	Entity = *Str (BPC) + *Col(4) This assignment prefixes the string BPC to the data coming from the data file for the Entity dimension.
*pad	This parameter has similar functionality as the *str parameter, but it can be used to prefix the data coming from a file. *pad(SAP)	Entity = *pad(SAP)
*if(condition1 (is true) action1; condition2 (is true) action2; action3)	If condition1 is true, action1 is performed. If condition1 is false, condition2 is executed. If condition2 is true, action2 is performed. If both condition1 and condition2 are not satisfied, action3 is performed.	Imagine that you have two columns, Source and Product, in the data file. You want to apply the following rules when mapping the data: If you have the value SAP in the source field, you want to take the value of the product as it is coming from the file. If you have the value LEGACY, you want to add two zeros to the product field. If neither SAP nor LEGACY is assigned to the source field, you want to assign the value Error to the product field.

Table 4.2 Mapping (Cont.)

Mapping Parameter	Description	Example
		This would look as follows: `Product=*IF (source= *Str(SAP) then product; source=*Str(LEGACY) then Str(00) + product;*Str(ERROR))`

Table 4.2 Mapping (Cont.)

Conversion File

A *conversion file* associates the external values of data to equivalent internal values of dimension members in SAP BPC. This is useful when data coming from a source does not match the data stored in SAP BPC for a dimension. For example, the data coming from a file or an InfoProvider may not match how a dimension member value is stored in the SAP BPC system, although they may mean the same. Using the conversion files, you can map the external values to the internal values.

When this conversion is required, you create a separate file or use a new spreadsheet for each dimension to define the conversions as necessary. The conversion sheet includes three columns: EXTERNAL, INTERNAL, and FORMULA. The EXTERNAL column identifies how the data is coming from the external system. The INTERNAL column defines how the data is stored in SAP BPC. The FORMULA column can be added if any calculations are required. It is not mandatory to have a conversion file and is required only when there are conversions between external and internal values. For example, the following conversions can be defined for the Time dimension data:

- 201101 can be defined as the EXTERNAL column.
- 2011.01 can be defined as the corresponding INTERNAL column.

When data is loaded into a model that includes this conversion, any data that includes 201101 for the Time dimension from an external data source will be interpreted as 2011.01 in SAP BPC. You can also use the asterisk (*) and question mark (?) as wildcards in the EXTERNAL or INTERNAL columns. An asterisk (*) refers to any string of characters, whereas a question mark (?) refers to a single character. You can skip a record containing an external value by setting the internal value as *SKIP. You can use the Formula column to represent any calculations. For example, you can have

a formula to increase the revenue member of the Account dimension by 10%. This formula would read *value 1.10*. After the transformation and conversion files are created, the data can be loaded into the SAP BPC dimension or model.

The actual process of loading data into SAP BPC is performed using a Data Manager package. We'll discuss this next.

4.2.2 Data Manager Packages

Data Manager packages in SAP BPC support a variety of tasks related to loading and maintaining data in a dimension or model. When a Data Manager package is executed, it in turn executes a corresponding process chain in SAP NetWeaver BW. A *process chain* automates the process of loading data in SAP NetWeaver BW. With the introduction of SAP BPC for NetWeaver, new process types have been introduced in process chains to enable loading and managing of the data for the SAP BPC environment. Each Data Manager package is associated with a process chain in the SAP NetWeaver BW system. Process chains automate the process of loading and managing data in the SAP NetWeaver BW system.

Starting with EPM 10.0, the SAP BPC Data Manager for NetWeaver supports loading data directly from an SAP ERP system in Full and Delta modes through SAP-delivered extractors. The Data Manager also supports loading data from a BW InfoProvider in Full and Delta modes. Users can also specify the selection criteria when loading data from an SAP ERP system and BW InfoProvider into SAP BPC. Delta load is not available when loading data from a virtual InfoCube or MultiProvider.

Four types of Data Manager packages are available to perform different tasks in SAP BPC:

- **Data Manager packages**
  This type of package is used for managing data—such as loading dimension and model data from flat files and InfoProviders in SAP NetWeaver BW, copying data, clearing data, and so on.
- **Financial process packages**
  Financial process packages can be used for executing specific functions related to planning and consolidation models. They can be used for executing tasks related to allocation, currency translation, intercompany elimination, and consolidation functions.

- **System administration packages**
  These packages are related to performing system administrative tasks, such as running Light Optimize, Full Optimize, and so on.
- **Miscellaneous packages**
  Miscellaneous packages serve more of a utility value for SAP BPC models. The import file and send mail Data Manager packages fall under this type and are useful for importing a file into SAP BPC and then sending an email to users. For this to work, email services should be configured in SAP BPC.

Note

Table 4.3 through Table 4.6 list the common Data Manager packages that are used in SAP BPC. If you do not see a Data Manager package you want to use listed in one of these tables, you can easily create a new one in SAP BPC and associate it with a process chain that will perform a specific function.

You can also create custom folders to store Data Manager packages.

The packages and process chains outlined in the following tables are delivered with the standard Data Manager packages.

Data Manager Package	Description	Process Chain
Import master data	This package is used for loading master data from flat files into an SAP BPC dimension.	/CPMB/IMPORT_MASTER
Import master data description	This package is used to load text data into a dimension.	/CPMB/IMPORT_DESCRIPT
Import transaction data	This package is used for loading transaction data using a flat file. The transaction file should be staged in the Application folder before it can be used in the Data Manager package.	/CPMB/IMPORT

Table 4.3 Data Manager Packages

Data Manager Package	Description	Process Chain
	A transformation file is necessary to load transaction data in SAP BPC for NetWeaver. The transformation file's reference in the transformation file should be available in the respective folder before the package is executed.	
Import transaction data from InfoProvider	This package is used to load data from SAP NetWeaver BW InfoProviders. The data can be extracted from different types of InfoProviders—InfoCubes, DSOs, and MultiProviders. When using the package, the selection criteria to use to load data are specified in the transformation file.	/CPMB/LOAD_INFOPROVIDER
Import transaction data from InfoProvider (enhanced user interface)	This package is available to load data from SAP NetWeaver BW InfoProviders. The selection criteria to use to load data can be dynamically specified when executing the package.	/CPMB/LOAD_INFOPROV_UI
Import InfoObject master data	This package is used to load master data from an SAP NetWeaver BW InfoObject to an SAP BPC dimension. This process chain eliminates the need to set up an open hub destination and custom process that was used in SAP BPC 7.0 for NetWeaver to transfer master data from SAP NetWeaver BW to SAP BPC.	/CPMB/IMPORT_IOBJ_MASTER
Import BW InfoObject master data hierarchy	This package is used to assign PARENTH% hierarchies in SAP BPC dimension member sheets.	/CPMB/IMPORT_IOBJ_HIER

Table 4.3 Data Manager Packages (Cont.)

Data Manager Package	Description	Process Chain
Move transaction data	This package enables you to move data within a model. A selection screen displays during the execution of the package. The following is the sequence of steps when this package is run: ▶ The data selected for the destination is removed from the corresponding InfoProvider. ▶ The data from the source is moved based on the selection. ▶ The data is cleared from the source for the selected data.	/CPMB/MOVE
Clear transaction data in InfoCube	This package enables you to clear or delete data in an SAP BPC model. A selection screen displays during the execution of the package. On the selection screen, you must select a value for at least one dimension.	/CPMB/CLEAR
Copy transaction data	This package is used when you must copy transaction data within a model. A selection screen displays during the execution of the package. This is very useful when you want to create a baseline of plan data for the next year. You can do this by copying the current year's actual data to next year's plan data. For example, the actual sales data for 2011 can be used as the baseline data for planning 2012 sales. You can achieve this by using this package.	/CPMB/COPY

Table 4.3 Data Manager Packages (Cont.)

Data Manager Package	Description	Process Chain
Append transaction data	Executing this package does not affect the existing data. The data is appended to the existing model. This is available only when using a flat file as the data source.	/CPMB/APPEND
Copy comments	This is a new Data Manager package introduced in EPM 10.0. It provides the ability to copy comments from a source to a target. For example, all of the comments can be copied from the Actual to the Plan member in the Category dimension. All comments across every other dimension will then be copied from the Actual category to the Plan category.	/CPMB/COPYCOMMENTS
Import transaction data from BW InfoProvider (delta)	This is a new Data Manager package introduced in EPM 10.0. It provides the ability to load delta data from an InfoProvider in BW.	/CPMB/LOAD_DELTA_IP
Import transaction data from SAP source system (full)	This is a new Data Manager package introduced in EPM 10.0. It provides the ability to load transaction data from an SAP system.	/CPMB/LOAD_DS_TD_FULL
Import master data from SAP source system (full)	This is a new Data Manager package introduced in EPM 10.0. It provides the ability to load master data from an SAP system.	/CPMB/LOAD_DATASRC_MD
Trigger BW process chain	This package allows you to trigger the execution of an SAP NetWeaver BW process chain. When you run the Data Manager package, the system prompts you for the process chain that needs to be triggered.	/CPMB/TRIGGER_BW_CHAIN

Table 4.3 Data Manager Packages (Cont.)

Financial Process Package	Description	Process Chain
Allocation	Used for executing the allocation logic.	/CPMB/ALLOCATION
FX restatement	Runs the currency translation logic.	/CPMB/FX_RESTATMENT
IC elimination	Used to eliminate the intercompany transactions of the business.	/CPMB/IC_ELIMINATION
IC data	Runs the ICDATA logic that creates entries for entities to reconcile intercompany transactions with their trading partners.	/CPMB/ICDATA
IC booking	Runs the IC booking logic that posts mismatches in intercompany transactions between entities and trading partners using the ICBOOKING business rule.	/CPMB/ICBOOKING
Opening balances	Runs the carry forward balance business rule.	/CPMB/ OPENING_BALANCES
Calculate ownership logic	Runs the calculate ownership logic. This is applicable for consolidation models.	/CPMB/OWNERSHIPCALC
Consolidation	Used for running the logic for performing financial consolidation.	/CPMB/ LEGAL_CONSOLIDATION
Run CalcAccount	Runs the account transformation business rule.	/CPMB/RUNCALCACCOUNT
Clear journal table	Clears the journal table.	/CPMB/CLEAR_JOURNALS
Export journal table	Exports the journal table to an output file.	/CPMB/EXPORT_JOURNAL
Restore journal table	Restores the journal table from a file.	/CPMB/ RESTORE_JOURNALS

Table 4.4 Financial Processes Packages

System Administrative Package	Description	Process Chain
Validate logic file	Validates a logic file for syntax.	/CPMB/ADMINTASK_VALIDATE
Archive audit activity	Archives audit activity. When audit activity is enabled, a lot of data may be generated. This data has to be archived periodically.	/CPMB/ARCHIVE_ACTIVITY
Archive audit data	Archives model data.	/CPMB/ARCHIVE_DATA
Clear comments	Clears comments from the comments table.	/CPMB/CLEARCOMMENTS
Light Optimize	Runs the Light Optimize process for a model. This process closes the open request, creates indexes, and updates database statistics. When the Light Optimize package is run from the Office Client or Web Client, it triggers a job to immediately execute this process chain.	/CPMB/LIGHT_OPTIMIZE
Full Optimize	Works like the Light Optimize process. In addition, the data model of the InfoCube is also optimized.	/CPMB/FULL_OPTMIZE

Table 4.5 System Administrative Packages

Miscellaneous Package	Description	Process Chain
Import and send mail	Imports a file and emails the result to a list of users.	/CPMB/IMPORT_A_SEND_EMAIL
Import using FTP	Imports a file into the current model.	/CPMB/IMPORT_USING_FTP

Table 4.6 Miscellaneous Packages

4.2.3 Process Chains

You use *process chains* to automate the process of extracting, transforming, and loading data and to provide different process types for managing data within the SAP NetWeaver BW and SAP BPC systems. To enable common tasks to be executed in SAP BPC, SAP provides several standard process chains as out-of-the-box functionality, ready to be used. The process chains provided by SAP have the prefix `/CPMB` and can also be enabled as Business Content. These process chains, created in SAP NetWeaver BW, are wrapped into a Data Manager package in SAP BPC.

You create and maintain the process chains using Transaction RSPC in the SAP NetWeaver BW system. Subsequently, Data Manager packages are created in SAP BPC and associated with the process chains created in SAP NetWeaver BW. Table 4.3 through Table 4.6 in the previous section list the process chains associated with Data Manager packages.

As mentioned, process chains used for SAP BPC models are associated with the `/CPMB` namespace. You can also create custom process chains based on standard process chains and adapt them to your requirements. Several process types are specifically provided for use in SAP BPC models.

Now that we've discussed transformation, conversion files, Data Manager packages, and process chains, we'll look into the process of loading data into a dimension using a flat file.

4.2.4 Loading Master Data from a Flat File into a Dimension

Rich Bloom stores entity data in a flat file. This data must be loaded from the flat file into the entity dimension. Before you can load transaction data into a model, you need to load master data into a dimension. If a master data record (dimension member) is not available in a dimension when you load transaction data to a model, the transaction data load will fail.

We'll now load the entity data for Rich Bloom from a flat file into the RB_ENTITY dimension. The flat file we'll use is a comma-delimited file and contains the entity ID, description, and currency of the entity. *Entity_Data.csv* is available for download at this book's page at *www.sap-press.com.* Proceed as follows:

1. Log in to the SAP BPC for NetWeaver Web Client and connect to the ZRB_PR_PLAN environment. The EPM add-in for Microsoft Office software is required

for reporting and using Data Manager packages. If you have not done so already, download the EPM add-in client to your desktop by selecting Download Center (Figure 4.15, ❶). This will open a new window and provide the link to install the EPM add-in client (Figure 4.15, ❷).

2. After the EPM add-in is installed, confirm you are connected to the ZRB_PR_PLAN environment (Figure 4.15, ❸). Then click on EPM Office Add-In Excel to launch the EPM add-in for Excel (Figure 4.15, ❹).

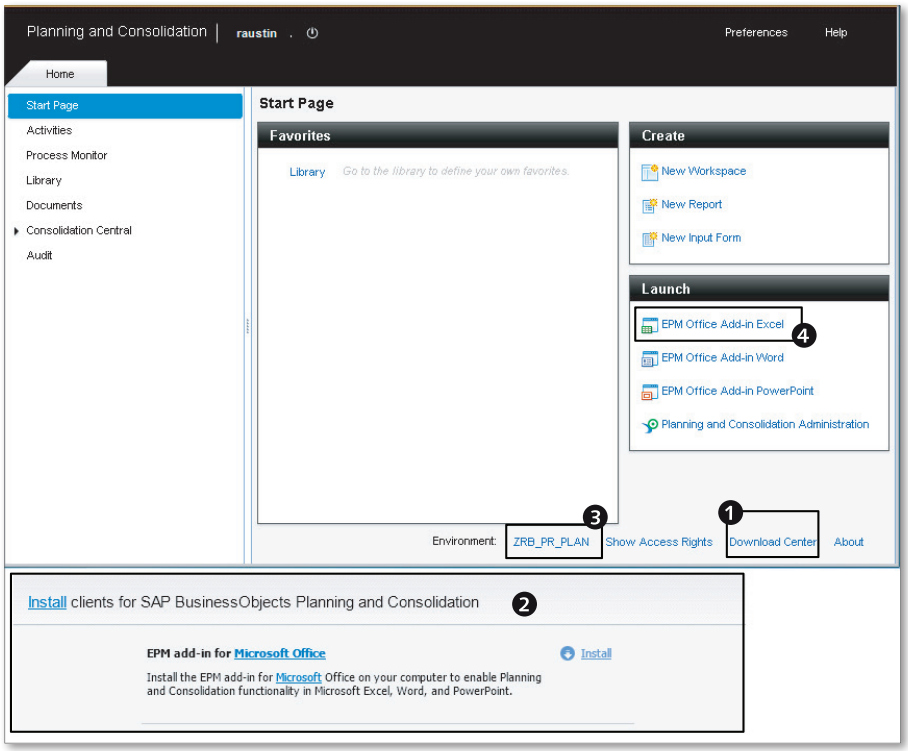


Figure 4.15 Loading Master Data from a Flat File into a Dimension—Part A

3. The system automatically authenticates your credentials inside the SAP EPM add-in client for BPC, displays the list of models in the environment, and lets you select a model to connect to. Select the ZRB_SALES_CMB model that you created in Chapter 3 to use for storing sales actuals and plan data (Figure 4.16, ❺).

4. Click on OK to connect to the model (Figure 4.16, ❻). A connection is established to the model. You will see two tabs applicable for EPM systems displayed inside Excel: EPM and Data Manager. (So far, you have seen how to log in directly to the EPM add-in interface from the Web Client without manually specifying the connection parameters for the SAP BPC system. In Chapter 5, you will see how to create a manual connection to the SAP BPC system inside the EPM add-in.)

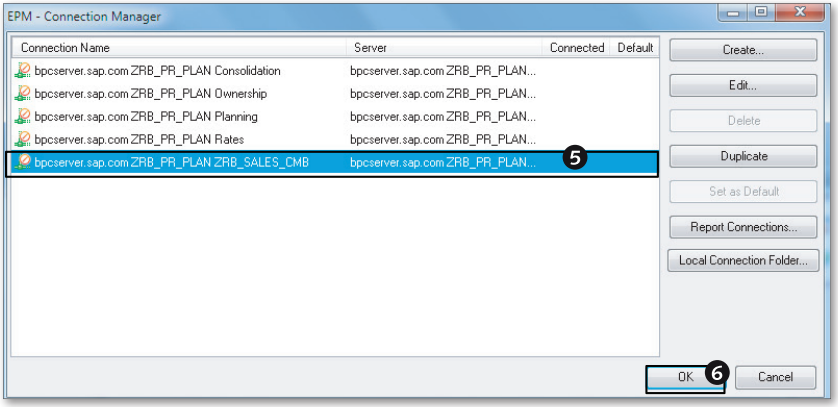


Figure 4.16 Loading Master Data from a Flat File into a Dimension—Part B

5. The data file that you want to load to the BPC server is in your desktop (Figure 4.17), which you'll access by selecting the DATA MANAGER tab (Figure 4.18, ❼). This tab lists the options available for executing Data Manager tasks. The options displayed are based on the authorizations you have been granted. The first step in the process is uploading the flat file containing entity data that is on your desktop to the Data Files folder of the environment/model. Click on UPLOAD DATA (Figure 4.18, ❽).
6. In the DATA MANAGER—UPLOAD dialog box, select the BROWSE option to locate and select the file on the desktop (Figure 4.18, ❾ and ❿). After selecting the file, click on UPLOAD (Figure 4.18, ⓫).
7. This opens the SAVE dialog box. Select the folder, and enter the name of the destination file to store in the BPC server (Figure 4.18, ⓬). After entering the destination file, click on SAVE (Figure 4.18, ⓭). The entity file is stored in the SAP BPC server under the selected folder.

```
20      Rich Bloom Inc., San Diego,CA,USA       USD
22      Rich Bloom Corporate Inc., Philadelphia,PA,USA  USD
23      Rich Bloom New Markets Inc., Houston,TX, USA    USD
25      Rich Bloom Ltd,London, UK       GBP
30      Rich Bloom AG,Frankfurt, Germany        EUR
```

Figure 4.17 Loading Master Data from a Flat File into a Dimension—Part C

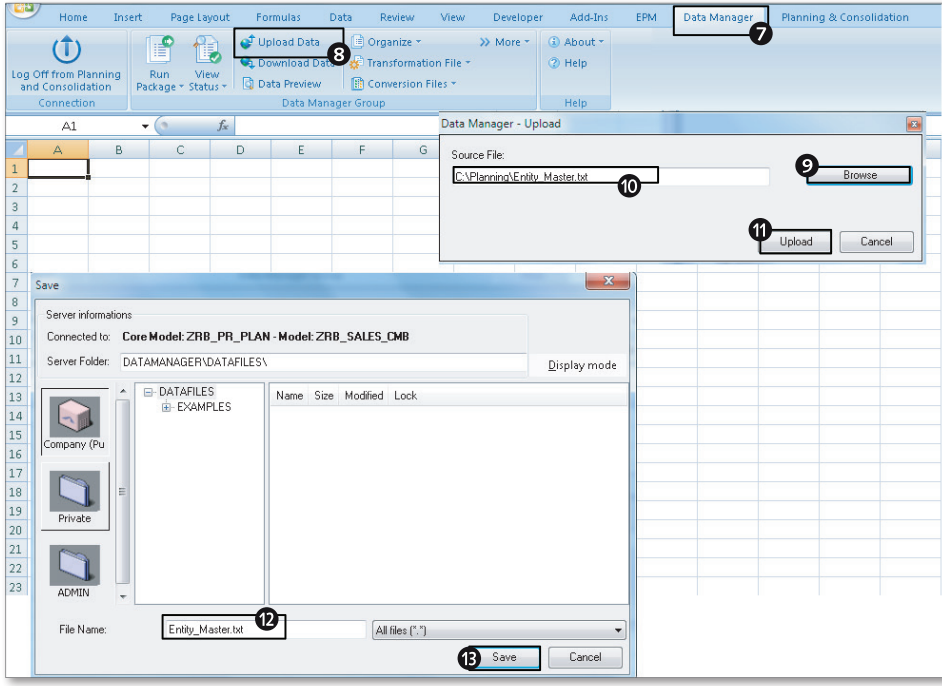


Figure 4.18 Loading Master Data from a Flat File into a Dimension—Part D

8. You're now ready to create a transformation file to define the structure of the file you'll use to load data into the Entity dimension. From the TRANSFORMATION FILE menu option, select NEW TRANSFORMATION FILE (Figure 4.19, ⓮).
9. The flat file we're using to load data is a tab-delimited file that contains the entity ID, description, and currency for the entity data. The mapping for the data is defined in the *MAPPING section. The delimiter for the data is specified as TAB

in the *Options section. No data conversion is involved, so we won't use the Conversion section. The transformation file is defined as shown in Figure 4.19 (⓯). From the Transformation File menu option, select Validate & Process Transformation File to validate and save the transformation file (Figure 4.19, ⓰). This opens the Validate & Process Transformation dialog box (Figure 4.20).

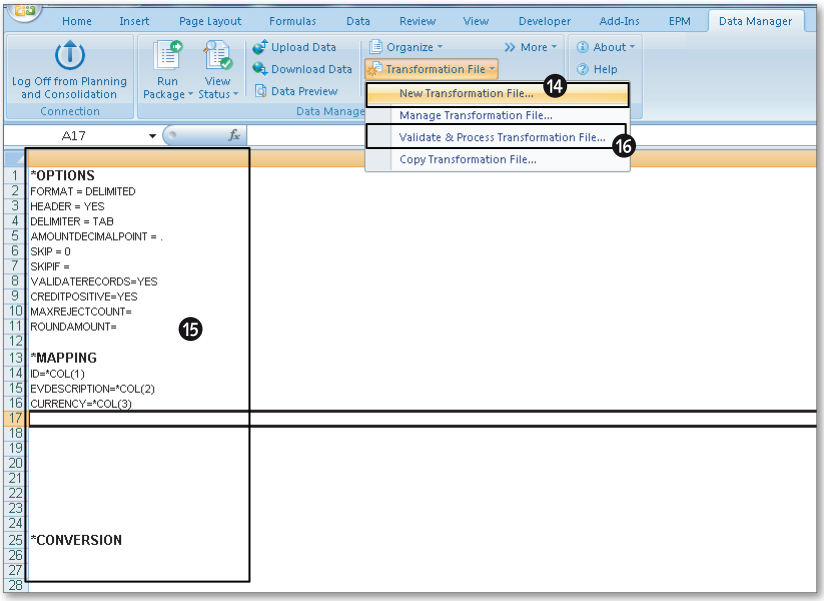


Figure 4.19 Loading Master Data from a Flat File into a Dimension—Part E

10. In the Validate and Process Transformation dialog box, select the Data Type for which this transformation file should be used (Figure 4.20, ⓱), the Data file that should be used as the source for verifying this load (Figure 4.19, ⓲ and ⓳), and the technical name of the dimension into which the data should be loaded (Figure 4.20, ⓴). Click on Save (Figure 4.20, ㉑).

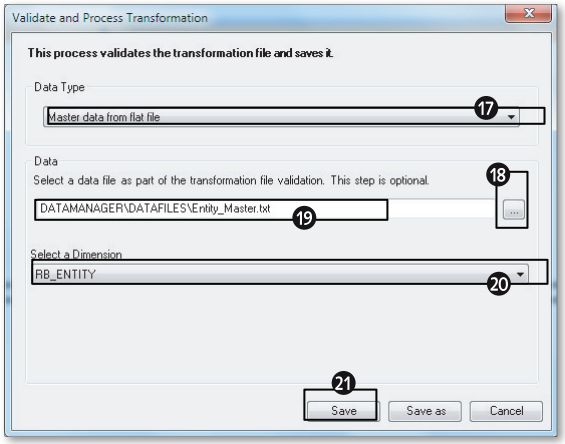


Figure 4.20 Loading Master Data from a Flat File into a Dimension—Part F

11. The system validates the transformation file and prompts you to save the transformation file in the BPC server (Figure 4.21, ㉒). Save the transformation file as "TR_MAS_ENTITY.XLS". The system then outputs a log of the validation (Figure 4.22).

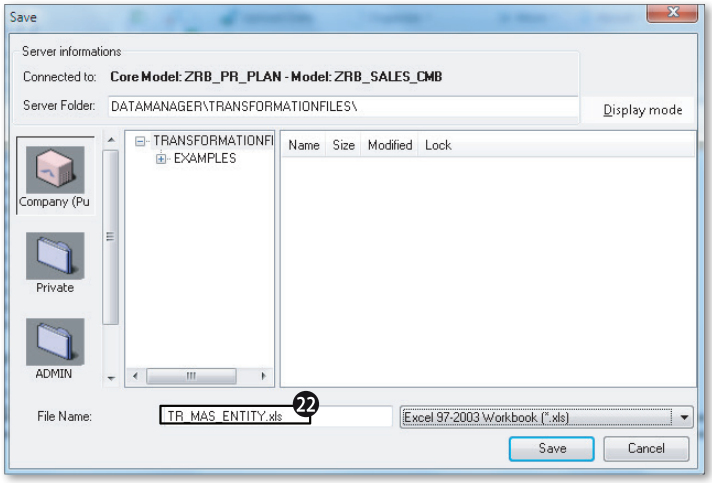


Figure 4.21 Loading Master Data from a Flat File into a Dimension—Part G

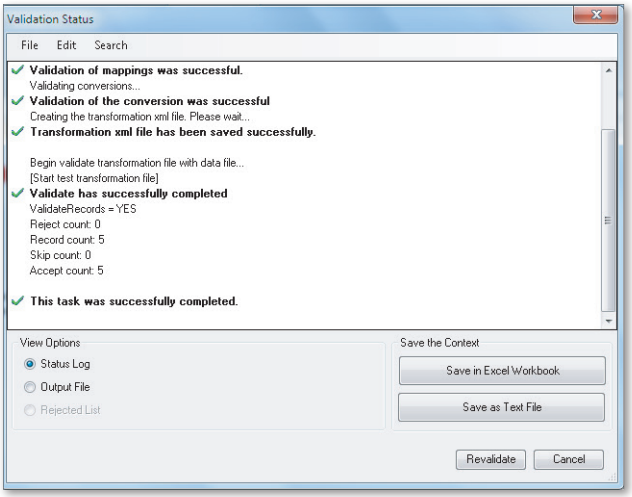


Figure 4.22 Loading Master Data from a Flat File into a Dimension—Part H

12. You're now ready to execute the Data Manager package to load the master data for entities. Select Run Package to run a Data Manager package (Figure 4.23). In the dialog box that lists the Data Manager packages, select the Data Management group, and execute the ImportMasterData Attrib and Hier Flat File Data Manager package (Figure 4.23, ㉓). This Data Manager package uses the `/CPMB/IMPORT_MASTER` process chain. Then click on Run (Figure 4.23, ㉔). This opens the Run Package dialog box to select the file to load data (Figure 4.24).

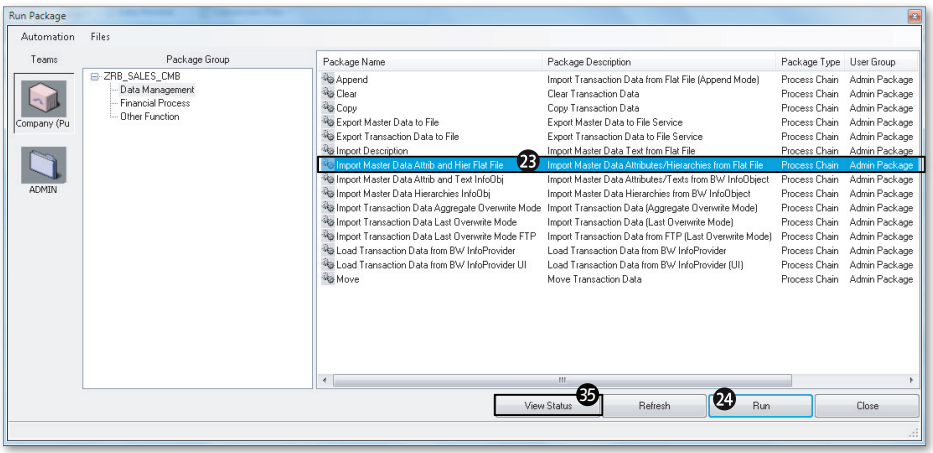


Figure 4.23 Loading Master Data from a Flat File into a Dimension—Part I

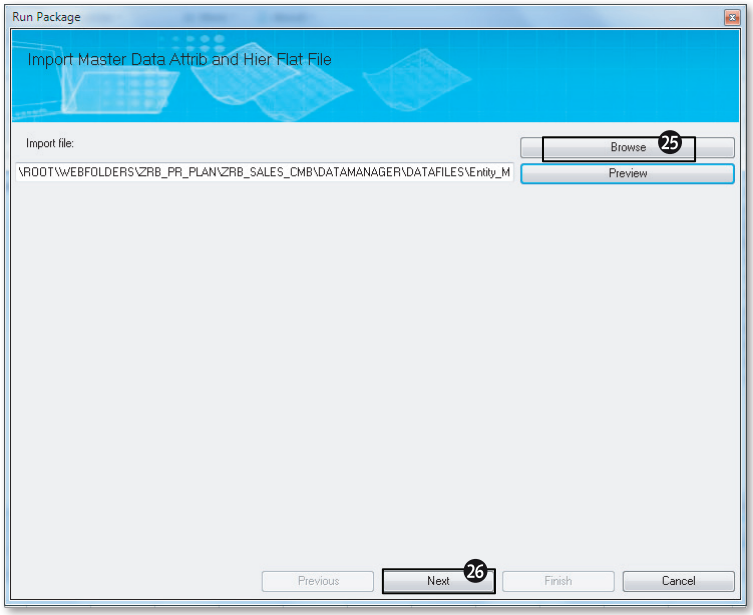


Figure 4.24 Loading Master Data from a Flat File into a Dimension—Part J

13. In the RUN PACKAGE dialog box, select the import file that should be loaded (Figure 4.24, ㉕). You can also preview the data file you'll load by clicking on the PREVIEW button. Then click on NEXT (Figure 4.24, ㉖). This opens a new RUN PACKAGE dialog box to select the transformation file for loading the data (Figure 4.25).

14. Select the transformation file TR_MAS_ENTITY.XLS that you created in the earlier step to use for loading the data (Figure 4.25, ㉗). Then click on NEXT (Figure 4.25, ㉘). This opens the RUN PACKAGE dialog box to select the dimension to load the data into (Figure 4.26).

15. Select RB_ENTITY as the dimension to load the data from the list of dimensions in the dropdown box (Figure 4.26, ㉙). Then click on NEXT (Figure 4.26, ㉚). This opens the RUN PACKAGE dialog box to select the option to load hierarchy data (Figure 4.27).

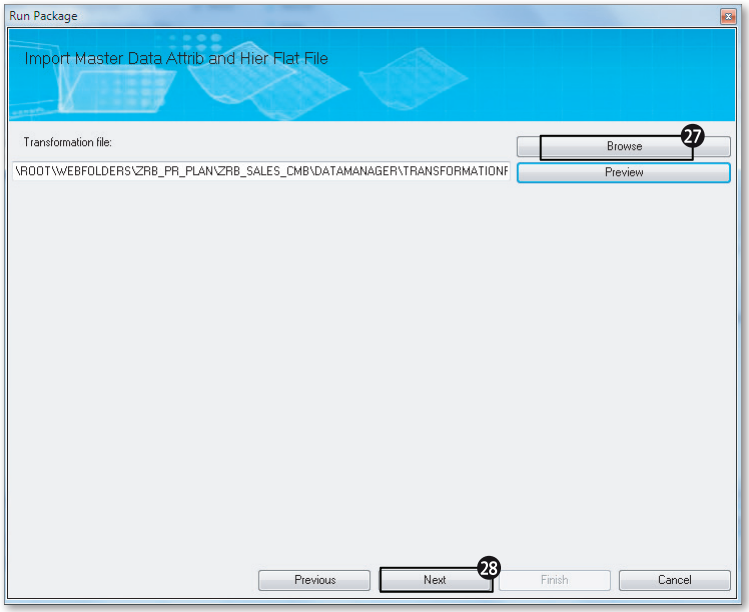


Figure 4.25 Loading Master Data from a Flat File into a Dimension—Part K

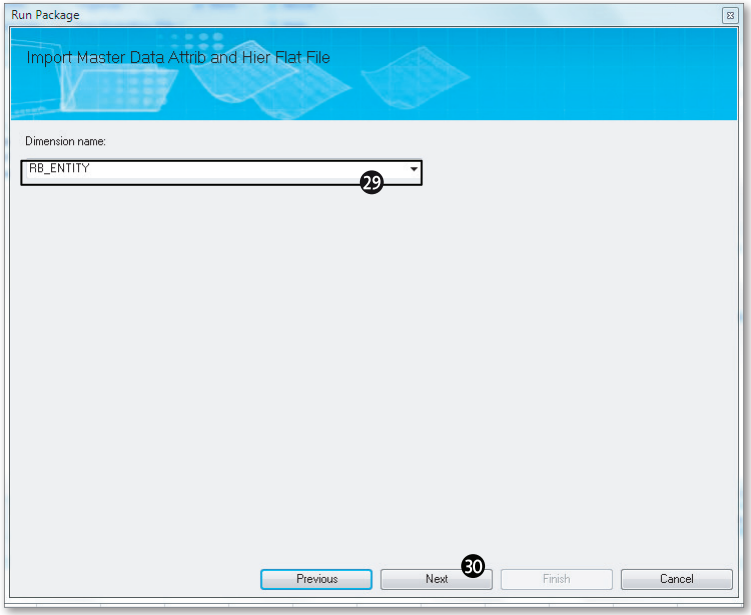


Figure 4.26 Loading Master Data from a Flat File into a Dimension—Part L

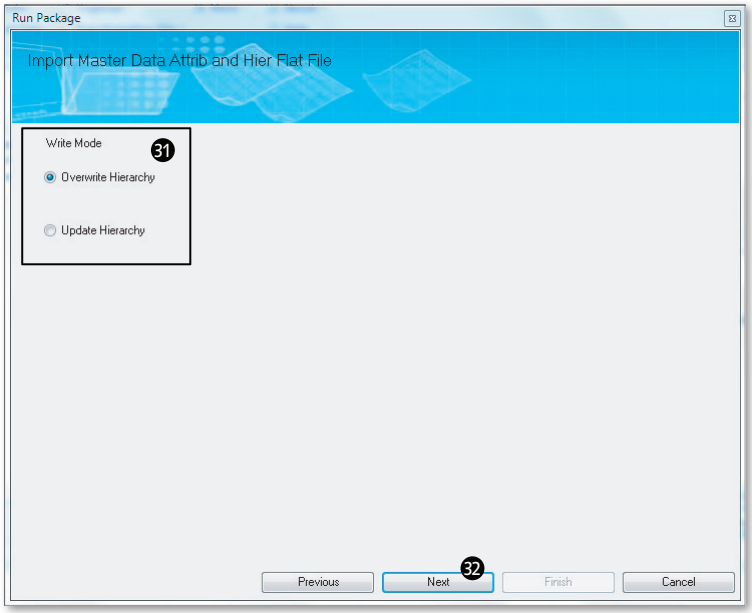


Figure 4.27 Loading Master Data from a Flat File into a Dimension—Part M

16. We are not loading hierarchy data in this example, so it doesn't matter which option you choose here. (You'll learn how to load hierarchy data later in this chapter.) In this example, the OVERWRITE HIERARCHY option is selected (Figure 4.27, ❸❶). Then click on NEXT (Figure 4.27, ❸❷). This opens the RUN PACKAGE dialog box to select the option to use to run the Data Manager package (Figure 4.28).

17. This is the last step before the Data Manager is executed. Select RUN, and click on FINISH (Figure 4.28, ❸❸ and ❸❹). Alternatively, if you want this to be loaded at a different time, you can select the SCHEDULE option instead of running the package now.

18. Click the VIEW STATUS button to view the status of the load (shown earlier in Figure 4.23, ❸❺). The job is scheduled as a background job, and the status of the job is refreshed every *n* seconds if the REFRESH STATUS EVERY N SECONDS checkbox is selected (Figure 4.29). The status is displayed as RUNNING, SUCCEED, or ERROR. You can display the log for the package by selecting the package and clicking on the DETAIL button (Figure 4.29, ❸❻ and ❸❼). Figure 4.30 displays the detailed log of the package results.

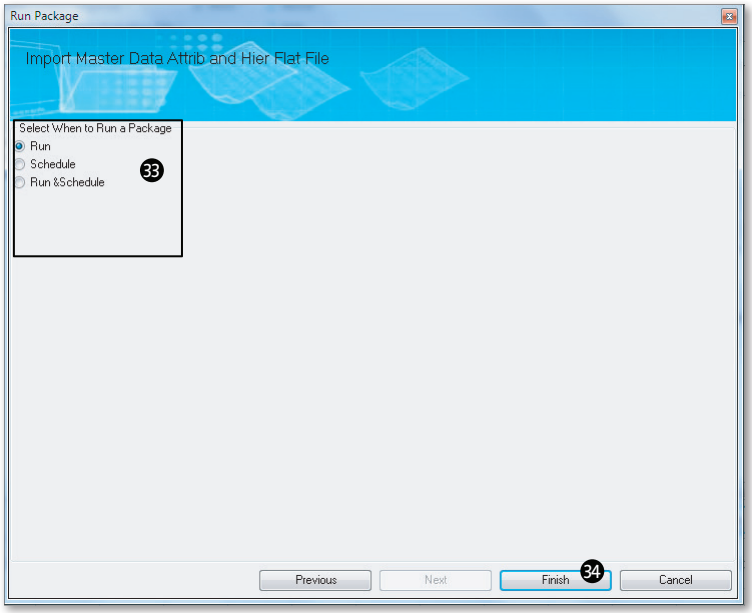


Figure 4.28 Loading Master Data from a Flat File into a Dimension—Part N

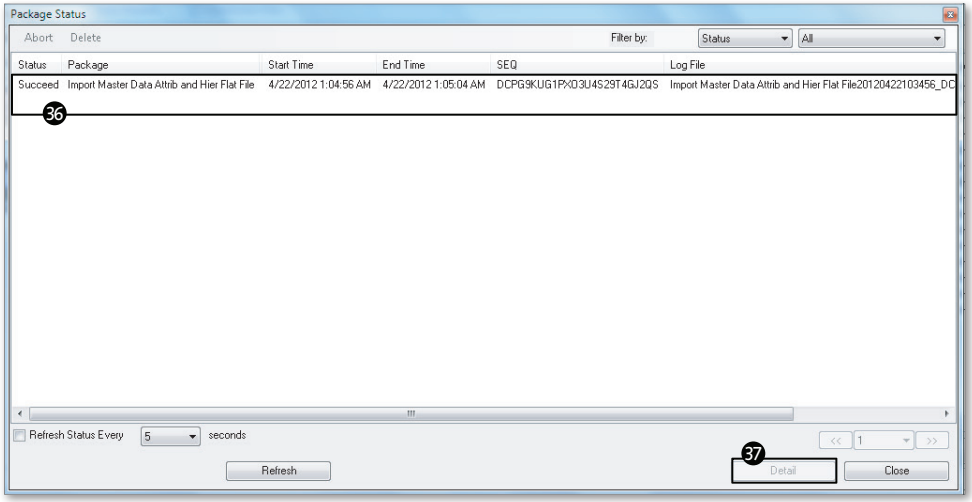


Figure 4.29 Loading Master Data from a Flat File into a Dimension—Part O

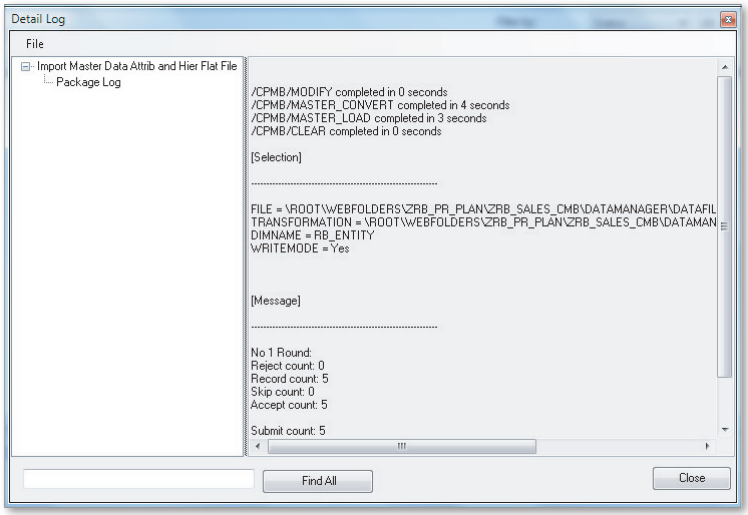


Figure 4.30 Loading Master Data from a Flat File into a Dimension—Part P

In this example, you've seen how to load master data from a flat file into an Entity dimension. In Chapter 3, we discussed manually entering data into a dimension. The steps outlined in this section can be used to load data to dimensions. The master data for dimensions as it appears in SAP BPC is available for download at this book's website at *www.sap-press.com.* The name of the file is *Dimension Members.xls*.

In the next section, we'll load the master and hierarchy data from the ZACCOUNT InfoObject in SAP NetWeaver BW into the RB_ACCOUNT dimension in SAP BPC.

4.2.5 Loading Master Data from an InfoObject into an SAP BPC Dimension

Rich Bloom stores its account data in the BW system. This data must be loaded from the ZACCOUNT InfoObject as master data into the RB_ACCOUNT dimension in SAP BPC. In this section, you will learn how to load master data members to a dimension. In the next section, you will learn how to load hierarchy data for the dimension. Follow these steps:

1. Log in to the Web Client, and connect to the ZRB_PR_PLAN environment. From the Web Client, launch the EPM add-in for Excel, and connect to the ZRB_SALES_CMB model.

2. Begin by creating a new transformation file (Figure 4.31, ❶). The first command included in the *MAPPING section takes the characteristic value of an InfoObject in SAP NetWeaver BW and assigns it as a dimension member in SAP BPC. The second command in the *MAPPING section takes the first two characters of the value in the ZACCOUNT InfoObject and assigns them to the ACCTYPE property. Every member in the Account dimension should contain a valid value for the ACCTYPE property. The ACCTYPE property contains the values AST, LEQ, INC, or EXP to denote the type of account.
3. Save the transformation file as "TR_MAS_ACCOUNT_BW.XLS". We will use a conversion file to further transform the value assigned to the ACCTYPE property. The third command in the *MAPPING section assigns a constant value of AVG to the RATETYPE property. The RATETYPE property is used in currency translation.
4. Select the NEW CONVERSION FILE option under the CONVERSION FILES menu to create a conversion file (Figure 4.31, ❷). This file is created to convert the values mapped to the ACCTYPE property.

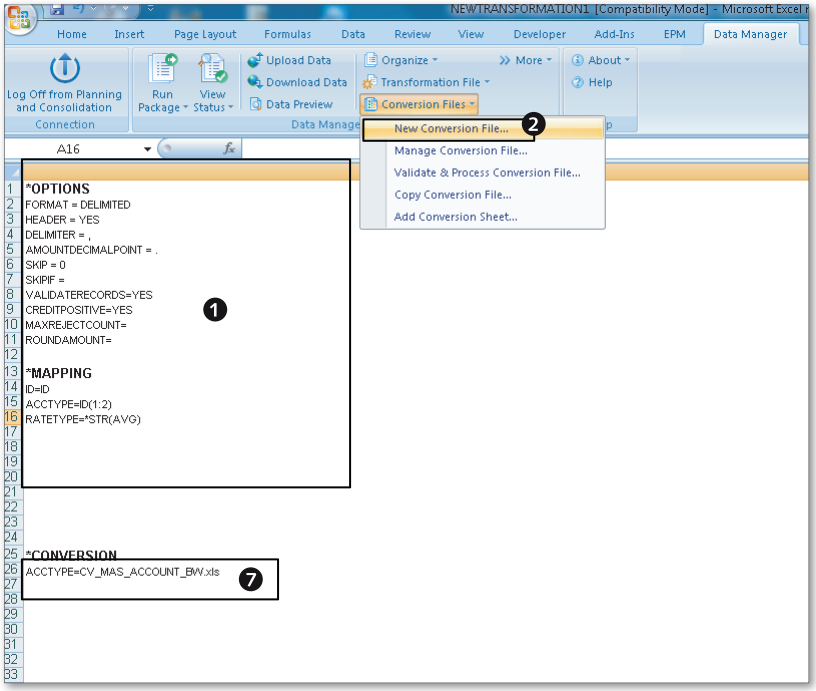


Figure 4.31 Loading Data from an InfoObject into a Dimension—Part A

5. Enter external values in the EXTERNAL column and the corresponding values in the INTERNAL column (Figure 4.32, ❸). The value in the EXTERNAL column is converted to the corresponding INTERNAL column value when converting the data. For example, any value in the `ACCTYPE` property that begins with "1" is converted to the value `AST`.
6. Select the Validate & Process Conversion File option under the Conversion Files menu to validate, and then save the conversion file (Figure 4.32, ❹). This opens the Save dialog box.
7. In the Save dialog box, enter the name of the conversion file "CV_MAS_ACCOUNT_BW", and click on Save (Figure 4.32, ❺ and ❻).
8. Include the reference to the conversion file you created in the previous step in the conversion section of the transformation file (refer to Figure 4.31, ❼). When the transformation file is used, the conversion file included here will convert the data for the account type property.

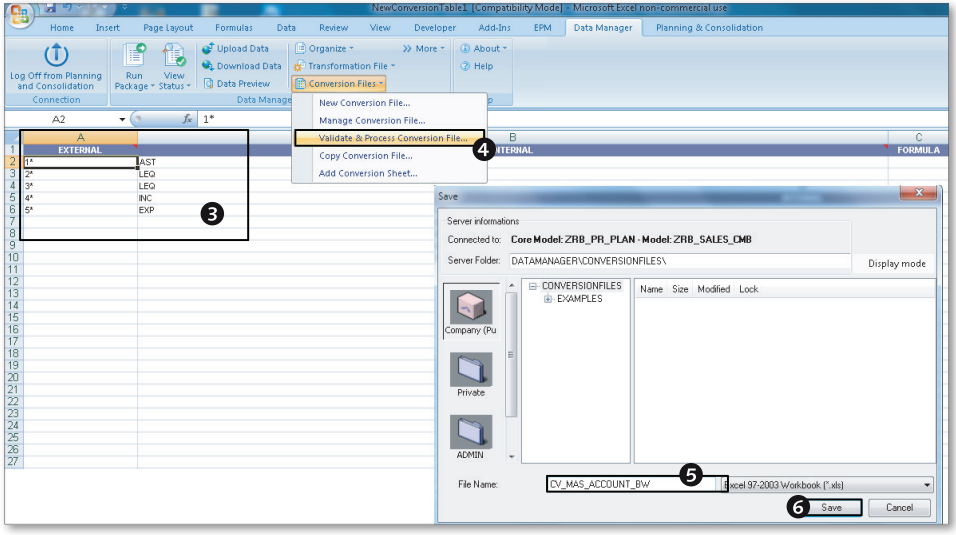


Figure 4.32 Loading Data from an InfoObject into a Dimension—Part B

9. You're now ready to execute the Data Manager package to load the master data for entities. Select Run Package to run a Data Manager package. In the dialog box that lists the Data Manager packages, select the Data Management group,

and execute the ImportMasterData Attrib and Text InfoObj Data Manager package (Figure 4.33, ❽). This Data Manager package uses the `/CPMB/IMPORT_IOBJ_MASTER` process chain and is available to load master data from an SAP NetWeaver BW InfoObject. Click on Run (Figure 4.33, ❾) to open the Run Package dialog box that provides the options to select the InfoObject and values to use in loading data.

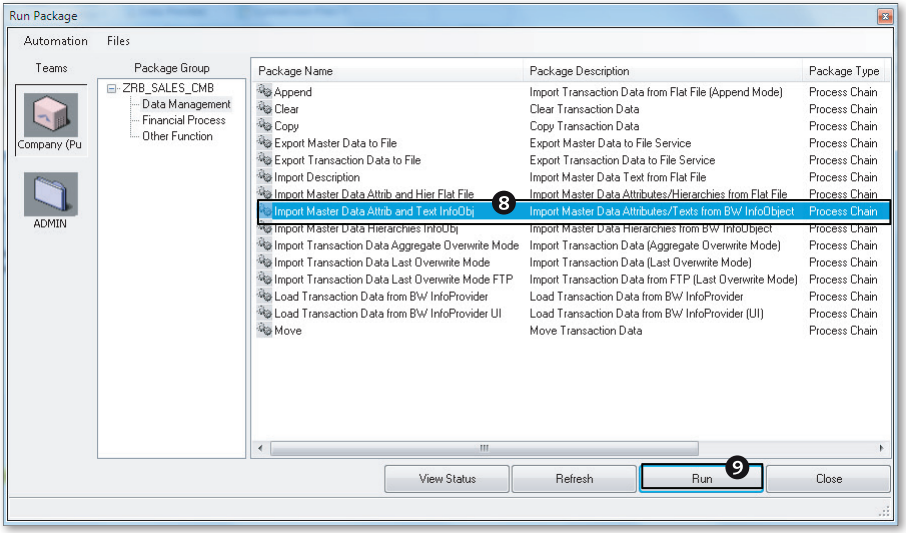


Figure 4.33 Loading Data from an InfoObject into a Dimension—Part C

10. In the Run Package dialog box, click on the Select button to select the source InfoObject for loading the data (Figure 4.34, ❿).
11. In our example, the ZACCOUNT InfoObject is used as the source of data for loading the RB_ACCOUNT dimension. Enter "ZACCOUNT" in the search box, and click on Next (Figure 4.35, ⓫ and ⓬). The system highlights the InfoObject. Select the ZACCOUNT InfoObject, and click on OK (Figure 4.35, ⓭ and ⓮). The InfoObject ZACCOUNT is transferred into the selection box (Figure 4.36, ⓯).

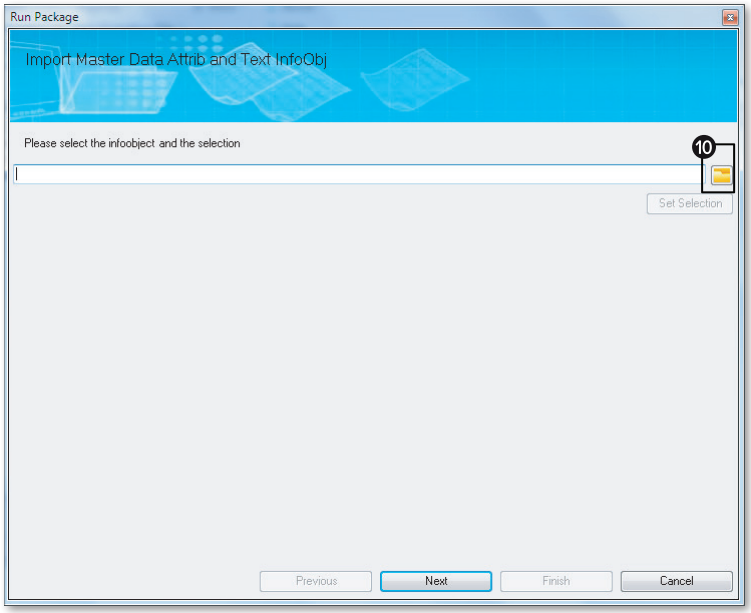


Figure 4.34 Loading Data from an InfoObject into a Dimension—Part D

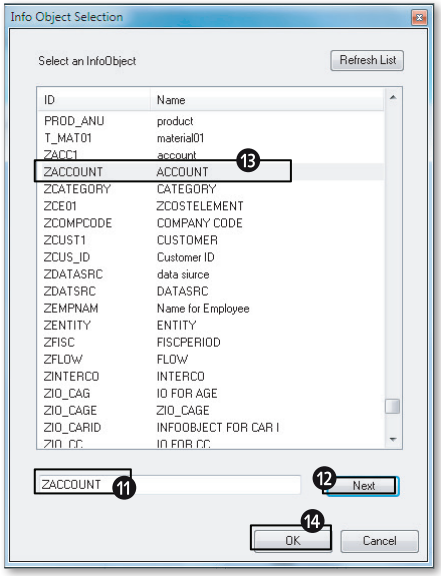


Figure 4.35 Loading Data from an InfoObject into a Dimension—Part E

12. Click on SET SELECTION to set the selection criteria for loading the data. This opens the SET SELECTION dialog box (Figure 4.36, ⑯).

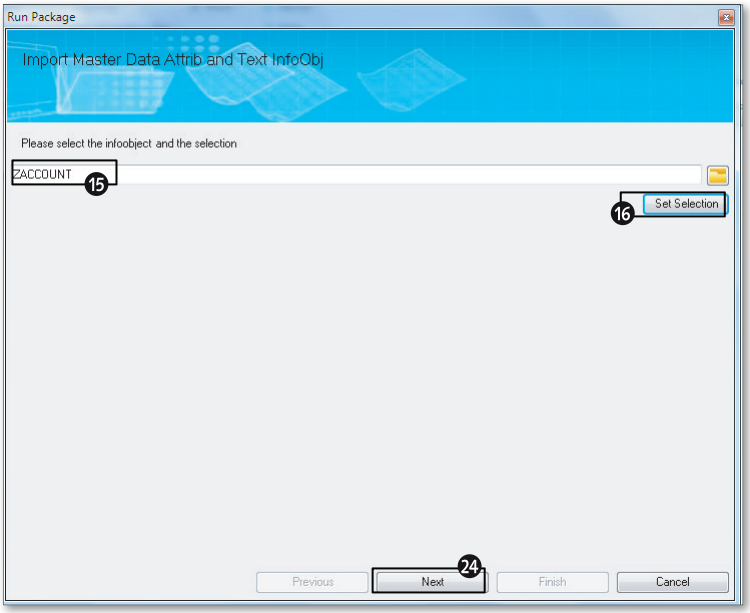


Figure 4.36 Loading Data from an InfoObject into a Dimension—Part F

13. The SET SELECTION dialog box includes four tabs: ATTRIBUTE, HIERARCHY, LANGUAGE, and ATTRIBUTE LIST. You specify the criteria to load the master data in the ATTRIBUTE tab.
    - In the HIERARCHY tab, specify the hierarchy name and the nodes for which the hierarchy text nodes are to be loaded. Hierarchy nodes in SAP NetWeaver BW are required to be loaded as regular dimension members before the hierarchy data is loaded. Even though we are not loading hierarchy data in this section, we are loading hierarchy text nodes as dimension members.
    - In the LANGUAGE tab, specify the language to use to pull text data. This is relevant when data in SAP NetWeaver BW is maintained in more than one language. The type of text data to load (SHORT, MEDIUM, or LONG) is also specified in this tab.
    - In the ATTRIBUTE LIST tab, identify the attributes you want to load from the InfoObject.

14. Because we want to load all of the master data in the ZACCOUNT InfoObject, do not specify any selections in the Attribute tab. Select the Set filters by Attributes OR Hierarchies radio button to ensure selections based on the values selected in the Attribute or Hierarchy tab are included (Figure 4.37, ⓯).
    - In the Hierarchy tab, make the selections to Import Text Nodes. The hierarchy created for the ZACCOUNT InfoObject is used as the source for loading the data (Figure 4.38, ⓲ and ⓳). Select the hierarchy name, parent node, and the levels to use for the selection (refer back to Figure 4.37, ⓴, ㉑, and ㉒).
    - Select EN as the language in the Language tab.
    - No specific selections are required in the Attribute tab.
15. After making the selections, click on OK (refer back Figure 4.37, ㉓). Now click on Next to go to the next step (refer back to Figure 4.36, ㉔).

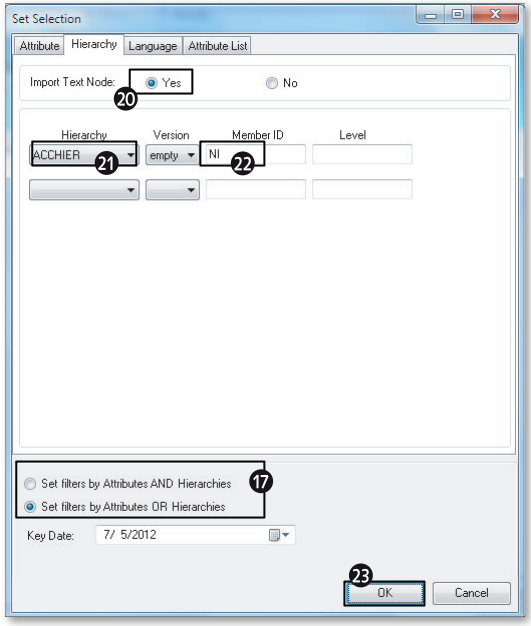


Figure 4.37 Loading Data from an InfoObject into a Dimension—Part G

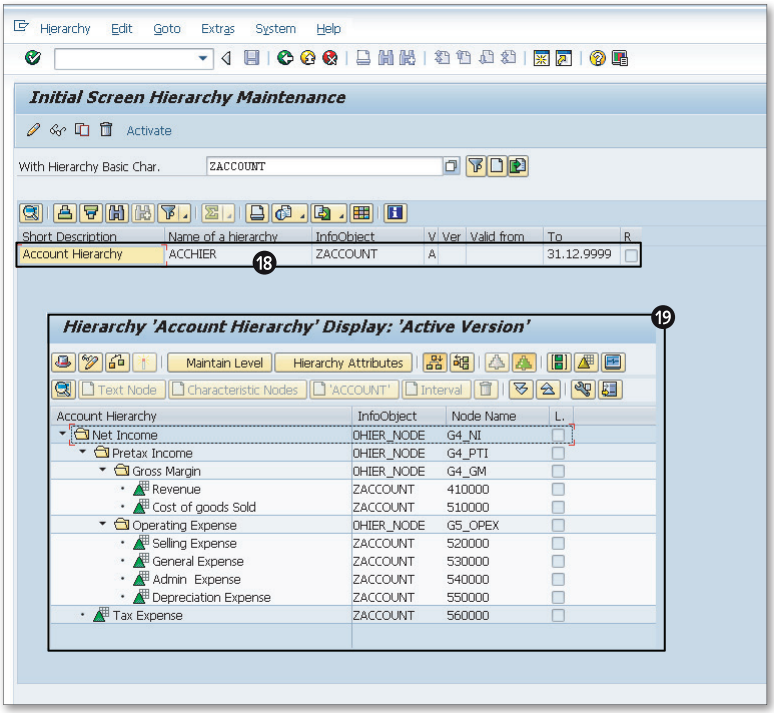


Figure 4.38 Loading Data from an InfoObject into a Dimension—Part H

16. In the next step, select the option for updating the data.

    - In OVERWRITE mode, if mapping is not set for a property in the transformation that is used to load the dimension, the property value is set to blank.
    - In UPDATE mode, the existing values of the property are maintained when mapping is not set in the transformation.

    We have selected the UPDATE mode (Figure 4.39, ㉕). Click on NEXT to proceed (Figure 4.39, ㉖).

17. Select the format for loading the data. In INTERNAL FORMAT, the data is transferred from the InfoObject to SAP BPC as it is stored in the database. In EXTERNAL FORMAT, the data is transferred from the InfoObject to SAP BPC in the display format. This is the format in which the users view the data. For example, an InfoObject may be defined to not display leading zeros even though the data is stored internally with the leading zeros. We have selected the EXTERNAL FORMAT (Figure 4.40, ㉗). Click on NEXT (Figure 4.40, ㉘).

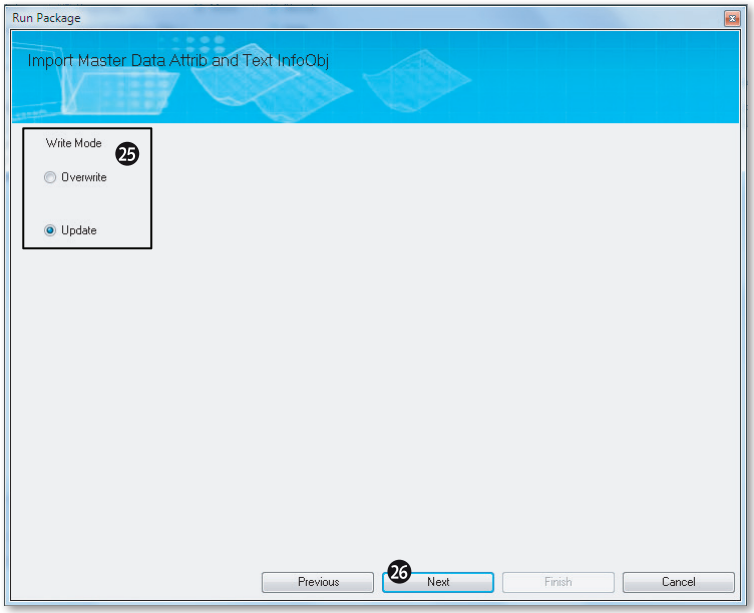


Figure 4.39 Loading Data from an InfoObject into a Dimension—Part I

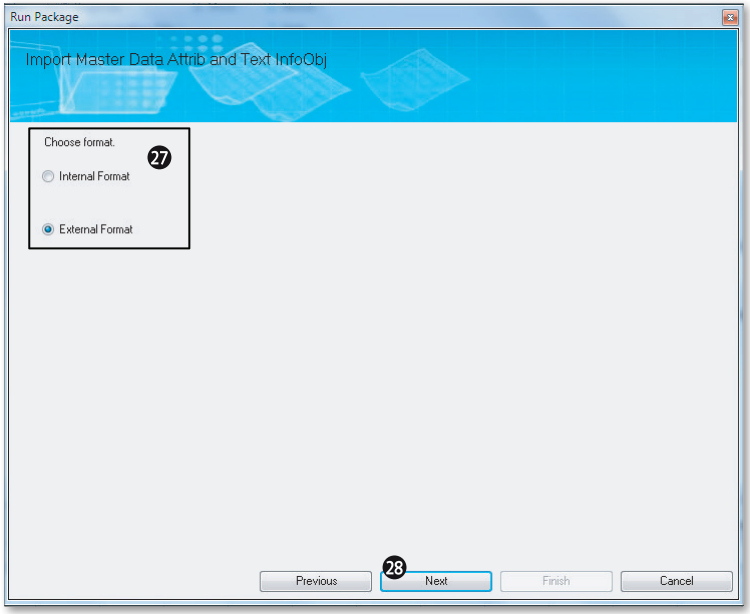
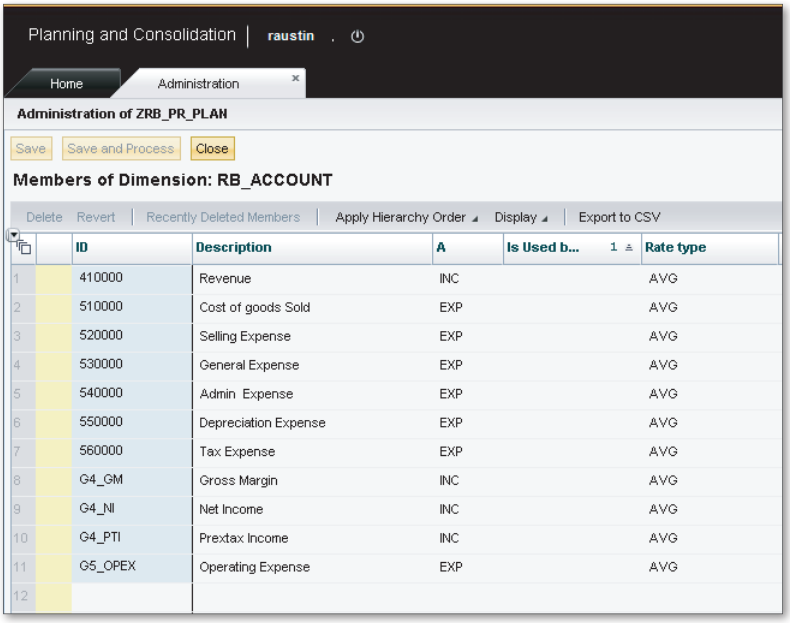


Figure 4.40 Loading Data from an InfoObject into a Dimension—Part J

18. You are prompted to select a transformation file to use to run the package. Select the transformation file we created in the earlier steps for this purpose (TR_MAS_ACCOUNT_BW.XLS). Click on NEXT.
19. Select the RB_ACCOUNT as the dimension name to load the data example. Click on NEXT.
20. Finally, select the RUN option to run the package immediately. This will load the master data from the ZACCOUNT InfoObject into the RB_ACCOUNT dimension. View the results of the data package and confirm the data loaded successfully into the RB_ACCOUNT dimension. The data in the RB_ACCOUNT dimension after loading the master data from the ZACCOUNT InfoObject is displayed in Figure 4.41.



	ID	Description	A	Is Used b...	Rate type
1	410000	Revenue	INC		AVG
2	510000	Cost of goods Sold	EXP		AVG
3	520000	Selling Expense	EXP		AVG
4	530000	General Expense	EXP		AVG
5	540000	Admin Expense	EXP		AVG
6	550000	Depreciation Expense	EXP		AVG
7	560000	Tax Expense	EXP		AVG
8	G4_GM	Gross Margin	INC		AVG
9	G4_NI	Net Income	INC		AVG
10	G4_PTI	Prextax Income	INC		AVG
11	G5_OPEX	Operating Expense	EXP		AVG
12					

Figure 4.41 Loading Data from an InfoObject into a Dimension—Part K

We have seen how to load master data from an InfoObject in BW into a dimension in SAP BPC. In the next section, we will see how to load hierarchy data from the ZACCOUNT InfoObject in BW into the RB_ACCOUNT dimension.

4.2.6 Loading Hierarchy Data from an InfoObject into an SAP BPC Dimension

Now we will see how to load hierarchy data from the ZACCOUNT InfoObject into a dimension in SAP BPC. Proceed as follows:

1. Log in to the BPC Web Client, and connect to the ZRB_PR_PLAN environment. From the Web Client, launch the EPM add-in for Excel, and connect to the ZRB_SALES_CMB model.
2. Begin by creating a conversion file. A conversion file is required when loading hierarchy data from a BW InfoObject into an SAP BPC dimension. The hierarchy created for the ZACCOUNT InfoObject is shown in Figure 4.38. The conversion file requires the technical name of the BW hierarchy from where the data is loaded in the EXTERNAL column and the name of the hierarchy where it is loaded in SAP BPC in the INTERNAL column. The conversion file mapping required for loading the hierarchy data is shown in Figure 4.42 (❶).
3. Save the conversion file as "CV_HIER_ACCOUNT_BW.XLS". Confirm that the hierarchy you specify in the EXTERNAL column exists in the BW InfoObject that you will use as the source of data. Also, make sure that the technical name of the hierarchy referenced in the internal column exists in SAP BPC. You can confirm it exists by reviewing the structure of the target dimension in the Web Client. If the hierarchy does not exist, create a hierarchy.

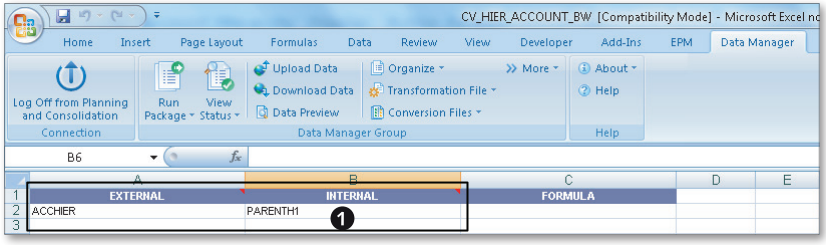


Figure 4.42 Loading Hierarchy Data from an InfoObject into a Dimension—Part A

4. Create a transformation file. The Data Manager package for loading the hierarchy data from SAP NetWeaver BW will use the *OPTIONS, *MAPPINGS, and *CONVERSION commands specified in the transformation file shown in Figure 4.43 (❷). The commands specified in the *MAPPING section are used to transform node, parent, and hierarchy names loaded from the SAP NetWeaver BW

InfoObject. The *CONVERSION section of the transformation file includes a reference to the conversion file that you created in the previous step to map the hierarchy in SAP NetWeaver BW to the hierarchy in SAP BPC. Save the transformation file as "TR_HIER_ACCOUNT_BW.XLS".

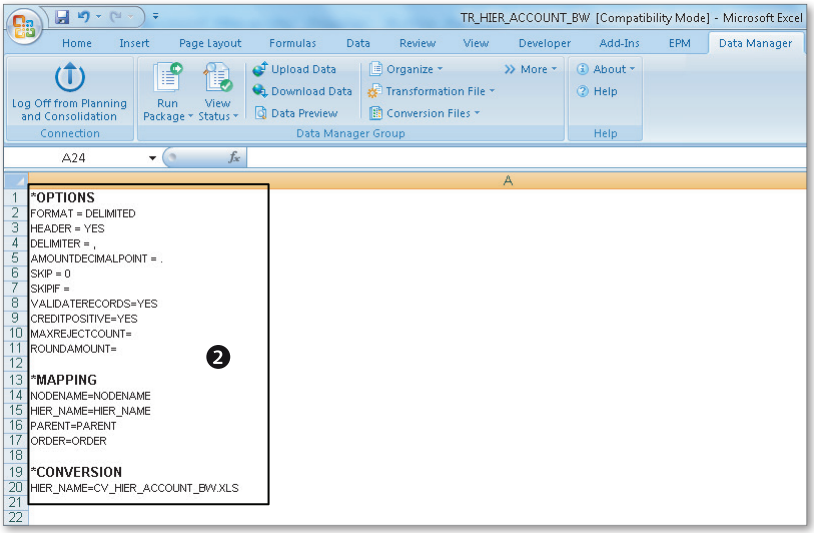


Figure 4.43 Loading Hierarchy Data from an InfoObject into a Dimension—Part B

5. You're now ready to execute the Data Manager package to load the hierarchy data for the account dimension from SAP NetWeaver BW. Click on the RUN PACKAGE menu option in the DATA MANAGER tab to execute a Data Manager package. In the dialog box that lists the Data Manager packages, select the DATA MANAGEMENT group, and execute the IMPORTMASTERDATA HIERARCHIES INFOOBJ Data Manager package (Figure 4.44). This Data Manager package uses the `/CPMB/IMPORT_IOBJ_HIER` process chain and is available to load hierarchy data from a BW InfoObject. Click on NEXT. This opens the RUN PACKAGE dialog box to select the InfoObject and hierarchy selections to use when loading data.
6. Similar to loading the master data from a BW InfoObject, select the InfoObject for loading the hierarchy data. Click on the dimension selection icon to open the dialog box to select a dimension (Figure 4.45, ❸). The dimension selected in the SELECT DIMENSION dialog box is transferred as the source InfoObject for the load (Figure 4.45, ❹). Click on SET SELECTION to select the selection for hierarchy data (Figure 4.45, ❺).

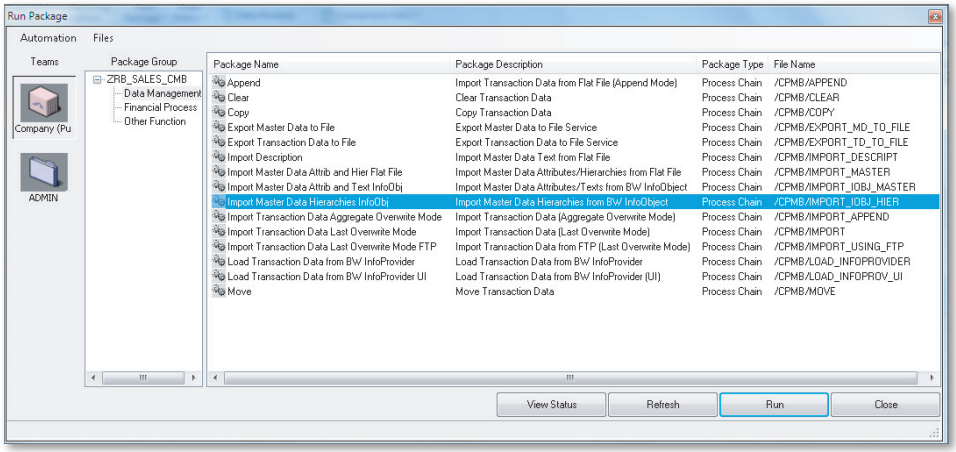


Figure 4.44 Loading Hierarchy Data from an InfoObject into a Dimension—Part C

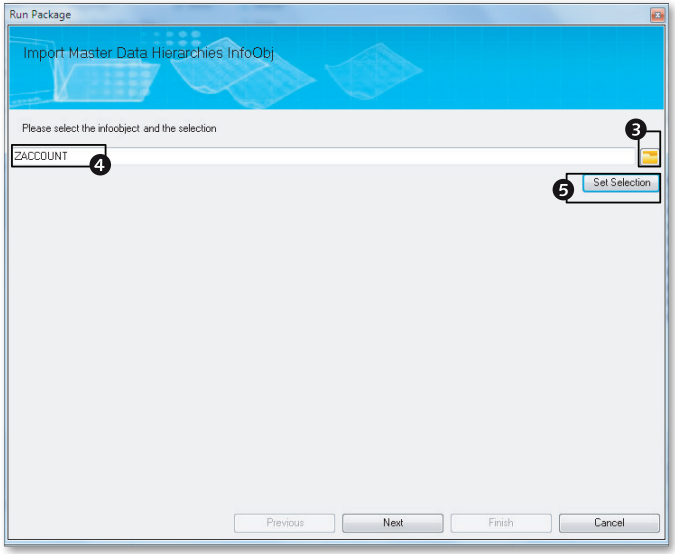


Figure 4.45 Loading Hierarchy Data from an InfoObject into a Dimension—Part D

7. In the SET SELECTION dialog box, select the technical name of the hierarchy nodes and the number of levels you want to load from the InfoObject. In our example, we will load the entire hierarchy from the top node (Figure 4.46, ❻). Click on OK to transfer the selections, and then click on NEXT to go to the next step. You will be prompted to select the transformation file and the dimension to which this data is loaded.

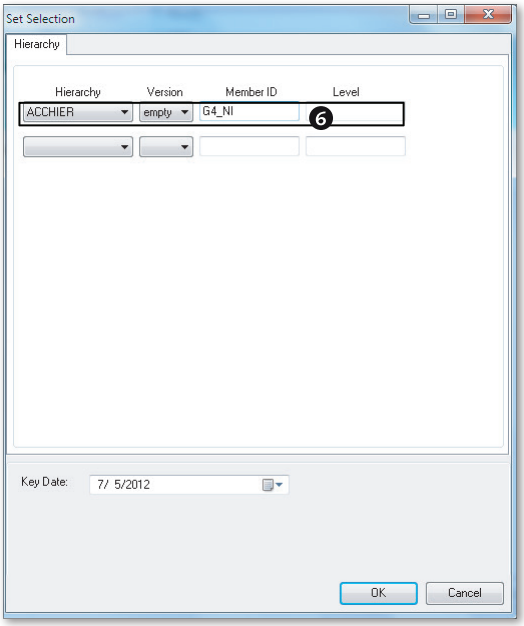


Figure 4.46 Loading Hierarchy Data from an InfoObject into a Dimension—Part E

8. In the final step, select the RUN option to run the package immediately. This will load the hierarchy data into the RB_ACCOUNT dimension. The hierarchy loaded into the RB_ACCOUNT dimension in shown in Figure 4.47.

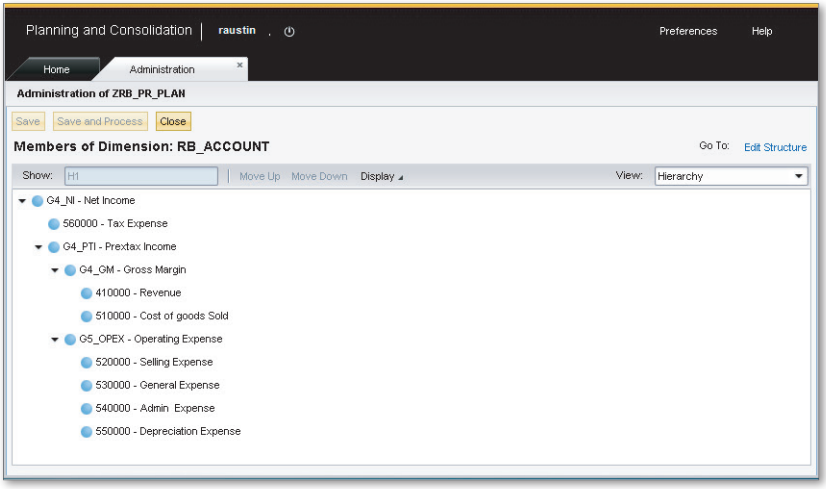


Figure 4.47 Loading Hierarchy Data from an InfoObject into a Dimension—Part F

We have seen how to load hierarchy data from an InfoObject in SAP NetWeaver BW into a dimension in SAP BPC. In the next section, we'll load the actual sales and cost data from the ZSLSACT InfoCube in SAP NetWeaver BW into the planning model in SAP BPC.

4.2.7 Loading Data from an InfoCube into an SAP BPC Model

Rich Bloom stores its sales and cost data in the ZSLSACT InfoCube in SAP NetWeaver BW. This data needs to be loaded into the ZRB_SALES_CMB planning model in SAP BPC. Follow these steps:

1. Log in to the SAP BPC Web Client, and connect to the ZRB_PR_PLAN environment. From the Web Client, launch the EPM add-in for Excel, and connect to the ZRB_SALES_CMB model.
2. A transformation file is required that defines the structure of the data coming from the InfoCube and maps the InfoObjects in the InfoCube to the dimensions in the model. The transformation will reflect the structure of the data in the ZSLSACT InfoCube. In the *MAPPING section, map the dimension names in the model to the corresponding technical names of the InfoObjects in the ZSLSACT InfoCube. All of the dimensions in an SAP BPC model must be mapped to an InfoObject or field from an external source when loading the data. When the external data source does not supply a value for a dimension, you can use the keyword `*STR(<value>)` to specify a value for the dimension. The `<value>` represents a member ID for the dimension. The data in the Sales InfoCube does not include the data for the Category dimension. Map the Category dimension to the Actual member ID value in the *MAPPING section. Dimension members are case-sensitive. Similarly, because the amounts are all in local currency, set the Reporting Currency dimension to LC (local currency).

   Before we create a transformation file, let's create a conversion file for the Time dimension.
3. The data for calendar month is in the YYYYMM format in SAP NetWeaver BW. You have to convert this into an equivalent SAP BPC format for the Time dimension, so you must define a mapping file for this conversion. The conversion file contains three columns: EXTERNAL, INTERNAL, and FORMULA. In the EXTERNAL column, specify the value coming from the external source. In the INTERNAL column, specify the value as it is stored in SAP BPC. The FORMULA can

be used for any calculations on amount values. Create a conversion file as shown in Figure 4.48. Name the conversion file "CV_MAS_TIME_BW.xls".

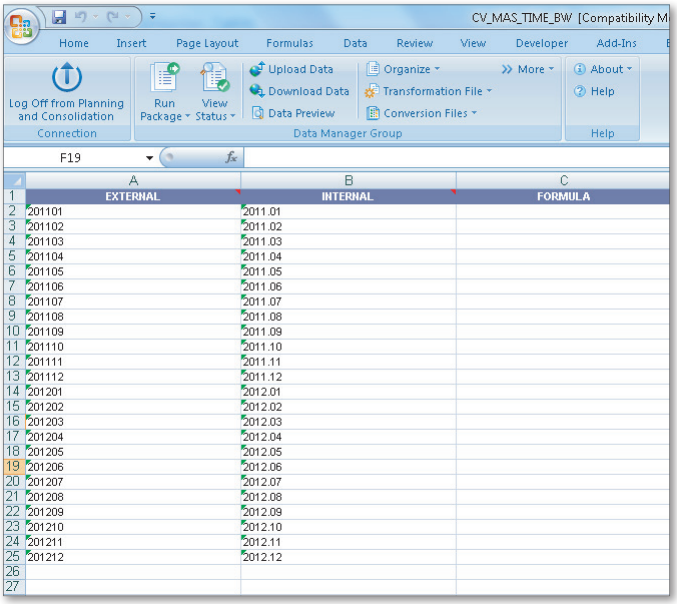


Figure 4.48 Loading Data from an InfoCube into a Model —Part A

4. Create a transformation file with the definitions shown in Figure 4.49. In the *MAPPING section, the dimension names are mapped to the corresponding InfoObject in the ZSLSACT InfoCube. The assignments for the Category and Reporting Currency dimensions are set to a constant value. In the *CONVERSION section, map the Time dimension to the conversion file created in the previous step. Save the transformation file as "TR_SALES_ACTUAL_BW.xls". Add the command to reference the conversion file in the *CONVERSION section for the Time dimension in the transformation file.

5. You're now ready to execute the package to load the data from the ZSLSACT InfoCube into the SAP BPC model. Select the Load Transaction Data from BW InfoProvider UI Data Manager package under the Data Management group, and click on Run (Figure 4.50). This Data Manager package uses the `/CPMB/LOAD_INFOPROV_UI` process chain. This will open the Run Package dialog box.

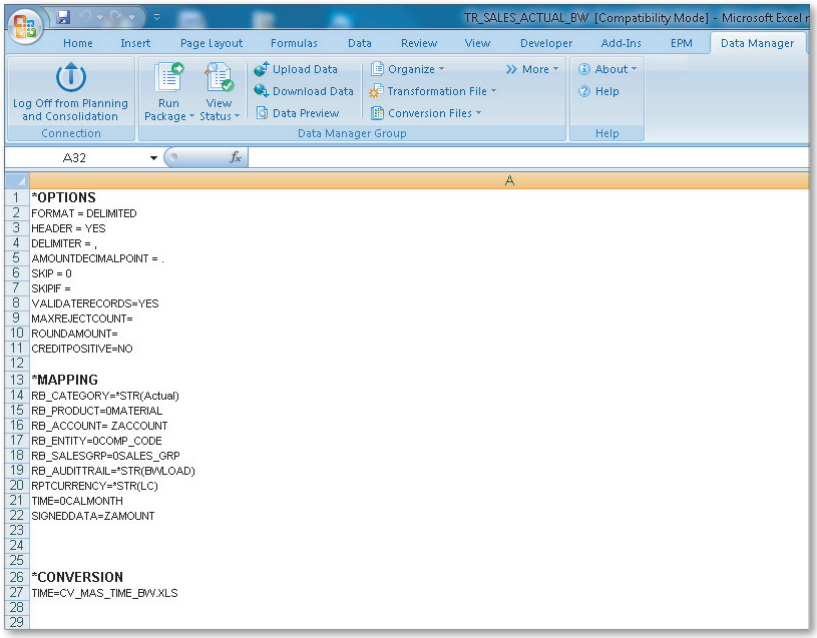


Figure 4.49 Loading Data from an InfoCube into a Model—Part B

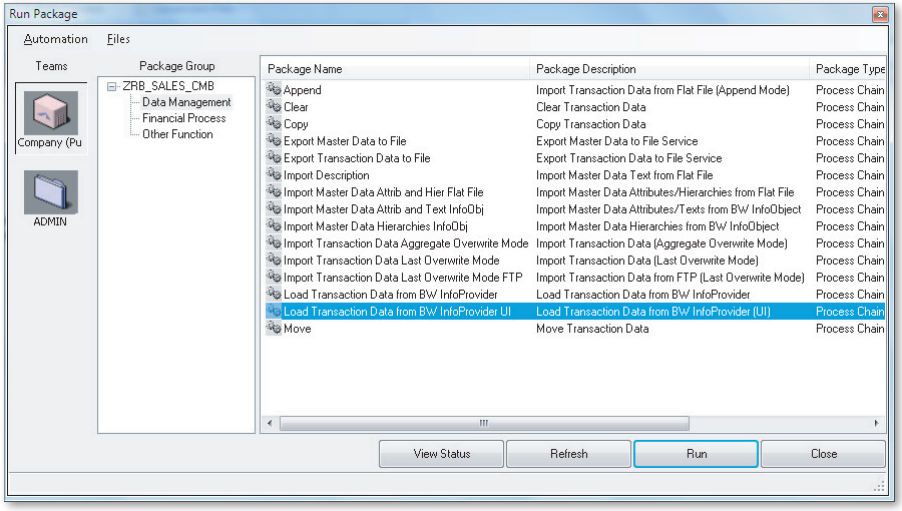


Figure 4.50 Loading Data from an InfoCube into a Model—Part C

6. In the RUN PACKAGE dialog box, enter "ZSLSACT" as the InfoProvider from which to extract data (Figure 4.51, ❶). Alternatively, you can select the InfoProvider by clicking on the SELECT icon (Figure 4.51, ❷) and then selecting the ZSLSACT InfoProvider in the following dialog box. We want to extract the data only for the calendar year 2011, so we need to specify a criterion here to restrict the data loaded only for calendar year 2011. If you want to restrict data that is loaded from the ZSLSACT InfoProvider to the model, click on the SET SELECTION button (Figure 4.51, ❸), and enter selections. Enter the selections as shown in Figure 4.52.

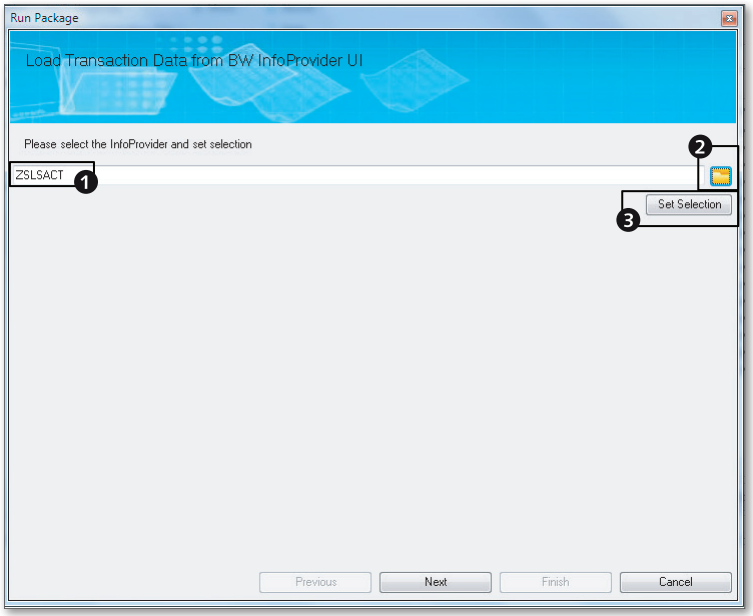


Figure 4.51 Loading Data from an InfoCube into a Model—Part D

7. Specify the transformation file you created in the previous steps: "TR_SALES_ACTUAL_BW" (Figure 4.53).

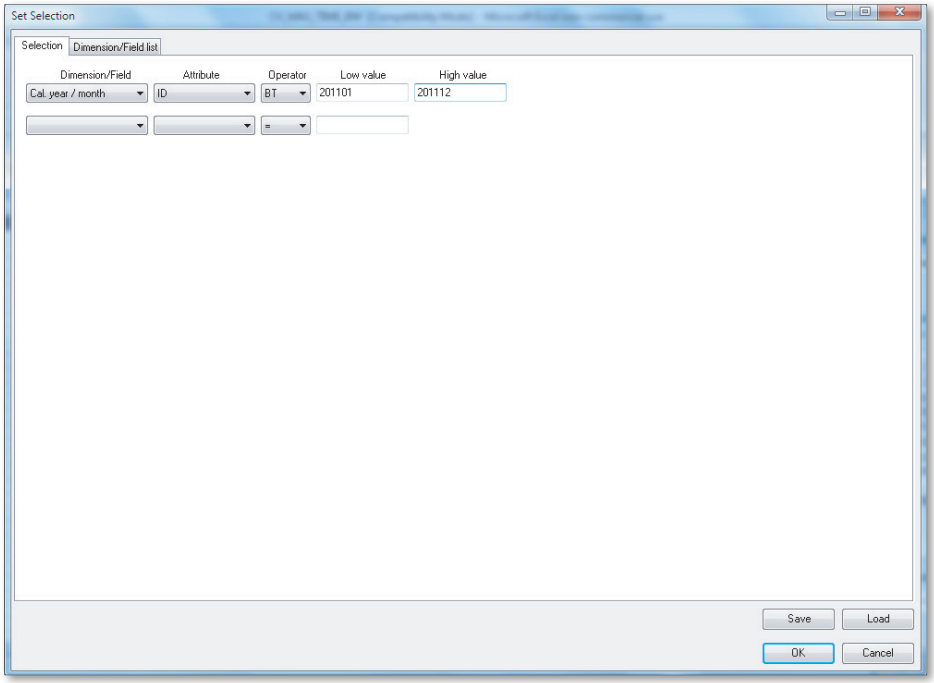


Figure 4.52 Loading Data from an InfoCube into a Model—Part E

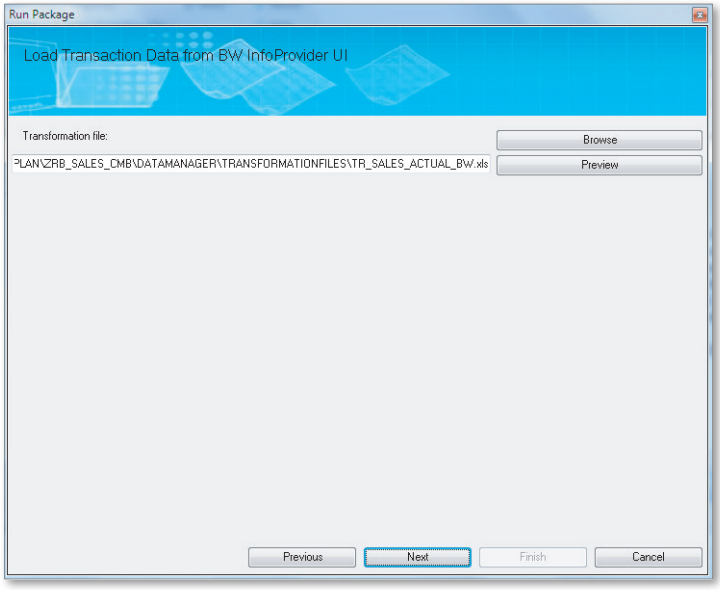


Figure 4.53 Loading Data from an InfoCube into a Model—Part F

8. Select the DATA-TRANSFER MODE option for loading the data from the InfoCube. Two options are available:
   - AGGREGATE OVERWRITE: The source data is aggregated by all of the dimensions before the data is loaded into the SAP BPC model.
   - APPEND: The source data is appended into the target SAP BPC model.

   For our example, select AGGREGATE OVERWRITE (Figure 4.54). Click on NEXT to proceed to the next step.

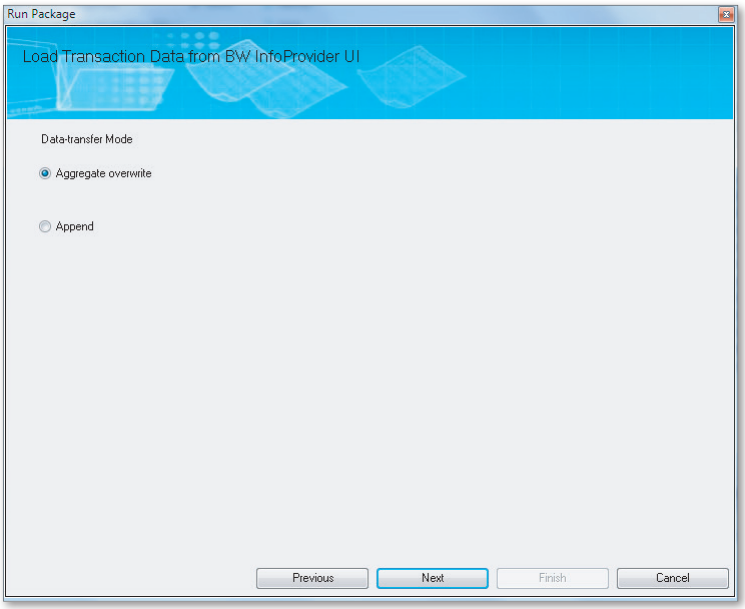


Figure 4.54 Loading Data from an InfoCube into a Model—Part G

9. Select the method of importing data from the database. Two options are available for this:
   - MERGE DATA VALUES: This option merges data that is being loaded to the model with data that already exists in the model. When an InfoCube sends data for a combination of dimension values, and the corresponding data exists in the model, the record coming from the InfoCube updates the existing data in the model. If the record does not exist in the model, the record is added to the model.
   - REPLACE & CLEAR DATAVALUES: The system first clears the records from the model based on the values of the Category, Entity, Time, and AuditTrail

dimensions in the incoming records and then loads the data in the source to the model.

For our example, select REPLACE & CLEAR DATAVALUES (Figure 4.55). Click on NEXT to proceed to the next step.

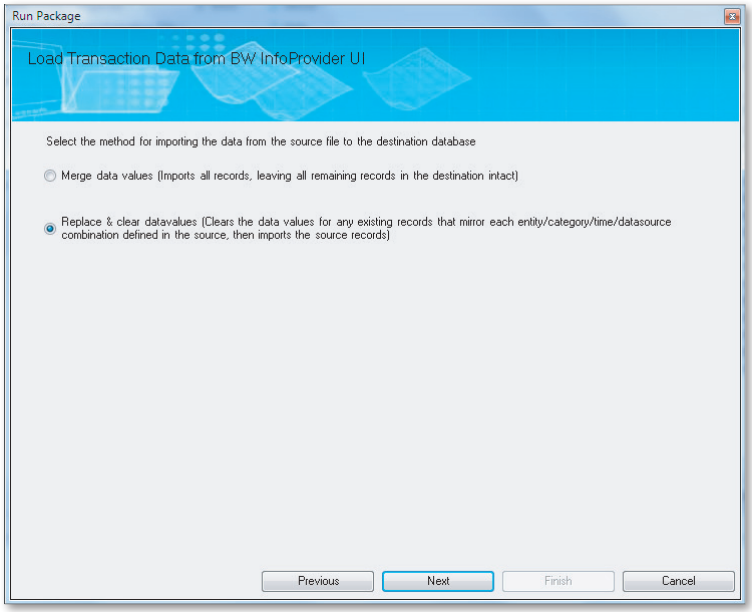


Figure 4.55 Loading Data from an InfoCube into a Model—Part H

10. The option to run default logic can be selected when this Data Manager package is run. We have not included any default logic for our model, so choosing YES or NO should not matter (Figure 4.56). Click on NEXT to proceed to the next step.

11. The work status lets you control updates to the data in a model. We'll discuss work status in detail in Chapter 7. For this example, select YES, CHECK FOR WORK STATUS SETTINGS BEFORE IMPORTING so that users can't load data if the work status setting does not allow them to do so (Figure 4.57). Click on NEXT to proceed to the next step.

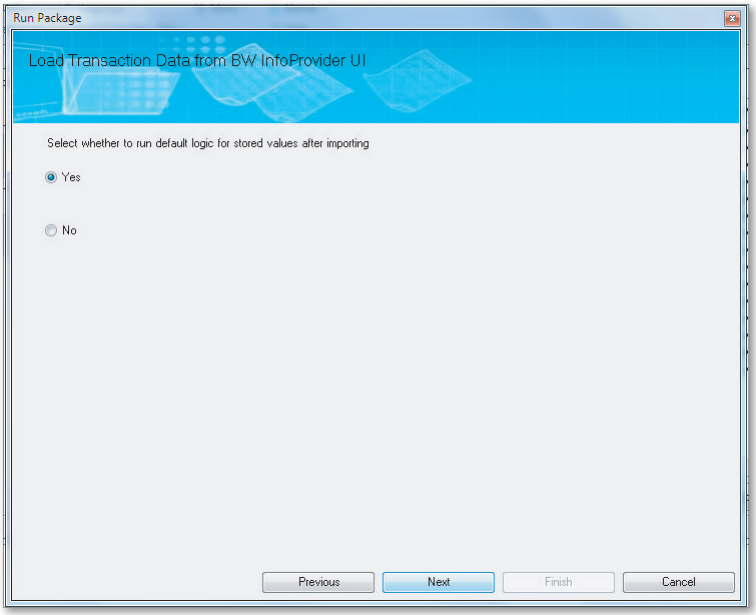


Figure 4.56 Loading Data from an InfoCube into a Model—Part I

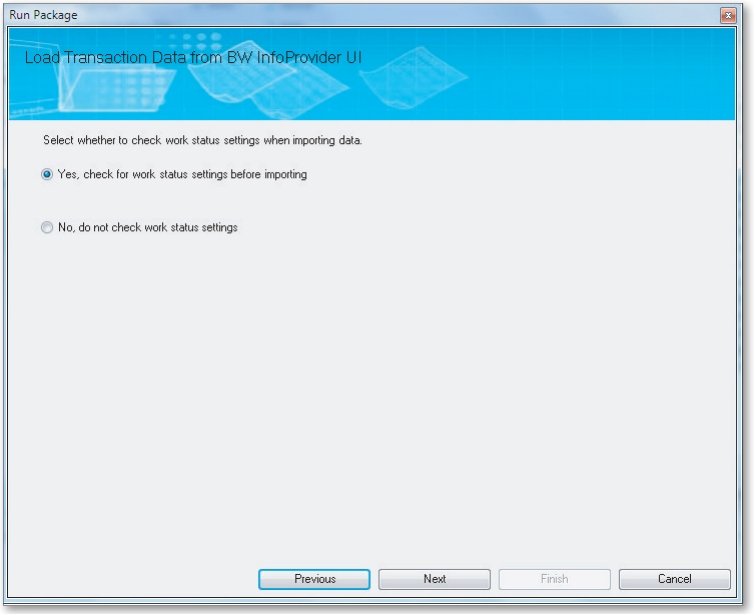


Figure 4.57 Loading Data from an InfoCube into a Model—Part J

12. Select Run to immediately execute the package, and then click on Finish.

13. View the package status and display the details of the log. Figure 4.58 displays the details of the package executed, along with the records processed.

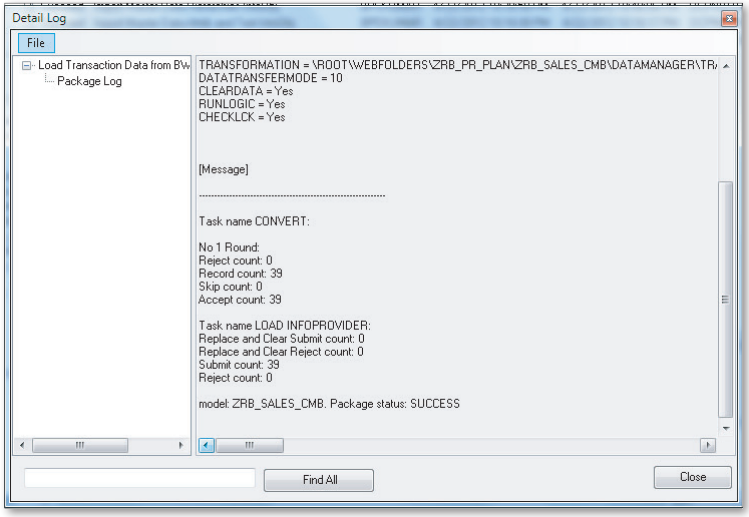


Figure 4.58 Loading Data from an InfoCube into a Model—Part K

In this example, you've seen how to load transaction data from an InfoProvider into a model in SAP BPC. We loaded the actual sales data for 2011 from the Actual Sales InfoCube in SAP NetWeaver BW into a planning model in SAP BPC. In the next section, we'll create baseline plan data for 2012 by copying the sales and cost data for 2011 as the plan data for 2012.

4.3 Copying Data Inside an SAP BPC Model

To copy data inside an SAP BPC model, follow these steps:

1. Log in to the SAP BPC Web Client, and connect to the ZRB_PR_PLAN environment. From the Web Client, launch the EPM add-in for Excel, and connect to the ZRB_SALES_CMB model.

2. You're now ready to copy the actual sales and cost data for 2011 as the plan data for 2012. Click on Run Package in the Data Manager tab. In the dialog box that lists the Data Manager packages, select the Data Management group, select

the Copy Data Manager package, and click on Run (Figure 4.59). The Copy Data Manager package uses the /CPMB/COPY process chain. This opens the Data Manager—Run Package dialog box.

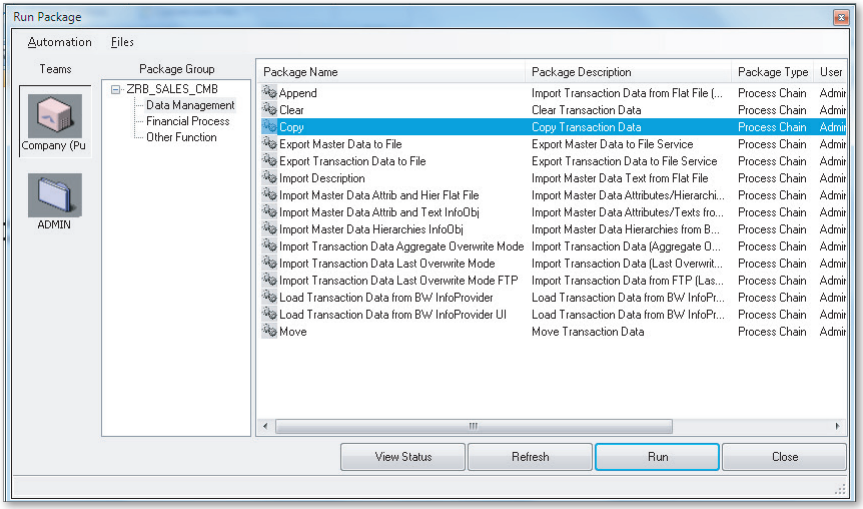


Figure 4.59 Copying Data Inside a Model—Part A

3. In the subsequent screen, select the options to replace and clear data values, run the default logic, and check the work status before data is imported.
4. You are now presented with the options to select the source value selections for every dimension in the model and the corresponding target values to which the data is to be copied. Our objective is to copy the data from the Actual category for 2011 to the Plan category for 2012. Under Source dimensions, click on the Add button for RB_CATEGORY to select a member value for the Category dimension (Figure 4.60, ❶). This opens the EPM - Member Selector dialog box to select values.
5. Select the Actual member ID from the dimension members, and click on OK (Figure 4.61, ❷, ❸, and ❹). The selected value gets copied to the category dimension (refer back to Figure 4.60, ❺).
6. Under Source dimensions, select the Add button for the TIME dimension to display a dialog box to select time periods. Select all of the 2011 time periods, and click on OK to return the values as the source data selected for the TIME dimension (refer back to Figure 4.60, ❻).

7. Under DESTINATION dimensions, select the PLAN member ID for the RB_CATEGORY dimension (refer back to Figure 4.60, ❼). Also under DESTINATION dimensions, select all of the 2012 time periods for the TIME dimension (refer back to Figure 4.60, ❽).

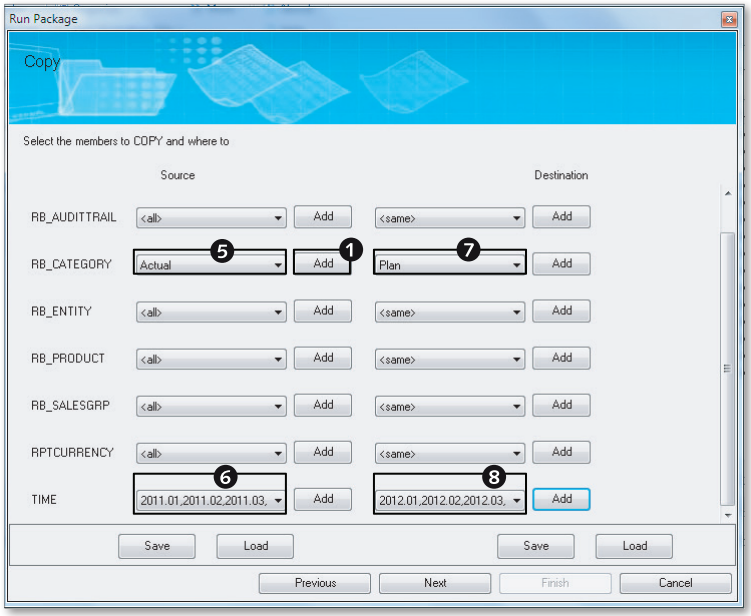


Figure 4.60 Copying Data Inside a Model—Part B

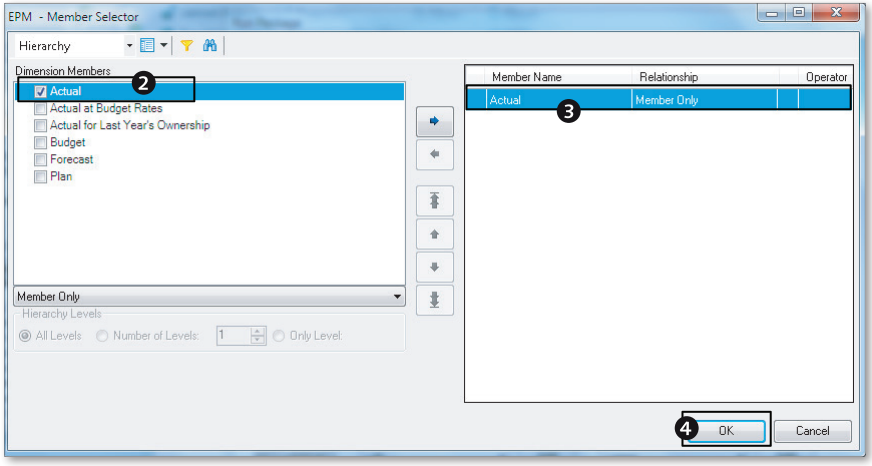


Figure 4.61 Copying Data Inside a Model—Part C

8. In the subsequent screens, select the options to replace and clear data values, run the default logic, and check the work status before data is imported.
9. Select RUN to immediately execute the package, and then click on FINISH.

Display the status of the package to confirm the data loaded successfully. The detailed log of the package is displayed in Figure 4.62.

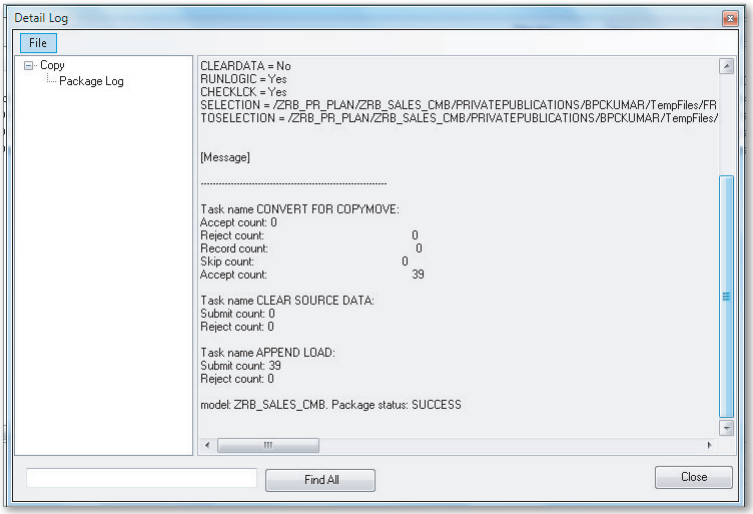


Figure 4.62 Copying Data Inside a Model—Part D

You should now understand how to copy data within a model. At this point, we have created the baseline plan data for Rich Bloom for 2012. This plan data can be revised as necessary to reflect market demand.

4.4 Summary

In this chapter, you learned how to load data from a flat file into an InfoCube in SAP NetWeaver BW. You also learned how to load data from a flat file to a dimension, how to load master and hierarchy data from an InfoObject into a dimension, and how to load data from an InfoProvider in SAP NetWeaver BW to a model in SAP BPC. In this process, you learned how transformation and conversion files are used when loading dimension and model data into SAP BPC. To satisfy the requirements of our model company Rich Bloom, we performed the following actions to bring data into our planning model:

- Loaded the entity master data from a flat file into the Entity dimension
- Loaded the master data and hierarchy data from an InfoObject into the Account dimension
- Loaded the actual sales and cost data for 2011 from an InfoCube into the planning model in SAP BPC
- Created baseline plan data for 2012 by copying the actual sales and cost data for 2011 to plan data for 2012

In the next chapter, you'll learn how to report the data available in SAP BPC. You'll also learn how to enter plan data using input forms.

SAP Business Planning and Consolidation provides all business users with an intuitive, user-friendly, uniform, and familiar Microsoft Excel and web environment for developing a user interface that can be used not only to report data but also to enter and save data into an SAP BPC model. In this chapter, we will look at how to use the Excel and the web interfaces for developing reports and cover the advanced features for developing a report.

5 Reporting, Planning, and Analysis in SAP BPC

In this chapter, we'll discuss how to report, plan, and analyze data using the SAP EPM add-in tool for Microsoft Excel and the SAP BPC web interfaces. In the previous chapter, you learned how to load transaction data into an SAP BPC model; in this chapter, we'll describe the steps for setting up input forms to enable users to manually enter or modify data in a model.

Section 5.1 introduces the SAP EPM add-in tool for Microsoft Excel and the SAP BPC Web Client for creating reports in SAP BPC. We'll discuss the components of the Report Editor, which is available in the EPM add-in tool. We'll highlight the features available in the tool to format a report and discuss, with examples, the steps to develop dynamic and static reports. We'll explain how to build a custom report using EPM functions, which are the standard functions used in reporting when using the EPM add-in tool. Subsequently, we will discuss migration of EvDRE reports created in SAP BPC 7.x version to the SAP BPC 10.0 version.

Section 5.2 discusses input forms and explains how to manually enter or modify data in an SAP BPC model. When you set up an input form, you enable users to enter or modify data in a model; we'll explore the options available for this setup.

Section 5.3 discusses the drill-through feature that allows users to drill down from SAP BPC to an external system via URL. The SAP BusinessObjects Dashboard tool (formerly Xcelsius) that is used to create dashboards is discussed in this section as well.

The chapter ends with a summary of the topics covered in this chapter and concludes that SAP BPC provides an intuitive and robust interface for entering plan data and for reporting and analyzing data.

5.1 Reporting and Analysis in SAP BPC

SAP BPC provides an easy-to-use and powerful interface for creating reports and maintaining data. The reporting options for SAP BPC can be classified under two broad categories:

- SAP EPM add-in for Microsoft Office
- SAP BPC Web Client

The SAP EPM add-in interface for Excel is the primary tool used to report data in SAP BPC. The data in SAP can also be integrated with Microsoft Word and PowerPoint documents. Meanwhile, the SAP BPC Web Client in SAP NetWeaver Business Warehouse (BW) provides the interface to design and run reports on the web.

The following are some of the key features of the EPM add-in reporting tool within SAP BPC:

- **Offers Excel tools with SAP data storage**
  When you use the SAP EPM add-in interface for Excel, you can use the functions and features available in Excel and, at the same time, store the data in an SAP BPC database. This enables users to use both Excel and a robust and secure database for storing data, which is necessary for high-volume enterprise-wide applications.
- **Provides dynamic templates**
  The SAP EPM add-in for Excel offers standard templates, also called *dynamic templates*, which support different types of analysis commonly requested by businesses. Standard templates can be used to develop different types of reports, including monthly comparison reports, yearly comparison reports, trend reports, and variance reports. The ability to use these templates with very little development effort reduces TCO.
- **Allows you to customize standard templates**
  Standard templates are easy to customize to meet a particular business requirement. This easy-to-customize feature makes standard templates more appealing.

- **Offers flexible ways to display data**
  The reporting interface is designed to provide maximum flexibility to display data. There is no code involved in creating a report. Using selection criteria, the report displays the data that is required for analysis. The system-supplied Measures dimension lets you view the data in different ways, either by period, quarter to date (QTD), or year to date (YTD).
- **Provides EPM functions**
  SAP BPC for Excel provides *EPM functions*, which is a table of functions used for reporting. This aids in creating and delivering sophisticated reports for efficient data analysis.
- **Allows offline analysis**
  The data displayed in SAP BPC reports can also be used for offline analysis. Specific features allow users to take the data from a report offline, modify it, and retract it back into SAP BPC.
- **Enables data distribution**
  The data in a model can be distributed to other users who may or may not have access to SAP BPC. This enables distributing data to users who may need access to it. These users can make changes to the data and send it back to SAP BPC based on their level of access.
- **Enables book publishing**
  This is a new feature introduced in the EPM 10.0 version for SAP NetWeaver. The ability to publish books has existed in the Microsoft version of SAP BPC in the earlier versions. A book template can be designed to include one or more reports with dimension member selections. The book template can be run interactively or scheduled to run at specified times. The output of the reports can either be sent to a user desktop or stored inside the SAP BPC system for users to view.
- **Facilitates data maintenance**
  In addition to reporting, SAP BPC can be used for entering and modifying data. We'll discuss this in detail when we talk about input forms later in this chapter.
- **Provides features unique to planning and consolidation**
  SAP BPC provides features that are used specifically in planning and consolidation models. The ability to use spread, trend, and weight data makes it easy to allocate data and create projections for the future.

- **Offers standardized reports**
  The reporting functionality includes out-of-the-box system reports that can be displayed and executed on the web (for example, journal reports, audit reports, and so on). They are tools to analyze metadata and to monitor changes to the objects in the SAP BPC system.

5.1.1 SAP EPM Add-In Interface for Excel: Connecting to a Model

The SAP EPM add-in for Excel interface is primarily used to create reports to analyze data and to create input forms to enter and modify data in SAP BPC. You can install the EPM add-in client to your desktop by selecting the Download Center menu option available in the Start Page view under the Home Workspace in the Web Client. You can launch the interface directly from the SAP BPC Web Client by clicking the EPM Office Add-In Excel link under the Launch area. The system will automatically authenticate the user to the EPM interface, establish a connection to the environment, and present a dialog box with the list of models available in the environment. The user can select a model to use for creating a report.

The following is the process to connect manually to an SAP BPC model from the EPM add-in Excel interface.

1. Launch the EPM add-in directly from the desktop by opening Microsoft Excel and then clicking the EPM tab (Figure 5.1, ❶).
2. Click on the Log On button to connect (Figure 5.1, ❷).
3. This opens the EPM – Logon dialog box. Click on the ellipsis icon to select a connection (Figure 5.1, ❸). A new dialog box, EPM – Connection Manager, opens to maintain connections or select an existing connection (Figure 5.2).

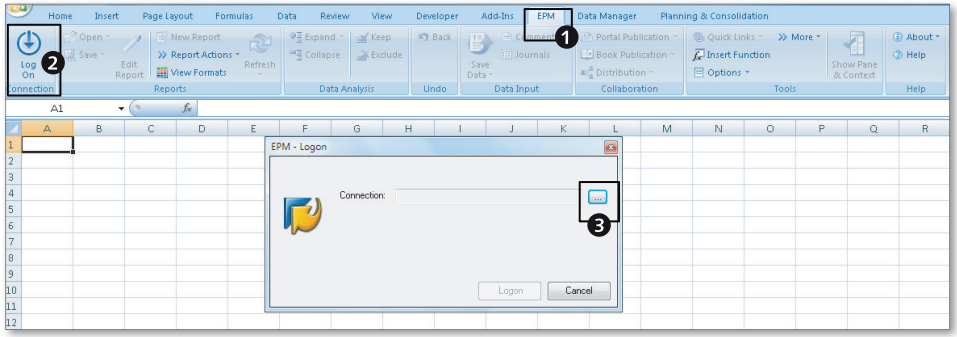


Figure 5.1 Creating a Manual Connection—Part A

4. To create a new connection, click on the CREATE button (Figure 5.2, ❹). This opens a new dialog box, CREATE CONNECTION, for the user to specify the connection parameters.

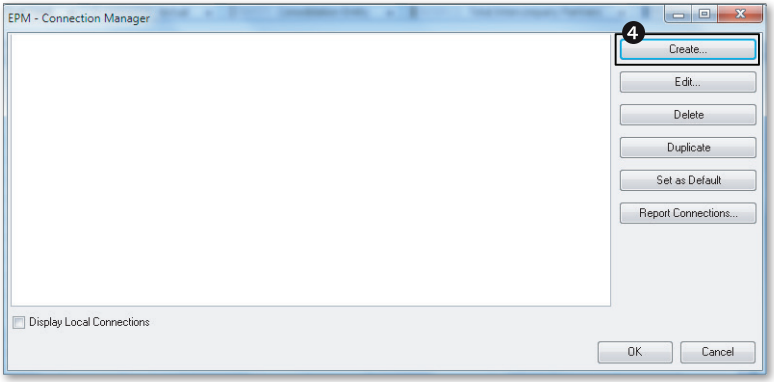


Figure 5.2 Creating a Manual Connection—Part B

5. Select the PLANNING AND CONSOLIDATION option under CONNECTION TYPE (Figure 5.3, ❺). In the PLANNING AND CONSOLIDATION CONNECTION section, for the SERVER URL value, enter the URL for your BPC server (Figure 5.3, ❻). For this example, we are entering "http://rbus.richbloom.com:8000/sap/bpc/".
6. For TYPE, select VERSION FOR SAP NETWEAVER from the dropdown list when you are connecting to a NetWeaver BW system (Figure 5.3, ❼). Click on the CONNECT button (Figure 5.3, ❽).

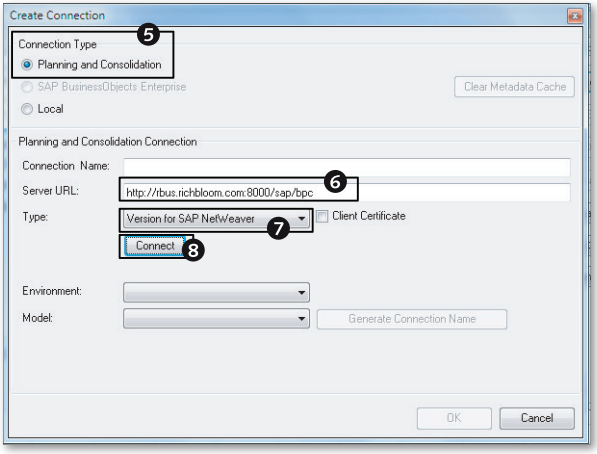


Figure 5.3 Creating a Manual Connection—Part C

7. The system will open the EPM – Logon dialog box for the user to enter the user ID and password (Figure 5.4). For this example, we are entering "raustin" as the User Name.

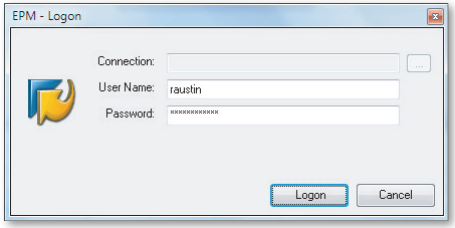


Figure 5.4 Creating a Manual Connection—Part D

8. The system authenticates the user and in the next screen provides the option to select an environment from the Environments dropdown list (Figure 5.5, ❾). Only the environments that the user has access to will be displayed in this list. After selecting the environment, a list of models available in the environment is displayed, and a model is selected (Figure 5.5, ❿). You can let the system generate a connection name by clicking on Generate Connection Name. The connection name generated can be overwritten with a name of your choice.

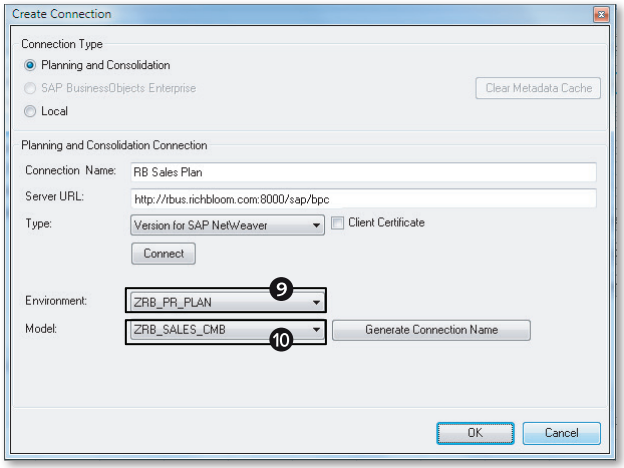


Figure 5.5 Creating a Manual Connection—Part E

You have now created a connection to access a model inside an environment in SAP BPC (Figure 5.6).

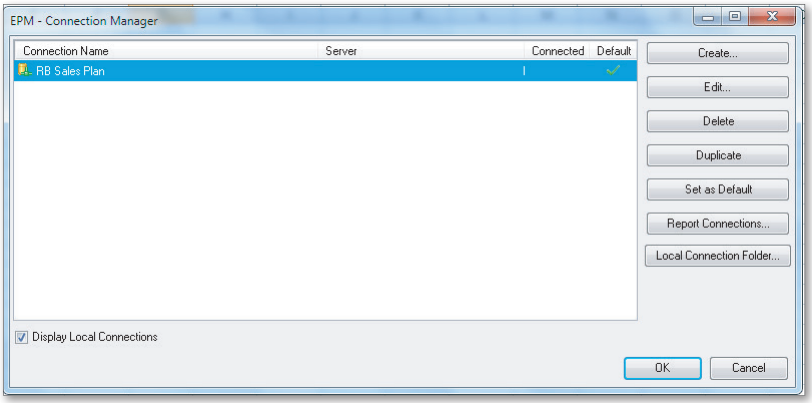


Figure 5.6 Creating a Manual Connection—Part F

Before we create a new report, you need to understand some features available in the tool for reporting purposes. Let's first discuss the Measures dimension.

Measures Dimension

The Measures dimension is important in reports because it impacts the way the data is displayed. The Measures dimension works in conjunction with the Time dimension and is available automatically in every SAP BPC model.

You use the Measures dimension to report quantitative information for a particular period of time, using a formula that distinguishes between balance sheet accounts and profit and loss accounts and reports information accordingly. Three types of measures are supplied with the Measures dimension to be used in reports:

- **Periodic**
  This measure displays the data for the period for which the data is selected. For example, if you select the Feb 2012 period in your report, selecting the periodic measure displays the data for that period. For profit and loss accounts, the total value for the period is reported; for balance sheet accounts, the ending balance for the period is reported.
- **Quarter to date (QTD)**
  This measure displays the quarter-to-date data up to the period selected. The system determines the quarter for the period displayed in the report and cumulates the data from the beginning of that quarter to the period displayed in the report. For profit and loss accounts, if you select Feb 2012 and select the QTD

measure, the data from the beginning of that quarter—Jan 2012—through Feb 2012 is reported. For balance sheet accounts, the ending balance for Feb 2012 is reported.

- **Year to date (YTD)**
  This measure displays the year-to-date data up to the period selected. For profit and loss accounts, if you select May 2012 and select the YTD measure, all of the data from the beginning of the year—Jan 2012—through May 2012 is reported. For balance sheet accounts, the ending balance for May 2012 is reported.

The data for a model can be stored as either periodic or YTD, and the storage type is dictated by the parameter associated with the model. The default storage type for a model is periodic.

The formula for the Measures dimension is maintained at the model level. You can view the formula in the SAP NetWeaver BW system using Transaction UJA_MAINTAIN_MEASURE_FORMULA. Enter the environment ID, model, and user ID.

Listing 5.1 shows the YTD measure for a model whose data storage type is set to "periodic." The YTD measure formula, based on the type of account (profit and loss or balance sheet), either sums up the values from the beginning of the year or provides the balance as of the end of a period.

```
MEMBER [MEASURES].[YTD] AS 'IIF([%P_ACCT%].CURRENTMEMBER.PROPERTIES("2/
CPMB/ACCTYPE")="INC",SUM(PERIODSTODATE([%TIME%].[LEVEL00], [%TIME%].
CURRENTMEMBER),-[MEASURES].[/CPMB/SDATA]),IIF([%P_ACCT%].CURRENTMEMBER.
PROPERTIES("2/CPMB/ACCTYPE")="EXP",SUM(PERIODSTODATE([%TIME%].
[LEVEL00], [%TIME%].CURRENTMEMBER),[MEASURES].[/CPMB/SDATA]),IIF([%P_
ACCT%].CURRENTMEMBER.PROPERTIES("2/CPMB/ACCTYPE")="AST",([MEASURES].
[/CPMB/SDATA], CLOSINGPERIOD([%TIME%].[LEVEL02])),IIF([%P_ACCT%].
CURRENTMEMBER.PROPERTIES("2/CPMB/ACCTYPE")="LEQ",-([MEASURES].[/
CPMB/SDATA], CLOSINGPERIOD([%TIME%].[LEVEL02])),-[MEASURES].[/CPMB/
SDATA])))))';SOLVE_ORDER=3
```

Listing 5.1 YTD Measure Formula for Periodic Data

You've now seen how to use the Measures dimension in SAP BPC to report data for different periods. Next, let's discuss command groups and the commands that are available in the EPM add-in for Excel.

Command Groups

The EPM add-in tab is organized into distinct command groups. Every command group includes commands associated with that group (see the top of Figure 5.7). The following command groups are included in the add-in interface:

- Connection
  This group aids in connecting and managing the connections to the different EPM systems, environments, and models in SAP BPC.
  - Log On and Log Off: This command aids in logging in and logging out of an EPM system.
- Reports
  The commands included in this group provide the options to create new reports and maintain existing reports:
  - Open: Enables users to open a report on their desktop or on an SAP BPC server.
  - Save: Enables a report to be saved in the desktop of the user or on an SAP BPC server.
  - Edit Report: Allows the current report in the worksheet to be edited for changes.
  - New Report: Allows a new report to be created in the current worksheet.
  - Report Actions: Allows a report to be copied, deleted, or pasted. A report copied using the copy option in the EPM add-in tool for Excel can be pasted as a report in Word or PowerPoint. The Worksheet Generation menu option available under this command provides the ability to create multiple reports in multiple worksheets based on the selection of unique values of dimension members for a dimension. The menu option to Manage Connections is also available under this command.
  - View Formats: Allows viewing and editing of formatting options for the current report.
  - Refresh: Offers a variety of options to refresh data in reports, including refreshing the data from the database for a single report or refreshing the data from the database for all reports in the entire worksheet.
- Data analysis
  The commands included in this group provide a wide range of options to analyze the data presented in a report:

- Expand: Determines the dimension members that are to be displayed when a dimension member is expanded in a row or column.
- Collapse: Determines the dimension members that are to be displayed when a dimension member is collapsed in a row or column.
- Keep: When the cursor is placed on a dimension member displayed in a row or column of a report and this command is selected, the selected dimension member is retained in the report.
- Exclude: When the cursor is placed in a dimension member displayed in a row or column of a report and this command is selected, the selected dimension member is excluded from the report.

- Undo
  This command group includes the Back command to undo a report action. The Back command reverses the prior navigation action. The system allows the ability to undo five recent navigation actions.
- Data input
  The commands in this group are used when data is to be updated in an SAP BPC model:
  - Save Data: Saves data to a database. This command is applicable only when a report is set as an input form.
  - Planning: The spread, trend, and weight techniques are often used to distribute or allocate the data in planning. This command is applicable only when a report is set as an input form.
  - Comments: Enables user to add a comment or search for a comment.
  - Journals: Opens the interface in the Web Client to maintain journals.
- Collaboration
  This command group provides the interface to publish an SAP BPC report in the SAP BusinessObjects Enterprise server that can be consumed by portal users. It also provides the interface for reports to be published inside SAP BPC and to distribute reports to both internal and external BPC users. The following commands are included under this group.
  - Portal Publication: Publishes a report in the SAP BusinessObjects Enterprise portal system.
  - Book Publication: Configures and schedules publication of book of reports.

- Distribution: Distributes reports by email to both users and nonusers of SAP BPC. This is particularly useful when a distributed report is to be updated by the users. The report distributed can be collected back to update the data in the SAP BPC system.

- Tools
  The commands included in this group provide the options to perform functions that offers enhanced user interface. The commands included under this group are as follows:
  - Quick Links: Provides the options to pass the dimension members displayed in one report as selected when jumping to another report.
  - Drill Through: Runs a drill-through URL defined in the SAP BPC Web Client.
  - Offline Mode: Sets the entire report offline.
  - Insert Function: Enables a user to insert an EPM function in a report.
  - Options: Defines settings for reports via different options. The Sheet Options settings are relevant to a worksheet. Context Options settings govern the values of dimension members that are displayed in the Context pane and whether the values in the Context pane can be changed during execution of a report. User Options define the settings that are relevant for all of the reports used in an EPM add-in environment for the current user. The Command Display option is available for selecting the command groups to be displayed in the EPM ribbon.
  - More: Enables the migration of an EvDRE report developed in BPC 7.5 to an EPM report. In addition, it provides the ability to update the dimension data in the client via the Clear Metadata Cache option, view user logs, and display the property values for a selected dimension member in the report.
  - Show Pane & Context: Displays the active connection and the name of the current report. The dimensions selected for the different areas of the report (Page Axis, Row Axis, and Column Axis) are displayed and maintained here. The EPM Context pane displays the dimension members selected for the report. This command acts as a toggle to display or hide the EPM pane and the EPM Context pane.

- Help

  The About and Help commands under the Help command group display the version of EPM add-in that is installed in the user desktop and provide the links to open the EPM documentation for SAP BPC on the web.

Next, we will explore the Context pane and the EPM pane that are available in the EPM add-in for Excel.

EPM Context Pane

The EPM Context pane in the EPM tab allows users to select the data to display in a report. If you want to make member selections for a dimension, use the drop-down box to select a member value. By default, the members selected in the EPM Context pane are used to display information in a report (Figure 5.7, ❶). In the SAP BPC 7.5 version, the EPM Context pane was referred to as Current View (CV).

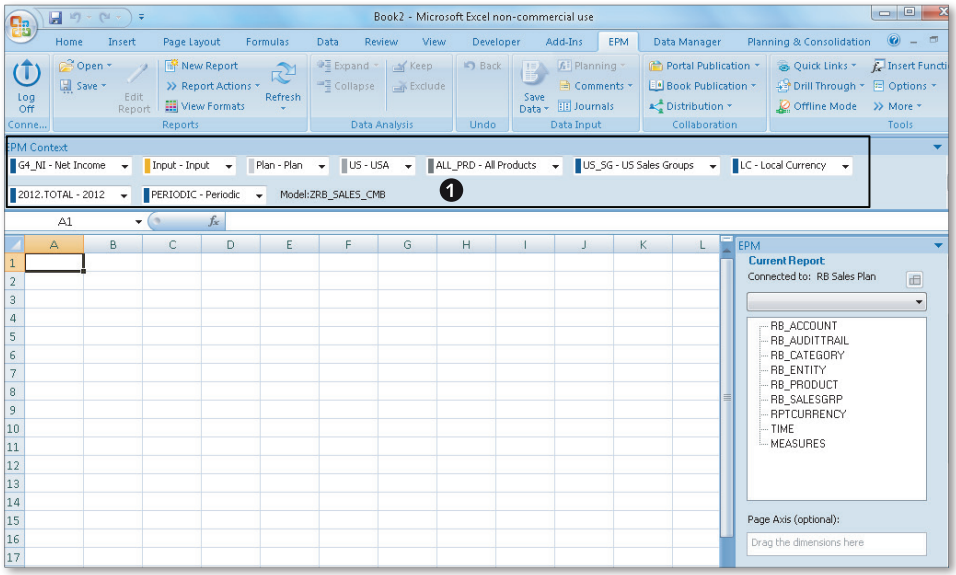


Figure 5.7 EPM Context Pane

EPM Pane

The EPM pane displays the name of the active connection and the dimensions that belong to that model (Figure 5.8, ❶). You can drag the dimensions to the Page Axis, Row Axis, and Column Axis to create a report.

When you add a dimension to a row or column, the report is automatically created and displayed in the Excel worksheet. You can check the Defer Layout Update checkbox to postpone the update of the report (Figure 5.8, ❷). When this is checked, you can make the necessary changes to the dimensions selected in the Page Axis, Row Axis, and Column Axis. During the course of the changes, the report is not updated. When you are ready to accept the changes, click on the Update button (Figure 5.8, ❸). You can swap the dimension displayed in the Row Axis and Column Axis by clicking the Switch icon (Figure 5.8, ❹).

Now that you understand the basic features of the EPM add-in for Excel, we will go over the steps of developing a report.

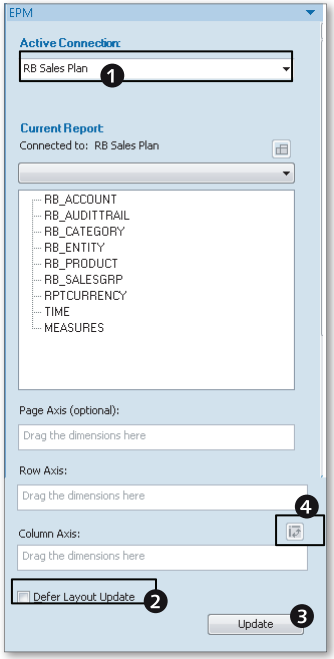


Figure 5.8 EPM Pane

5.1.2 Creating Reports

A report is divided into four areas:

- Row axis: Contains dimension members displayed in the rows. You can select the dimension members to be displayed in the rows. More than one dimension can be included in the row.

- Column axis: Contains dimension members displayed in the columns. You can select the dimension members to be displayed in the columns. More than one dimension can be included in the column.
- Page axis: Can include dimensions not in a row or column, and one or more dimension members can be selected for those dimensions.
- Data grid: Contains values and is displayed in the intersection of the row axis and column axis.

There are three different options available to create a new report.

- Click on the New Report command under the Reports group in the EPM ribbon to create a new report.
- Drag a dimension from the list of dimensions to the page axis, row axis, or column axis in the EPM pane. When at least one dimension is included in the row and column axis, a report is created and displayed in the worksheet.
- Enter the data dimension members in the row or column in an EPM Excel worksheet directly to create a report. When a dimension member is entered in a row and a dimension member is entered in a column that is above the row, a report is created with the data grid in the intersection of the row and column data.

Next we'll use the EPM Report Editor to edit a report. First, however, we'll describe the different tabs that are available in the EPM Report Editor.

Report Edit Options

When you create a new report using the New Report command, the Report Editor dialog box includes the following tabs to define the report. For an existing report, click on the Edit Report command to display the Report Editor dialog box.

1. The Layout tab allows you to select the dimension and members to include in the row axis, column axis, and page axis.
2. The Options tab enables you to specify whether the current report should inherit the settings from the Sheet options. If this option is unchecked, the settings defined in this tab will apply to the report.
3. The Sorting tab allows you to sort data displayed in the report.
4. The Filtering tab lets you filter data displayed in the report. You can set conditions to display data based on a value of data (for example, row or column value > 50,000).

5. The Ranking tab enables you to rank data displayed in the report. You can specify the number (*n*) of the top or bottom records to be displayed.
6. The Read-Only Data tab enables you to select dimension members that can only be used to display data but cannot be used for updates. This tab applies to input forms and where selected members should not be updated in the database.
7. The Local Members tab includes local members defined in a report. Local members works like Excel formulas and can be enabled or disabled here.
8. The Member Names tab allows you to specify those member names that should be overridden with a different name in the report.
9. The Excluded Members tab allows you to specify the members to be excluded from being displayed in the report.
10. The Member Insertion Filtering tab enables you to specify the criteria to display members when you right-click the context menu to add dimension members to rows or columns.

Let's examine the features of the EPM Report Editor in closer detail.

Report Layout and Selections

The first tab in the EPM Report Editor is the Layout tab. In this tab, you identify the dimensions in the row axis, column axis, and page axis of the report. The member selections for the dimensions in the row axis, column axis, and page axis determine the data displayed in a report.

You can perform report member selection by following these steps:

1. After connecting to a model, click on the New Report command under the Reports group in the EPM ribbon to create a new report. You are presented with the EPM – Report Editor dialog box (Figure 5.9) where you design the report. In the Layout tab, you identify the dimensions in the row axis, column axis, and page axis.
2. Click on the dimension selected in the row or column to set selections for the dimensions (Figure 5.9, ❶). You will see that the member and children of the dimension member that is selected in the EPM Context pane for that dimension is included in the EPM – Member Selector dialog box (Figure 5.10, ❷). This is the default selection for the dimension.

3. Let's assume that you don't want to use the member in the EPM Context pane as the selection for your report and instead want to make your own selection. First, remove the default selection by selecting the member and clicking on the arrow to remove it from the Selected Members list (Figure 5.10, ❸).

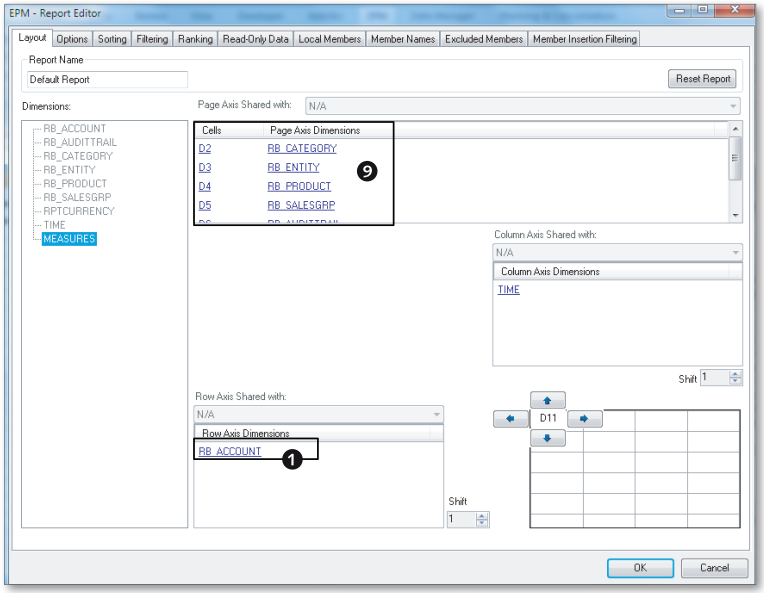


Figure 5.9 EPM Member Selections for Report—Part A

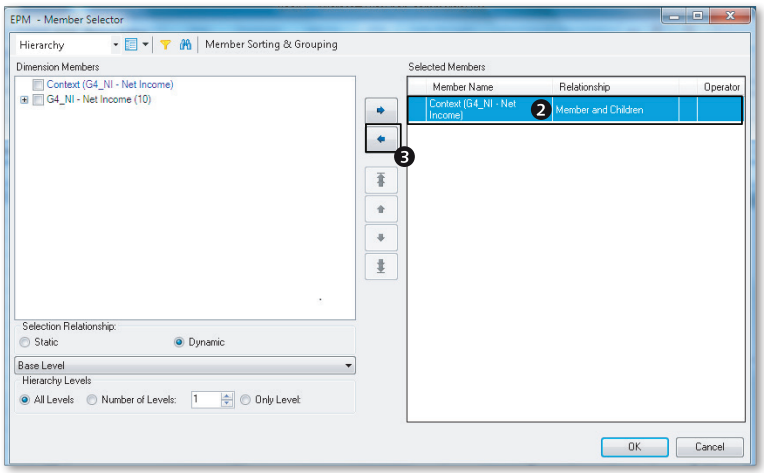


Figure 5.10 EPM Member Selections for Report—Part B

4. Select a dimension member you want to include in the report (Figure 5.11, ❹). Select Static or Dynamic under Selection Relationships, and select the relationship you want to use for member selections. The Static option sets the selection to the members selected based on the relationship at the time of the definition of the report, whereas the Dynamic option selects the members dynamically based on the relationship at the time of the execution of the report. If hierarchy data is to be displayed, set the levels to include under the Hierarchy Levels section.
5. Select the relationship for the data you want to display in the report by clicking the dropdown box (Figure 5.11, ❺). The dropdown box lists the different member selections you can to use for the report, including Member Only, Member and Children, Member and Descendants, Descendants, Base Level, Same Level, and so on (Figure 5.11, ❻). Click the arrow to select the member you checked in the left box (Figure 5.11, ❼).

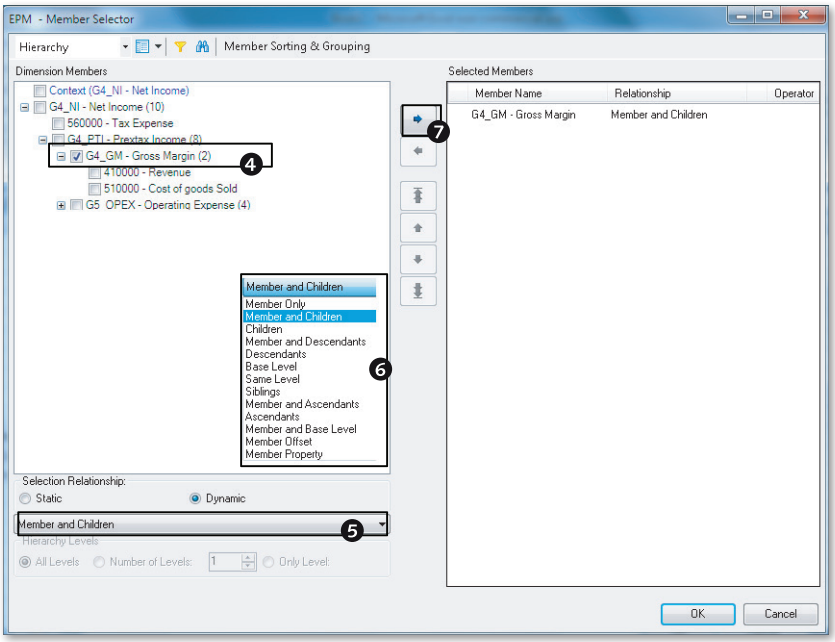


Figure 5.11 EPM Member Selections for Report—Part C

6. When you select the time dimension in the column, you will see that the report will display the Member and Children of the member selected for Time as displayed in the EPM Context pane. We will leave this default selection unchanged (Figure 5.12, ❽).

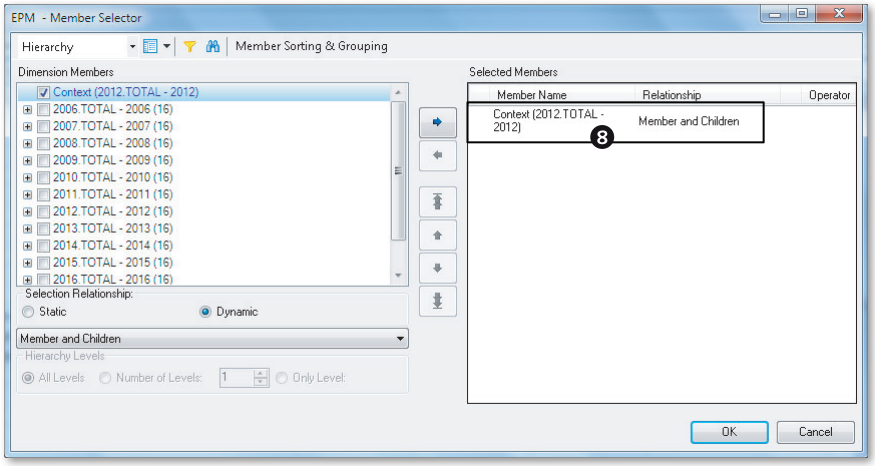


Figure 5.12 EPM Member Selections for Report—Part D

7. Optionally, you can include dimensions not included in the row axis and column axis in the page axis (refer to Figure 5.9, ❾). Dimension members can be selected for those dimensions included in the page axis. The selection of dimension members at the page axis level provides an additional filter for the report. More than one member can be selected by checking the ALLOW MULTI-SELECTION box in the dialog box that is displayed for selecting dimension members. However, the system does not provide the option to specify member relationships for dimensions in the page axis.
8. After making the selections, click on OK to display the report (Figure 5.13).

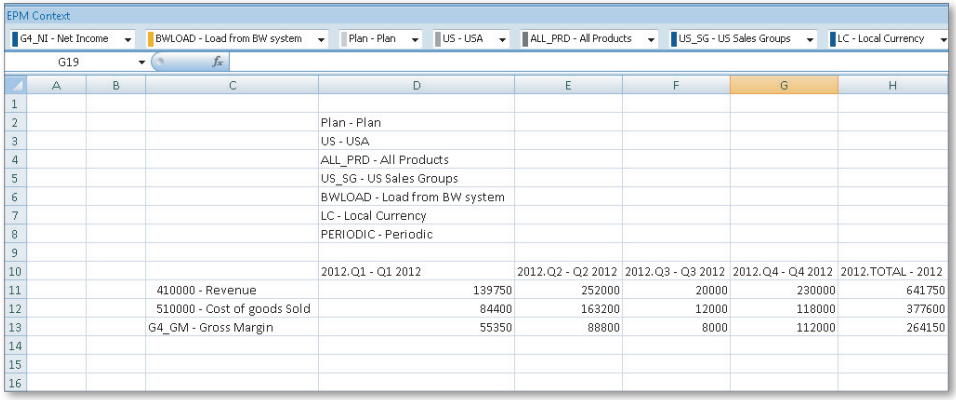


Figure 5.13 EPM Member Selections for Report—Part E

When a report is executed, the system uses an order of precedence to select the dimension members. For example, the dimension member selected in the Row Axis takes precedence over the selections for the same dimension in the Context pane. Table 5.1 lists the order of precedence.

Rank	Description
1	The dimension member selections defined in the Context pane
2	The dimension member selections defined in the Page Axis
3	The dimension member selections defined in the Row Axis and Column Axis

Table 5.1 Selection Precedence When Displaying a Report

Next, we will discuss the report settings that can be made using the Options command available under the Tools command group. This command provides the menu options to set the following:

- Sheet options
- User options
- Context options
- Command display options

Sheet Options

The Sheet Options setting is applicable to all of the EPM reports in the worksheet. The Sheet Options menu option can be accessed from the Options command. The Sheets Options settings are divided into four tabs: General, Formatting, Refresh, and Protection.

In the General tab, you define the following settings:

- Expand Options
  This setting determines the dimension members that are displayed when a dimension member is expanded in a row or column. Member and Children is selected in Figure 5.14.

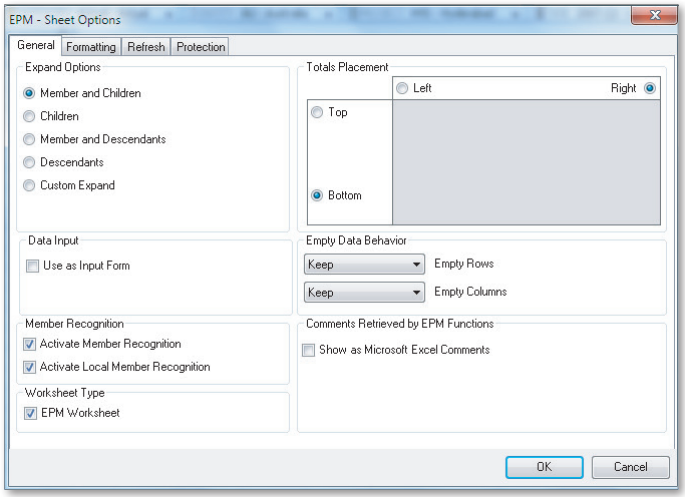


Figure 5.14 EPM Sheet Options

- Data Input
  When a report is to be used as an input form for update, the Use as Input Form checkbox is checked.
- Member Recognition
  There are two settings in the Member Recognition section.
  - Activate Member Recognition: Enables dimension members to be recognized when entered directly in a report. When the value entered is ambiguous, the system prompts the user to select the member from a list.
  - Activate Local Member Recognition: Enables formulas to be created inside the report.
- Worksheet Type
  This option should be checked if this report is to be considered an EPM worksheet. You can uncheck this option if you do not want a worksheet to connect to an EPM system or if you want to temporarily disconnect a worksheet from the EPM system(s) to which it is connected.
- Totals Placement
  Displays the totals when required.
- Empty Data Behavior
  Determines whether empty row or column data is displayed.

- Comments Retrieved by EPM Functions
  Determines how an EPM comment is displayed in a report.

In the Formatting tab, you define the following settings:

- Row Header Indentation
  Determines whether dimension members displayed in the rows are shown with or without indentations for hierarchy data.
- Display Name
  You set how the member should be displayed in the report using the two available options: Caption or Full Unique Name. The ID, Description, or ID and Description of the dimension member can be displayed when Caption is selected. The Full Unique Name option displays the fully qualified name, including the dimension name, hierarchy, and member ID.
- Auto Fit Column Width
  The width of all of the columns in the report are automatically adjusted to fit the contents after the data is refreshed.
- Repeat Row Headers
  When more than one dimension is displayed in the rows, this setting determines whether a dimension member in the outer dimension is repeated even if it is the same member as the one in the previous row, or whether a dimension member is shown only when it is different from the previous row.
- Repeat Column Headers
  When more than one dimension is displayed in the columns, this setting determines whether a dimension member in the outer dimension is repeated even if it is the same as the one in the previous column, or whether a dimension member is shown only when a dimension member is different from the previous column.
- Empty Cell Default Value
  Determines what is displayed when a value for a data cell in a report is empty. This setting is available under the Report Format section and is applicable only when the Set Default Value in Empty Cell checkbox is selected.

In the Refresh tab, you define the following settings:

- Keep Formula on Data
  A formula defined in a data cell in the report is lost when data is refreshed. Checking this option maintains the formula after refresh.

- Show Source Data in Comments
  When the Keep Formula on Data checkbox is enabled and, if the data that is displayed based on the formula is different from the value stored in the database, the value stored in the database is shown as an Excel comment icon in the upper-right corner of the cell. Hovering the cursor on the icon displays the value stored in the database for that intersection.
- Total Data in Hierarchies
  Setting this option automatically adds base member data and displays hierarchy data. The hierarchy data for the member is calculated only if all of the children of a parent are displayed in the report.
- Refresh the Data in the Report When Loading the File
  Checking this option refreshes the data every time the report is opened.
- Clear the Data When Saving the File
  When this option is selected, the data cells are cleared when you save the report.
- Show Unauthorized Cell Text
  When a user is not authorized to see a value for a data intersection, the unauthorized cell text is displayed to the user in the report.

The Protection tab provides the settings to set protection at the worksheet or workbook level for the reports created in EPM.

User Options

The User Options are set by users based on their preferences (Figure 5.15). The User Options option can be accessed from the Options menu. The User Options settings are divided into the following three tabs:

- Navigation
  Some of the preferences you can set here provide the ability to enable double-clicking in reports, determine the expand/collapse behavior on a single member or all members in an inner dimension when more than one dimension is displayed in the row axis or column axis, and determine the keep/exclude behavior on a single member or all members in an inner dimension when more than one dimension is displayed in the row axis or column axis. There are options to specify if the report is to automatically refresh the data when dimension members are selected in the Context pane, freeze data refresh, and check the settings to activate the metadata cache to stop metadata from being loaded in the client

every time you make a connection. The setting to clear the metadata cache and the frequency to load it is also set here.

- Server Configuration
  In this tab, you define the settings to configure the connection to an SAP BusinessObjects Enterprise system for publishing SAP BPC reports or set the Simple Mail Transfer Protocol (SMTP) settings for distributing email.
- Others
  In this tab, you can define the settings for your user preferences. Some of the preferences you can define here are color coding for quick links, selecting or deselecting member and local member recognition, and displaying or hiding the Context pane and the EPM pane.

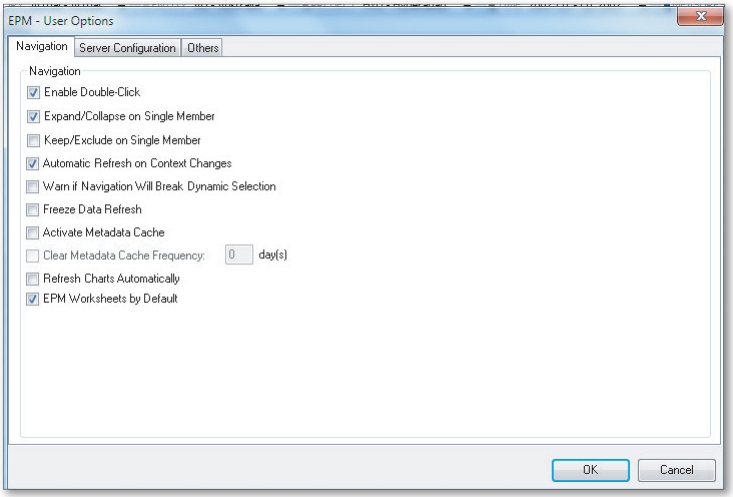


Figure 5.15 EPM User Options

Context Options

The options to define what is displayed in the EPM Context pane are set in this setting. The members explicitly selected for the dimensions identified in the row axis, column axis, and page axis are used in a report. When a member is not selected explicitly, the member selected in the EPM Context pane is used.

In the Context Display tab, checkboxes are available to specify whether the dimension name and color for the dimension are to be displayed. The options defined here apply to all of the reports in the workbook. You can also decide to individually hide dimensions from being displayed in the EPM Context pane.

In the Context Lock tab, you can set options to hide a dimension from being displayed or to lock a dimension member from being changed at the workbook or worksheet level.

Before we summarize the tabs that are available in the Report Editor, let's discuss local members that are used to define formulas in reports.

Local Members

In a report, you may want to display the property values of dimension members or use formulas to perform calculations. You create local members for this purpose. Local members work like Excel formulas and provide the ability to define a formula that will be applied to an entire report. To learn how to create a local member, take the example of the report that we created earlier and follow these steps:

1. To display the account type property of all of the dimension members in the row, you can create a local member as a formula that will display the account type value for the dimension member. Confirm that the Activate Local Member Recognition checkbox is enabled in the Sheet Options for the report before you proceed with the next steps.

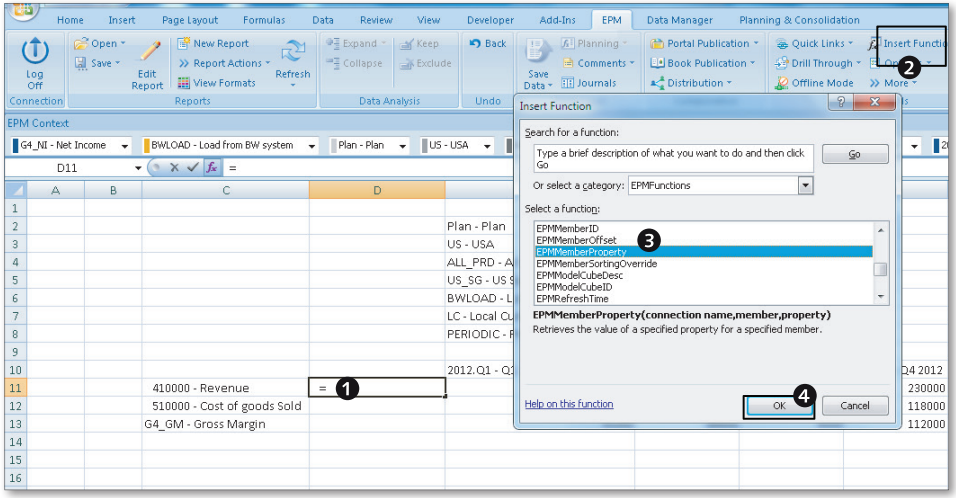


Figure 5.16 Define Local Members—Part A

2. Insert a new column inside the report to display the property value. Select the cell on the column inserted that is next to the first dimension member in the row to define the local member (Figure 5.16, ❶). Click on Insert Function in

the EPM tab (Figure 5.16, ❷). This will open a new dialog box to select an EPM function. Using the dropdown menu, select EPM Functions from the options displayed in the Select a category list. This will display a list of all the EPM functions. Select the EPMMemberProperty function, and click on OK (Figure 5.16, ❸ and ❹).

3. In the Functions Arguments dialog box, select the connection, dimension member, and property value to display (Figure 5.17). When a connection is not specified for the connection name parameter, the active connection is taken as default. The dimension member is selected for the member parameter. The dimension property that is required to be displayed is specified for the property parameter. For this example, enter "ACCTYPE" for the Property. After specifying the selections, click on OK.

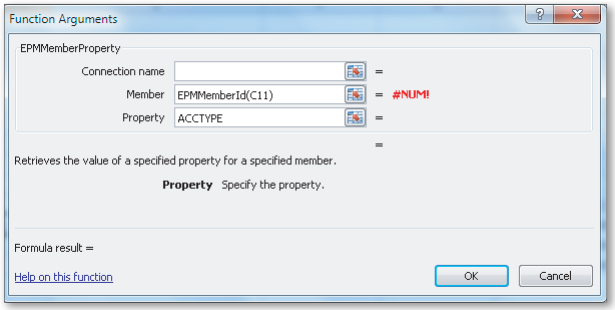


Figure 5.17 Define Local Members—Part B

The account type property value for the account data is displayed as local members in the rows (Figure 5.18).

4. Inside an EPM report, click on Edit Report to edit the report. The Local Members tab lists the local members defined for the report.
5. A local member defined for the report can be disabled by unchecking the Enable button (Figure 5.19). If you enter the description for the formula, the EPM equivalent syntax of the local member formula is displayed. You also define the setting for where the local member is to be displayed in this tab.

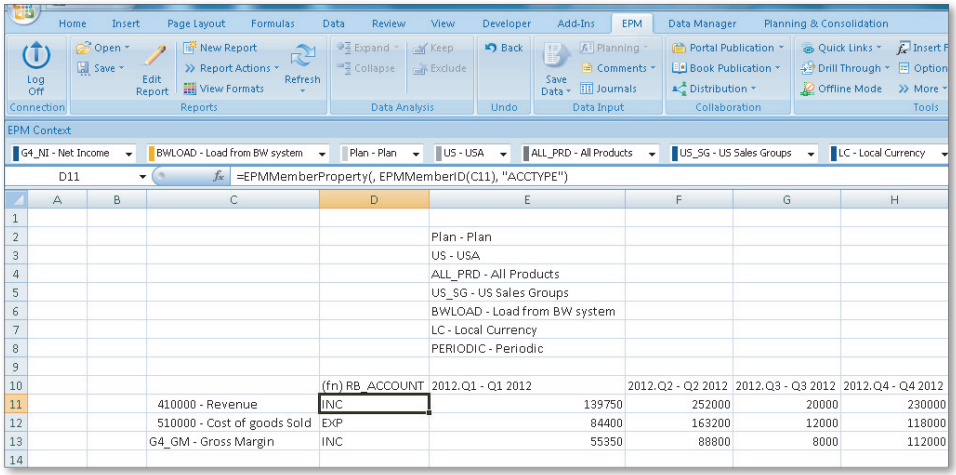


Figure 5.18 Define Local Members—Part C

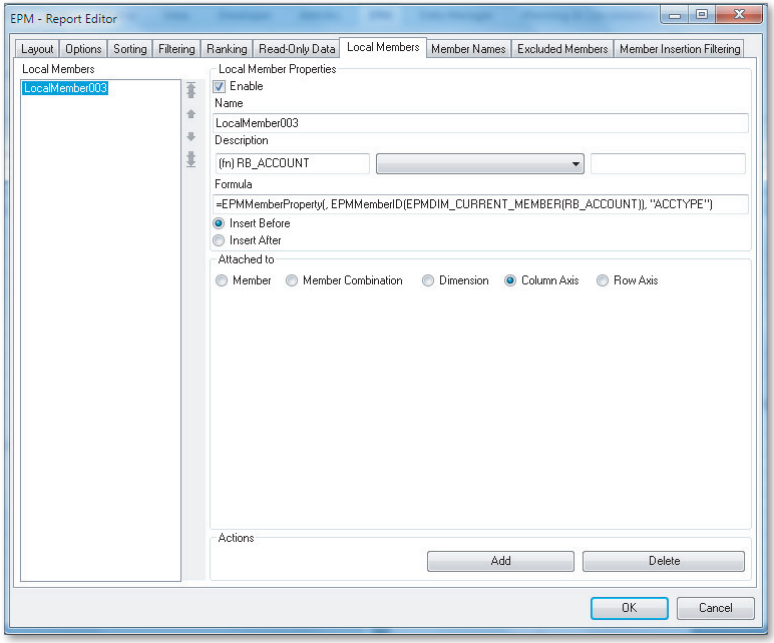


Figure 5.19 Define Local Members—Part D

Saving a Report

When you define a report and are satisfied with its format, follow these steps to save the report:

1. Under the Open command, select the Open Server Report Folder (Figure 5.20). This opens the Save dialog box.
2. In the Save dialog box, select the Local or Company (Public) folder. When the Local folder is selected, the report is saved in the user desktop. When the Company folder is selected, the report is saved in the SAP BPC server.

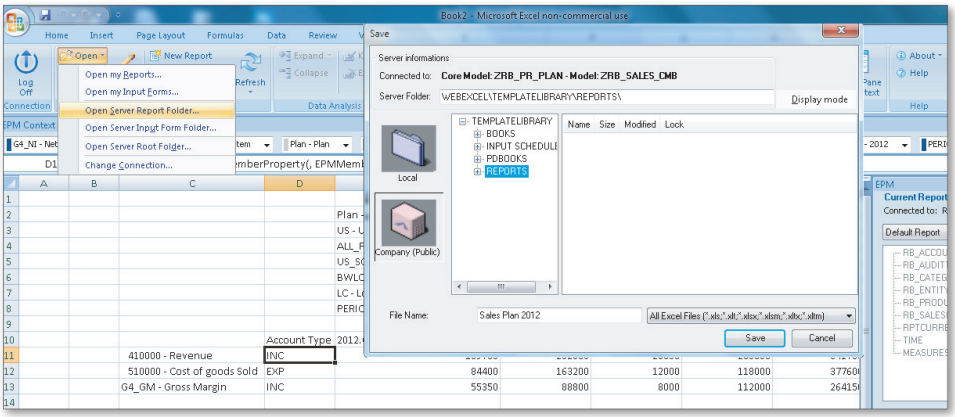


Figure 5.20 Saving a Report

Formatting Options

SAP BPC 10.0 offers dynamic formatting options to use in SAP BPC reports. Using the features available in the formatting template, you can specify a variety of custom formatting options in your reports.

Let's examine the features of the formatting template. A formatting template with the name EPMFormattingSheet is created by clicking on the View Formats command in the Reports command group. You can hide the formatting template by clicking again on View Formats.

The formatting template is divided into four sections:

- Hierarchy Level Formatting
- Dimension Member/Property Formatting

- Row and Column Banding
- Page Axis Formatting

Note that the format settings in the lower section take precedence over the settings in the upper section when there is a conflict.

The following features pertain to formatting:

- The formatting sections allow the formatting to be defined for row and column. However, formatting for row/column does not apply to the Page Axis Formatting section.
- All the formatting options available in Excel can be used in the template, including the option to use conditional formatting.
- The formatting options defined for a section will apply only when the section is checked (by using the checkbox at the top of each section). Inside a section, there is a checkbox to indicate whether formatting for row or column settings take priority. This setting does not apply to the Page Axis Formatting section.
- The Hierarchy Level Formatting section is used when formatting options are required for the hierarchy data displayed in a report. In this section, you can specify whether you want the formatting setting to apply to the Inner Dimension or Outer Dimension. This applies when you have more than one dimension displayed in the row or column. Additionally, there is a radio button to indicate if the formatting applies to the structure level or relative level of the hierarchy. The formatting options specified in each line of the section applies only if the Apply checkbox for the line is checked.
- In the Dimension Member/Property Formatting section, you specify how specific dimensions should be formatted based on property values of dimension members. Multiple property values can be specified when setting the formatting options in this section.
- The Row and Column Banding section is used for formatting row and column data.
- The Page Axis formatting section is used for formatting dimensions displayed in the page axis.
- You can create more than one formatting sheet. To make a copy of a formatting sheet, you can use the standard Excel Copy function.
- After a dynamic formatting sheet is created, you can apply it to a report. Recall that earlier in this section we discussed the settings for Sheet Options that are

set using the Options command in the Tools command group. By default, a report inherits the setting in the Sheet Options (the Inherit Sheet Options checkbox is checked in the Options tab when editing a report). If a report is set to inherit the settings from the Sheet Options, go to the Sheet Options, and select the Formatting tab. Check the Apply Dynamic Formatting checkbox. Select a formatting sheet from the dropdown list under Default Formatting Sheet. This formatting sheet will be used in a report that inherits settings from Sheet Option.

- If you want to apply the formatting sheet to a specific report, edit the report by selecting the Edit Reports command. Uncheck the Inherit Sheet Options setting in the Options tab. In the same tab, check the Apply Dynamic Formatting box. Select the relevant formatting sheet from the dropdown list under Default Formatting Sheet.

Now let's go over the steps to format data based on hierarchy levels. Refer to Figure 5.21 and Figure 5.22. You will notice the following formatting:

- The Hierarchy Level Formatting section checkbox is checked at the top.
- The Priority to Row Format radio button is selected when row formatting is required to take precedence over column formatting (Figure 5.21, ❶).

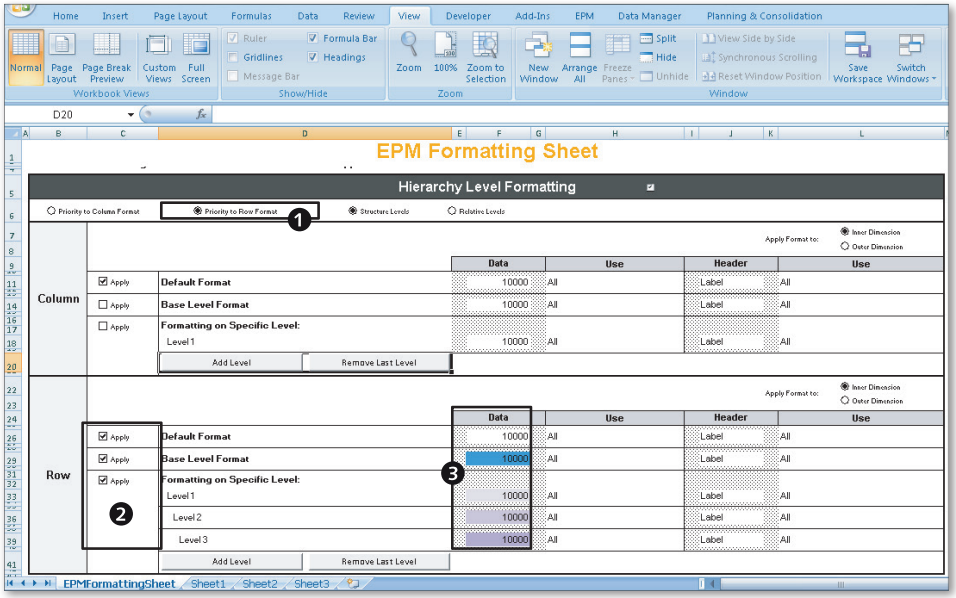
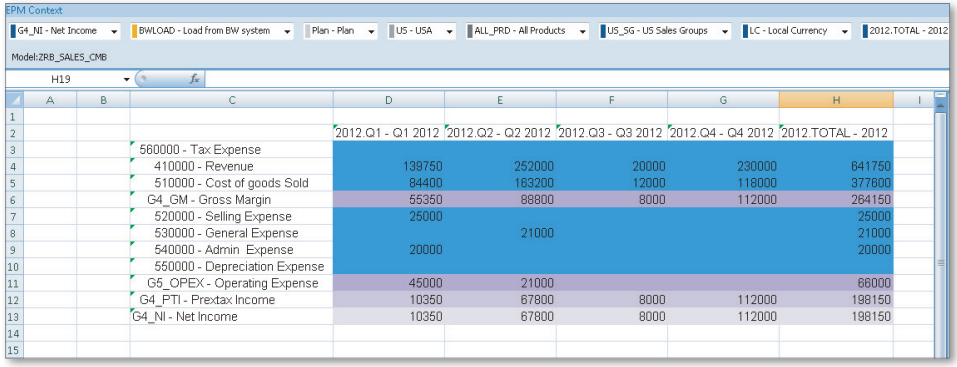


Figure 5.21 Formatting a Report—Part A

- You'll notice that the row formatting is enabled at various levels, and color coding is specified for the data columns (Figure 5.21, ❷ and ❸). When a report is executed, the formatting defined is applied to the report (Figure 5.22).



	2012.Q1 - Q1 2012	2012.Q2 - Q2 2012	2012.Q3 - Q3 2012	2012.Q4 - Q4 2012	2012.TOTAL - 2012
560000 - Tax Expense					
410000 - Revenue	139750	252000	20000	230000	641750
510000 - Cost of goods Sold	84400	163200	12000	118000	377600
G4_GM - Gross Margin	55350	88800	8000	112000	264150
520000 - Selling Expense	25000				25000
530000 - General Expense		21000			21000
540000 - Admin Expense	20000				20000
550000 - Depreciation Expense					
G5_OPEX - Operating Expense	45000	21000			66000
G4_PTI - Prextax Income	10350	67800	8000	112000	198150
G4_NI - Net Income	10350	67800	8000	112000	198150

Figure 5.22 Formatting a Report—Part B

Let's go over the steps to format data based on specific member/property values. Figure 5.23, Figure 5.24, and Figure 5.25 reflect this sequence.

1. In the Dimension Member/Property Formatting section, you can enable formatting to be applied based on specific member property values. Ensure that the checkbox at the top of this section is checked if you want to use this section for formatting the data.
2. Click on Add Member/Property to specify the selection values (Figure 5.23, ❶). This will open the EPM Formatting Sheet – Select Member/Property dialog box.
3. In the Property Selection tab, specify the criteria for formatting based on the member/property (Figure 5.23, ❷). In this tab, we have specified the criteria for formatting data based on the account type property. Specifically, we have set the criteria to format all of the account dimension members displayed in the report where the account type property equals to EXP.
4. After you make the selections, confirm them by clicking OK. You see the selections carried over and displayed in the Formatting on Specific Member/Property area (Figure 5.24, ❸).
5. Notice that the data displayed in the report is formatted based on the criteria specified in this section (Figure 5.25). The settings in the Dimension Member/

PROPERTY FORMATTING section have taken precedence over the formatting options in the HIERARCHY LEVEL FORMATTING section because it is one level below it.

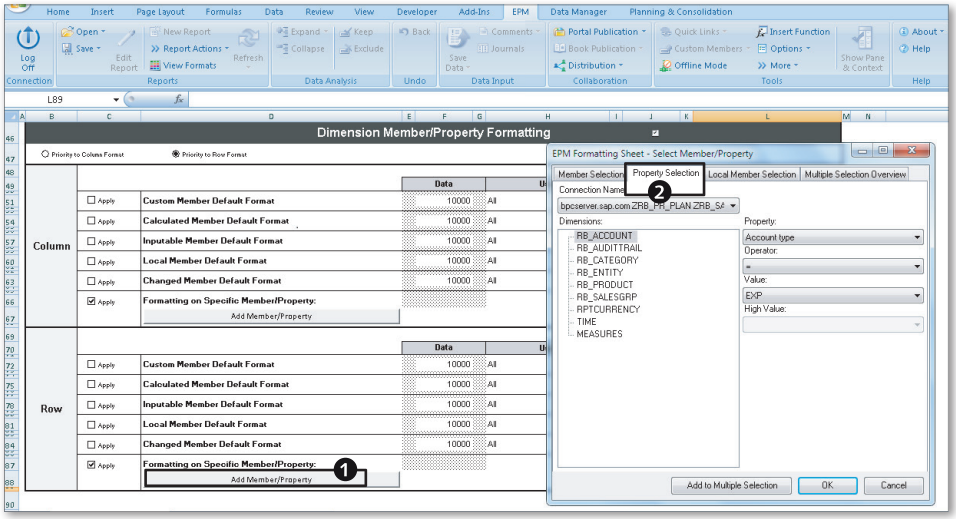


Figure 5.23 Formatting a Report—Part C

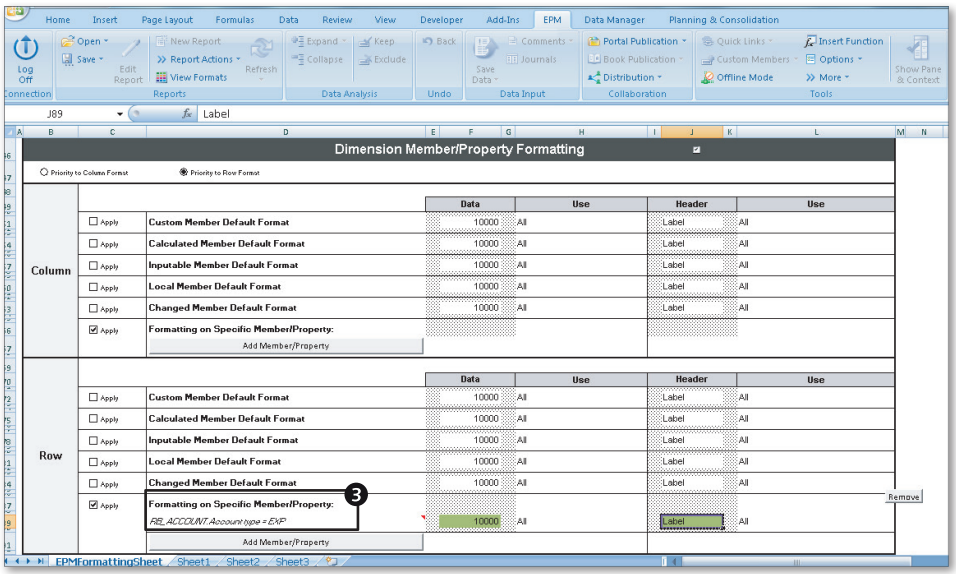


Figure 5.24 Formatting a Report—Part D

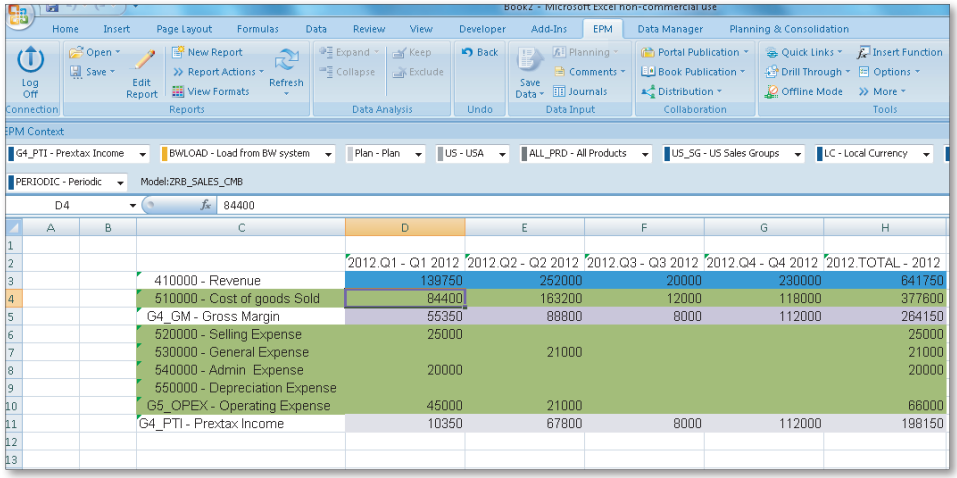


Figure 5.25 Formatting a Report—Part E

5.1.3 Developing Reports Using Dynamic Templates

A *dynamic template* provides a quick entry point to create a report. Predefined templates are available for different types of analysis, and you can select a template that best supports your reporting needs. You can further customize the report based on a particular business requirement.

Dynamic templates let you create a report quickly and reduce development effort. The templates serve a variety of standard analyses and can be used as the basis for developing a new report. Table 5.2 shows a list of standard templates and their descriptions.

Template Name	Description
Dynamic Report with SubTotals – Accounts	Displays the data for accounts in the rows and Time dimension in the column to demonstrate a trend.
Dynamic Report with Hierarchy – Entity	Displays the entity data in the rows and the Time dimension in the column.
Variance Report	Displays the Category dimension in the column and any dimension in the rows. This report is used to calculate the variance between two different categories for a time period. The difference between the values (e.g., actual vs. budget) is displayed for the selected period.

Table 5.2 Standard Templates for Reporting

Template Name	Description
Last Year and Comparison Report	Provides a comparison of the current and prior years' periodic and YTD values for different categories (e.g., plan vs. actual).
3-Year and Comparison Report	Provides for year-over-year comparisons.
Dynamic Report with Drill in Place	Performs drill in place.
Sort on Values Report	Sorts by data value.
Sort on Member Descriptions Report	Sorts by member descriptions.
Dynamic Report with Nested Rows	Allows more than one dimension to be nested in the rows, and allows one dimension in the column.
Dynamic Report with Dynamic Dimension in Row Axis	Allows you to drill from one dimension to another in the rows.

Table 5.2 Standard Templates for Reporting (Cont.)

5.1.4 Developing Reports Using EPM Functions

Several standard functions are available for reporting and updating data in SAP BPC. You can use them to perform tasks such as manage data, enter comments, and set the work status for a model.

SAP provides a variety of functions, called EPM functions, for reporting and managing data. They replace the Ev functions that existed in the SAP BPC 7.x versions. The EPM function provides a variety of commands to display data in SAP BPC. Other available functions display the name of the environment and model for an existing report and retrieve the metadata associated with the name of dimensions inside a model and the properties inside a dimension. Table 5.3 lists a few EPM functions. The descriptions explain the purpose of each function and its usage in reports.

Function Name	Description	Example
EPMSaveData	Sends data to an SAP BPC model using a cell as reference. This function can be used only in input forms.	The following function can be inserted in a cell to enter and send a value: EPMSaveData(<cell>, <connection_name>, <member1>, … <member n>) This function sends the value in the cell of the worksheet. The connection name and members are specified. Any dimension member that is not specified in the send function is taken from the context view.
EPMEnvDatabaseID	Returns the name of the environment for the specified connection.	EPMEnvDatabaseID (<connection_name>) If the connection name argument is blank, the active connection is used by default.
EPMEnvDatabaseDesc	Returns the description of the environment for the specified connection.	EPMEnvDatabaseDesc (<connection_name>)
EPMModelCubeID	Returns the name of the model for the specified connection.	EPMModelCubeID (<connection_name>)
EPMModelCubeDesc	Returns the description of the model for the specified connection.	EPMModelCubeDesc (<connection_name>)
EPMUser	Returns the user name of the user logged in to the system.	EPMUser()
EPMServer	Returns the server address for the specified connection.	EPMServer()

Table 5.3 List of EPM Functions

Function Name	Description	Example
EPMDimensionType	Returns the name of the dimension when the connection and a dimension type are specified. The following dimension types can be used: A (Account), E (Entity), C (Category), T (Time), R (Currency), G (Ownership), I (InterCompany), S (Subtable), U(x) (User Defined, where x is the number of the user-defined dimension)	EPMDimensionType(,"E") This function outputs the name of the Entity dimension for the current connection.
EPMMemberOffset	Specified time offsets	EPMMemberOffset (<connection_name>, 2012.Feb, 3,) This function returns "2012.May" as the output. The first parameter denotes the connection name. The second parameter denotes the time member or cell on which the offset should be performed. This parameter is mandatory. The last parameter denotes the offset. A positive or negative value can be specified. This parameter is mandatory. The last parameter denotes the hierarchy level and is optional.

Table 5.3 List of EPM Functions (Cont.)

Function Name	Description	Example
EPMContextMember	Returns the dimension member values displayed in the context pane for the dimension.	EPMContextMember (<connection_name>,<Dimension>,<Filter>) This function returns the value of the dimension member of the dimension. In addition, if you double-click on a cell that contains this function, it returns the list of members in a dimension filtered based on a condition specified in the filter.
EPMRetrieveData	Retrieves the data from a model based on dimension member selections that are passed to this function. If selection criteria are not specified for a dimension, the selections are taken from the CONTEXT pane.	EPMRetrieveData (<connection_name>, >,<member1>, … member n>) This function retrieves the value for the members. Any dimension member that is not specified in the send function is taken from the context pane.
EPMMemberProperty	Returns the specified property value for a dimension member.	EPMMemberProperty(<connection_name>, <member>,<property_name>)
EPMComparison	Performs a better or worse comparison of two values, based on the account type property of the account member.	EPMComparison(<connection_name>, < AccountMember>, <cell1>, <cell2>) This is a useful function when you're comparing actual data and budgets. For example, you can use this function to compare the revenue account to see how well you've performed with respect to the budget.

Table 5.3 List of EPM Functions (Cont.)

Function Name	Description	Example
EPMDimensionList	Retrieves the list of dimensions for a model.	EPMDimensionList(<connection_name>, <range>, <separator>) Specify a cell or range of cells in the range to retrieve the dimensions. Specify a separator if you want to list all of the dimensions in a single cell.
EPMMemberDesc	Retrieves the description for a dimension member.	EPMMemberDesc(<connection_name>, <Member>)
EPMSelectMember	Retrieves a specified connection member.	EPMSelectMember (CommentRange, AppName, Member1, Member2, …)
EPMWorkStatus	Retrieves the work status.	EPMWorkStatus(<connection_name>, <parameter>, <Member1>, <Membern>)
EPMReportID	Retrieves the technical name of the report.	EPMReportID(<reportMember>) You specify the cell of a report Member to retrieve the ReportID. This parameter is mandatory.
EPMFullContext	Retrieves the context dimension members for a report. If a dimension is in the page axis, the context member is overridden by the page axis member.	EPMFullContext(<connection_name>, <separator>, <reportid>) The separator is an optional parameter to distinguish the dimensions displayed. The reportid parameter is optional. If not specified, the default report is used for this function.

Table 5.3 List of EPM Functions (Cont.)

5.1.5 Developing Reports Using the EPMRetrieveData Function

In the previous sections, we used the standard process inside the EPM Excel interface to develop a report. There is an alternative EPM function that you can use to report data called EPMRetrieveData.

You can use the EPMRetrieveData function to retrieve data to a single cell of data in Excel based on the selection of dimension members. An EPMRetrieveData function calls the database to retrieve the data for the selection. Note that when a report containing the EPMRetrieveData function is refreshed, there will be as many calls made to the database as the number of cells in the report that contains this function.

The EPMRetrieveData function allows data to be retrieved based on the parameters defined in this function. The selections for which the data is retrieved are specified in the function. In the EPMRetrieveData function, the connection name and the dimension values for the dimensions in the model are specified.

The syntax for using the EPMRetrieveData function is as follows:

```
EPMRetrieveData(Connection Name,[dim:]member1,[dim:]member2,...[dim:]
memberN)
```

The EPMRetrieveData function can also be used inside a standard EPM report.

Note

If values are not specified for some dimensions in the EPMRetrieveData() function, then for those dimensions, the system uses the selection of dimension members in the Context pane to retrieve the data.

The EPMRetrieveData function provides flexibility in reports to retrieve data for a given intersection of dimension members. However, it is inefficient in the way it is run because each function costs a call to the database when the data is refreshed. Until now, we have discussed creation of dynamic reports and usage of EPM functions in reports. Next, we'll discuss how to develop a static EPM report.

5.1.6 Developing a Static EPM Report – Rolling Forecast Report

When the layout of a report is static and not expected to change, you can use a report with selections that are single members and not based on dynamic selection. You use a report with static selections when you know that the dimension members you want to report in the rows and columns are not going to change. You can also use this option if you can derive the members that are required in your rows and columns using an Excel function.

This report is created like any standard report, except that you do not specify dynamic selections in the members selections for dimensions specified in the rows and columns of the reports or input forms. You check Static and Member Only in

the Selection Relationships when defining the selections for the dimensions in the row axis and column axis of the report.

Let's look at an example to illustrate how to use a static SAP BPC report. Let's say that in May 2012 you want to display rolling forecast data for accounts. The data in the rolling forecast should display the actual data for the periods January 2012 through April 2012 (periods elapsed) and forecast data for periods May 2012 into December 2012 that are in the future.

The actual data and the forecast data reside in the same model. The category associated with actual data is ACTUAL, and the category associated to forecast data is FORECAST.

An easy way to develop this report is to copy actual data for the closed months to the forecast category. In this approach, the actual and forecast data resides in the same category (FORECAST), and it is easy to report on the same category using expansion.

Let's say we did not copy the actual data into the FORECAST category at the close of every period. We can still create a rolling forecast report if we know the period up to which actual data is available. Let's assume that the category dimension has a member called ACTUAL, and this member has a property called `CURR_MONTH` that is updated with the period up to which actual data is available. Because we are in the month of May 2012, we'll have the property `CURR_MONTH` for the ACTUAL member set to 2012.04 to indicate that actual data is available until this period.

In a rolling forecast report, we want to display data for all of the forecast periods. But for periods that have already elapsed, we want to display the actual data. We'll list accounts along the row, and the category and time periods along the column. The following steps are necessary to define this report:

1. Create an SAP EPM static report. The Account dimension is selected in the row and the Category and Time dimensions are selected in the columns.
2. For the members selected in the row, you can specify the selections as Dynamic. However, for the Category and the Time dimensions specified in the columns, select only one member and identify the selection as Static and Member Only. We will be deriving the value for the Category dimension and the Time dimension for the actual and forecast data using Excel formulas.
3. Assuming your forecast is for a year, you will display 12 months of data. Display the actual data for the periods that have elapsed and forecast data for future periods.

4. We are assuming that a property exists in the Category dimension that contains the period up to which actual data has been loaded into the model. We have included this property (with the name CURR_MONTH) in the Category dimension and updated it with the period for the ACTUAL category dimension member. Define an EPM function in, let's say, cell B2 that would retrieve the value of this property from the ACTUAL category dimension member (Figure 5.26, ❶).

   ```
   =EPMMemberProperty(,"Actual","CURR_MONTH")
   ```

5. Define another EPM function in cell B3 to read the year property for the time member selected in cell B2 (Figure 5.26, ❷).

   ```
   =EPMMemberProperty(,B2,"Year")
   ```

6. Define an Excel formula to derive the first time periods based on an Excel formula in cell B4 (Figure 5.26, ❸).

   ```
   =LEFT(B2,4)&".01"
   ```

7. Select the other 11 time periods from C4 through M4 using the EPM offset function. For example, in cell C4, you will define the following offset function (Figure 5.26, ❹):

   ```
   EPMMemberOffset(,B4,1)
   ```

   You have now defined the 12 time periods for the report.

8. You now have to select a category for each of the time periods. If actual data exists in the model for the period, we want to display the actual data. Otherwise, we want to display the forecast data. To achieve this, compare the value in cell B4 (2012.01) to the value in cell B2 (2012.04), and set the value of the category to either ACTUAL or FORECAST. Cell B2 contains the value of CURR_MONTH property of the ACTUAL dimension member. This formula is entered in cell B5 (Figure 5.26, ❺). In this specific scenario, direct comparison of data in cells B4 and B2 will work. However, if you named the TIME dimension members as 2011.JAN, 2011.FEB, etc., you must derive the TIMEID property associated to the TIME members and use the TIMEID property values to perform the comparison.

   ```
   =IF(B4 > $B$2, "FORECAST", "ACTUAL")
   ```

9. Create a similar Excel formula for the cells from C5 to M5 to derive the category for the 11 time periods in cells D4 to M4. For example, for cell C5, you will use the following formula (Figure 5.26, ❻):

   ```
   =IF(C4 > $B$2, "FORECAST", "ACTUAL")
   ```


10. Now, pinpoint the time members derived earlier to be used as column member selections (column keys) for the report. Use the following formula for the time member in the outer dimension in cell B8 (Figure 5.26, ❼). You already will see the `EPMOlapMember` function used in your report. You will have to change the first parameter for this function to point to the cell member you want to use in the report. The parameters used in this function are shown in Figure 5.27.

```
= EPMOlapMember(B4,"[TIME].[PARENTH1].[2012.01]","2012.01 - Jan
2012","","000")
```

11. Similar to the previous step, pinpoint the category members derived previously to be used as column member selections for the report. Use the following formula for the category member in the inner dimension in cell B9 (Figure 5.26, ❽):

```
= EPMOlapMember(B5,"[RB_CATEGORY].[].[ACTUAL]","Actual -
Actual","","000")
```

12. Drag the formula for the time and category selections from cell B8 all the way to M8 for the Time dimension and from cell B9 to M9 for the Category dimension.

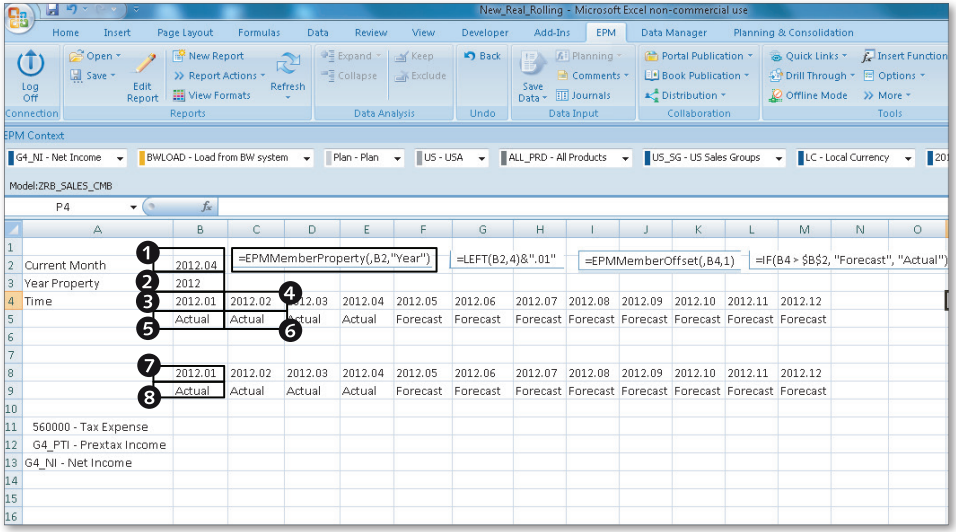


Figure 5.26 Rolling Forecast Report—Part A

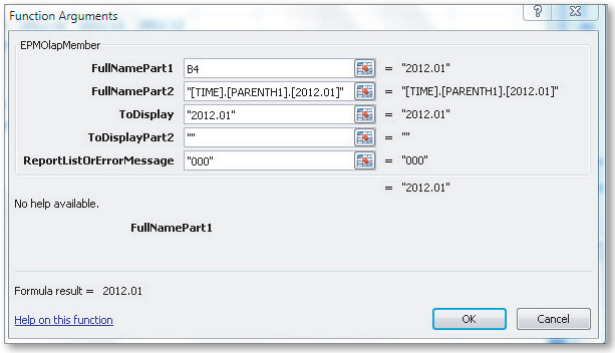


Figure 5.27 Rolling Forecast Report—Part B

When you refresh the report, you will see a rolling forecast report that displays actual data for the periods elapsed and forecast data for current and future periods (Figure 5.28).

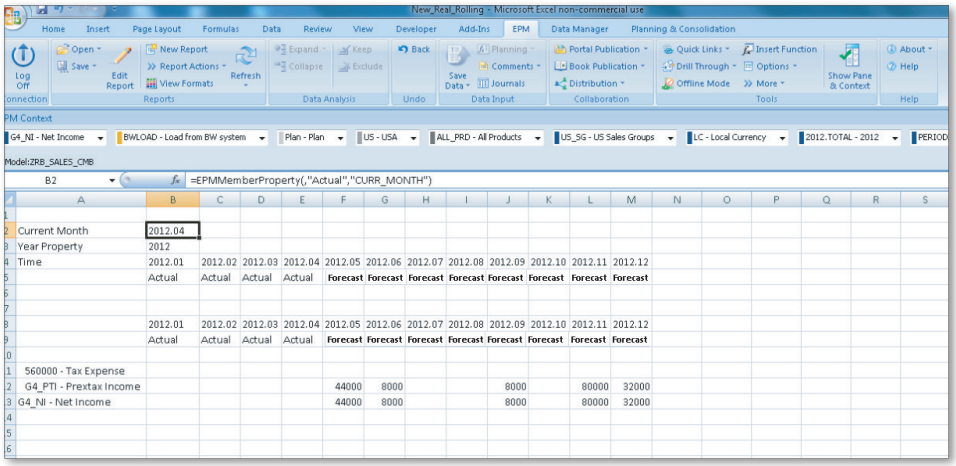


Figure 5.28 Rolling Forecast Report—Part C

Note

When developing a static report, it is recommended that you turn off the Activate Member Recognition option in the User Options and Sheet Options.

This section taught you how to define a rolling forecast report that displays actual data for elapsed periods and forecast data for current and future periods. You also

saw how to use the EPM functions to return the property value from a dimension and how to use time offsets in your report.

SAP BPC provides advanced functions to customize reports and aid in flexible analysis of the data. We'll now discuss some of the advanced EPM functions you can use to customize reports.

5.1.7 Developing Customized Reports

The new EPM add-in interface has provided several advanced features to facilitate reporting. One of the powerful features in the new version is the ability to create multiple reports inside a worksheet or workbook that contain data from more than one environment and model. In addition, the interface provides the option to connect to any EPM system or an SAP NetWeaver BW system for reporting data in the same worksheet where the SAP BPC data is displayed. This is known as multiple reports, and it provides the advantage of viewing and analyzing different sets of data in a single report. When you include multiple reports in a worksheet, you can configure the row or column data to be shared between the reports.

A report can include more than one dimension in the row or column of a report. Such a type of report is called a nested report. When you expand or collapse on an inner dimension, you can set the report to one of the two types of behavior: asymmetric and symmetric.

- *Asymmetric* is the default behavior and expands or collapses only for the members that are selected for expansion in the inner dimension.
- *Symmetric* behavior will expand or collapse for all of the members displayed in the inner dimension.

The default behavior can be changed in the User Options tab by un-checking Expand/Collapsing on Single Member. This changes the behavior to symmetric.

Next, we'll discuss two EPM functions that offer the ability to customize reports in SAP BPC: the `EPMDimensionOverride` and `EPMAxisOverride` functions.

EPMDimensionOverride

The `EPMDimensionOverride` function provides the parameters to override a member selection for a specified dimension in one or more reports. You pass the report ID, dimension, and members to this function. The dimension member selection

for the dimension in the report is overridden with values specified in the member parameter of this function.

In Figure 5.29, you see a report with the NET INCOME data for the Account dimension displayed in the row and the PLAN data for the Category dimension displayed in the column. If you want to use a different selection for the Account dimension in your report, you will use the `EPMDimensionOverride` function.

1. Select a cell in your Excel worksheet, and enter the "EPMDimensionOverride" function.
2. Specify the REPORTID, DIMENSION, and MEMBERS parameters (Figure 5.30, ❶). The REPORTID is usually "000" if there is only one report in the worksheet. The REPORTID parameter for a report can be obtained by using the `EPMReportID` function. It is also displayed in the EPM pane for that report. The DIMENSION parameter is the name of the dimension in the row axis. The MEMBERS parameter is the name of the member you want to use to override the value in the current selection. This can be hard-coded to a value, or it can reference a value in another cell. In our example, we have hard-coded this value to display the base of all member under the G5_OPEX node in the RB_ACCOUNT dimension.

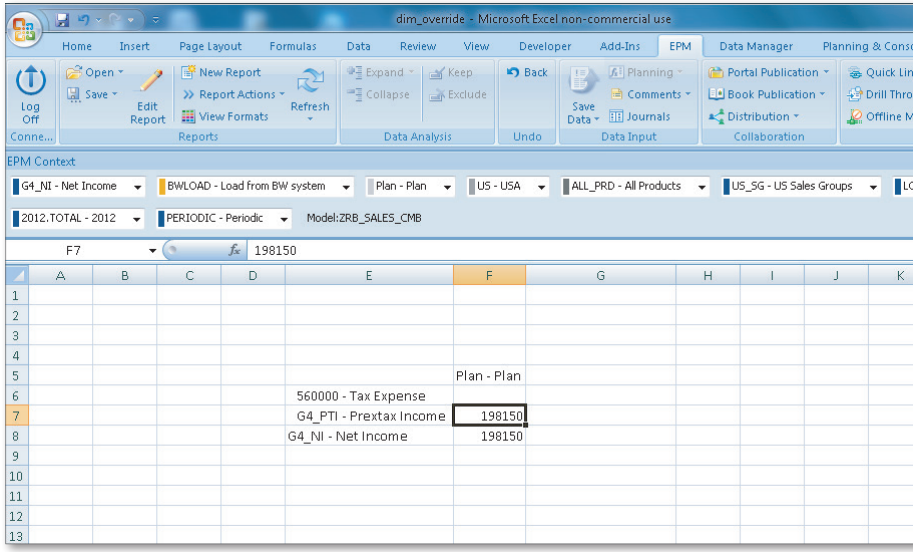


Figure 5.29 Using the EPM Dimension Override Function—Part A

3. When you refresh the report, you see the changes to the report based on the selections in the `EPMDimensionOverride` function (Figure 5.30, ❷).

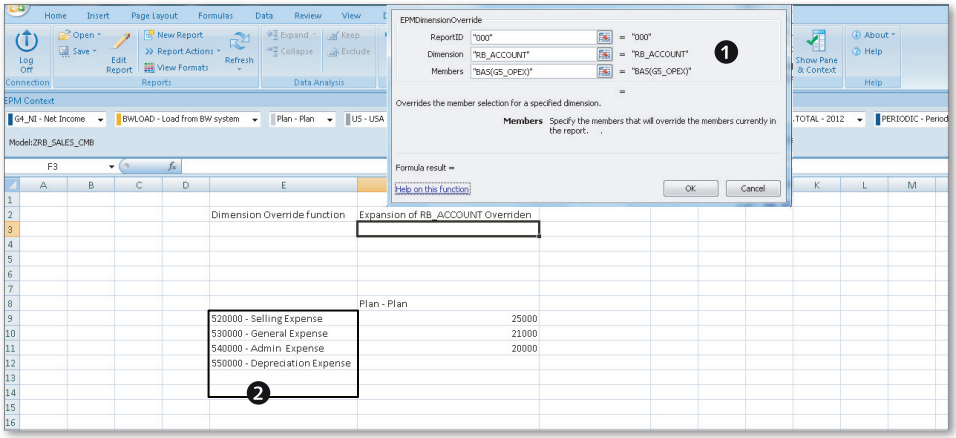


Figure 5.30 Using the EPM Dimension Override Function—Part B

EPMAxisOverride

The `EPMAxisOverride` function provides the parameters to override one or several dimensions with another dimension in the row axis or column axis. The dimensions in the row axis or column axis are replaced with the value passed from the parameters when the REFRESH command is clicked.

Figure 5.30 displays a report with the account data in the rows and category data in the column. Let's say that you want the entity data to be displayed in the rows instead of the account data. In that case, you will use the `EPMAxisOverride` function. The `EPMAxisOverride` includes five parameters: REPORTID, ROW AXIS, DIMENSION1, MEMBERS1, and DIMENSION2 (Figure 5.31, ❶). Specify the REPORTID, and then specify whether this function applies to the row axis in the ROW AXIS box. If it applies to the row axis, you pass the parameter as TRUE. If it applies to the column axis, you pass the parameter as FALSE.

1. Select a cell in your Excel worksheet, and enter the "EPMAxisOverride" function.
2. Specify the REPORTID parameter for the function.
3. Specify the ROWAXIS parameter for the function. Set this value to TRUE if the dimension specified in the DIMENSION1 parameter is to be displayed in the row axis.
4. Specify the name of the dimension you want to display in the DIMENSION1 parameter.

5. Specify the member selections for the dimension in the MEMBER1 parameter. More than one dimension can be selected to display in the rows and columns. You can specify additional dimension and member selections in the function as necessary.
6. When you refresh the report, you see the changes to the report based on the selections in the `EPMAxisOverride` function (Figure 5.31, ❷).

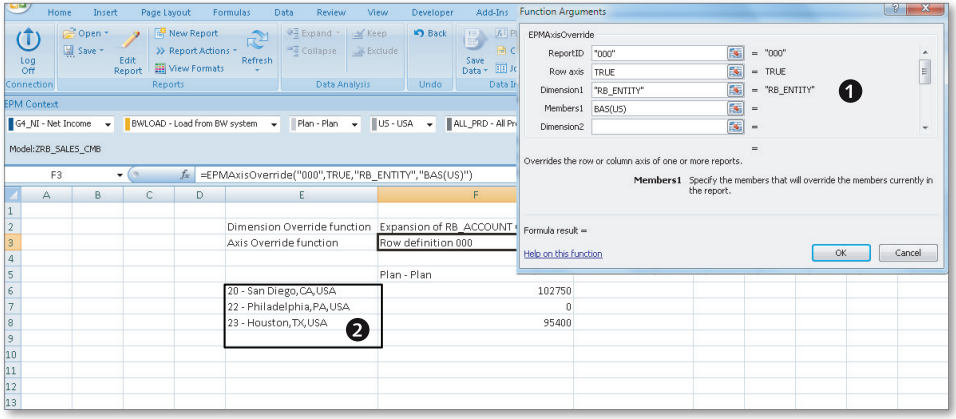


Figure 5.31 Using the EPM Axis Override Function

`EPMDimensionOverride` and `EPMAxisOverride` are powerful functions that can be used to provide enhanced reporting presentation to users. In the preceding examples, we hard-coded the values of dimension and dimension member used in the functions. Instead of hard-coding the values, you can reference cells containing the values of dimension and dimension member used in the functions. You can then present dropdown boxes to a user to choose the dimension and dimension member, populate the values chosen in the dropdown box by the user in the referenced cells, and provide a button in the worksheet that will execute the REFRESH command when clicked. When the user clicks on the button, the report will change based on the parameters passed to `EPMDimensionOverride` or `EPMAxisOverride` functions.

1. Insert a dropdown box for the user to choose the dimension. Using FORMAT CONTROL • CELL LINK, populate a specific cell—for example, M1—with the dimension chosen by the user.
2. Insert a dropdown box for the user to choose the dimension member. Using FORMAT CONTROL • CELL LINK, populate a specific cell—for example, M2—with the dimension member chosen by the user.

3. Insert EPM function `EPMAxisOverride` in a cell in your worksheet. Reference the dimension and dimension member parameters of the function with the values in cell M1 and M2. Instead of selecting just the dimension member chosen by the user, you can also create an Excel formula such as = "BAS("&M2&")" in another cell and reference this cell in the dimension member parameter.
4. After the user selects a dimension and the values to use for that dimension, he will be presented with a custom action button to refresh the report. In anticipation of that step, you should configure that button to execute the REFRESH command. The report will be changed based on parameters passed to the `EPMAxisOverride` function.

5.1.8 Quick Links

Replacing the `EvHOT` function that existed in the prior versions of SAP BPC, *Quick Links* provide the ability to pass member values from one report to another. Using this feature, you can easily jump from one report to another using the context data in the source report as the parameter for the target report. Three types of Quick Links can be created inside an SAP BPC report:

- **Data Link**
  The linked workbook inherits the definition of the selected data cell. The member dimensions (in the row and column) intersecting the data cell are also inherited. The members from the page axis and CONTEXT pane are inherited for the other dimensions that are *not* in the row and column dimension.
- **Member Link**
  The linked workbook inherits the definition of the selected member cell. The members from the page axis and CONTEXT pane are inherited for the other dimensions that are *not* in the selected member cell.
- **Report Link**
  The linked workbook inherits the definition of the selected report in the page axis and CONTEXT pane.

There is precedence in the order of priority when different types of links are created for a report. Data Links take precedence over Member Links, just as Member Links take precedence over Report Links.

You can store the workbook you want to open using the Quick Link on the SAP BPC server or the user's desktop.

5.1.9 Migrating 7.5 Version EvDRE Reports

With the introduction of the new EPM add-in interface for developing reports in the 10.0 version, the former SAP BPC 7.5 `EvDRE` functions continue to be recognized, except for a few functions. You can open an existing report developed in SAP BPC 7.5 in the new tool, but you won't be able to build a new report in SAP BPC 10.0 using the `EvDRE` function. The custom menu functionality is no longer supported in SAP EPM 10.0. The `EvMNU` function is recognized but has to be modified to support the new application programming interface (API) in the new version.

A special tool is available in the EPM 10.0 add-in interface for Excel to migrate reports developed in SAP BPC version 7.5 to the new version. This `EvDRE` migration tool is available as a menu option to migrate SAP BPC `EvDRE` reports to the new EPM version. You have four options for migrating the reports:

- Migrate the active worksheet
- Migrate the active workbook
- Migrate an entire folder of workbooks stored on the SAP BPC server
- Migrate a local report in the user's desktop

These options are constructed so that the user can be satisfied that the migration is complete and successful before saving the `EvDRE` report as an EPM report. If you are satisfied with the migration, you can close the old report without saving it.

Let's briefly cover several key features of the migration process:

- Sort range is not migrated, and neither are ranking, suppressing, and inserting options.
- If the report includes custom SAP BPC macros based on the old API, they have to be manually modified to accommodate the new API introduced in SAP BPC 10.0.
- A log is generated after each migration that can be accessed using the Log menu option under the More command group in the EPM add-in ribbon.

5.1.10 Developing Reports Using the New API for EPM

With the advent of the SAP BPC EPM add-in version 10.0, a new and improved API has been introduced to use inside the Visual Basic application interface in Excel. The EPM add-in comes with an extensive list of APIs you can use in the reports

and input forms developed for the EPM environment. In addition, out-of-the-box functions are provided so you can customize your reporting solution in SAP BPC.

Using Visual Basic Macros in Reports

You can call the APIs as custom buttons in your reporting interface. When you click the button, the API specified in the code will be executed. You can also develop code in VBA and assign the code to specific out-of-the-box functions such as AFTER_REFRESH, AFTER_CONTEXTCHANGE, and so on. If code is added under the AFTER_REFRESH function, then the underlying code is executed after a user refreshes the data in a report or form. Table 5.4 lists the functions that can be used inside a Visual Basic macro.

Function
BEFORE_CONTEXTCHANGE
AFTER_CONTEXTCHANGE
BEFORE_REFRESH
AFTER_REFRESH
BEFORE_SAVE
AFTER_SAVE
BEFORE_EXPAND
AFTER_EXPAND

Table 5.4 Functions Usable Inside a VB Macro

The name of the function tells you when the function will be executed; for example, the AFTER_CONTEXTCHANGE function is executed after a user selects a member in the EPM Context pane.

Consider this example of code inside a custom button to call the EPM API to refresh the data in the current report:

```
Dim client As New EPMAddInAutomation
Sub Button5_Click()
client.Refresh
End Sub
```

You should now have an idea of how to use EPM APIs and functions to customize your reports. Reports can also be developed using the SAP BPC Web Client, which we'll cover in the next section.

5.1.11 Developing Reports Using the SAP BPC Web Client

From the SAP BPC Web Client, you can develop reports using the drag-and-drop feature. This provides you with an easy-to-use interface for developing reports, which allows users to start developing reports with little training. These types of reports are used less frequently than Excel-based SAP BPC reports, but they can be used by users who prefer to report on the web. To build a web report, follow these steps:

1. Click on the Report link in the Start Page view from the Home workspace of the Web Client.
2. Click on the Manage Context icon in the upper left to set the model for which you want to create a report (Figure 5.32). This opens a dialog box titled Change Model. Select the desired model from the dropdown list.

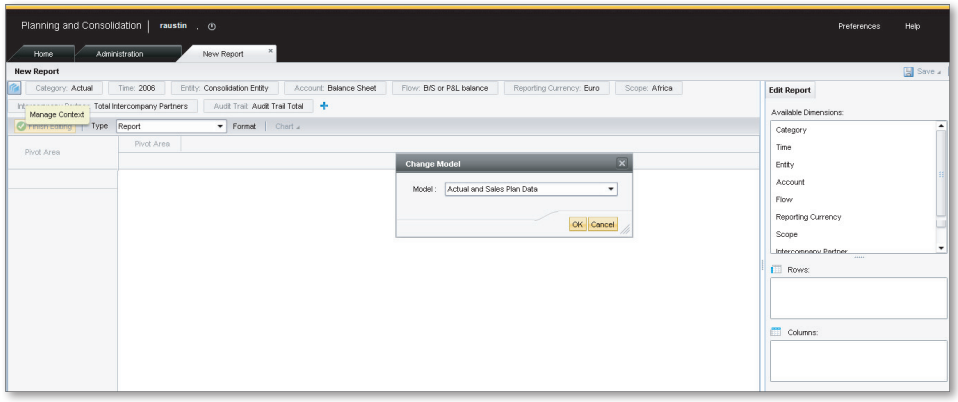


Figure 5.32 Creating a Web Report—Part A

3. Click on the plus icon to display the dimensions in the scope context for selecting the dimension member values (Figure 5.33). The default setting of what to display here is determined by the settings made in the Administration workspace under the Context Default view of the Rules Domain. For every model, you can define the default member displayed for a dimension, specify if a dimension is to be displayed, specify if a dimension name or dimension ID is

to be displayed, and determine whether the member ID or description (or both) are to be displayed when creating or maintaining a web report. You can add all of the dimensions in a single click by clicking on the ADD ALL DIMENSIONS link.

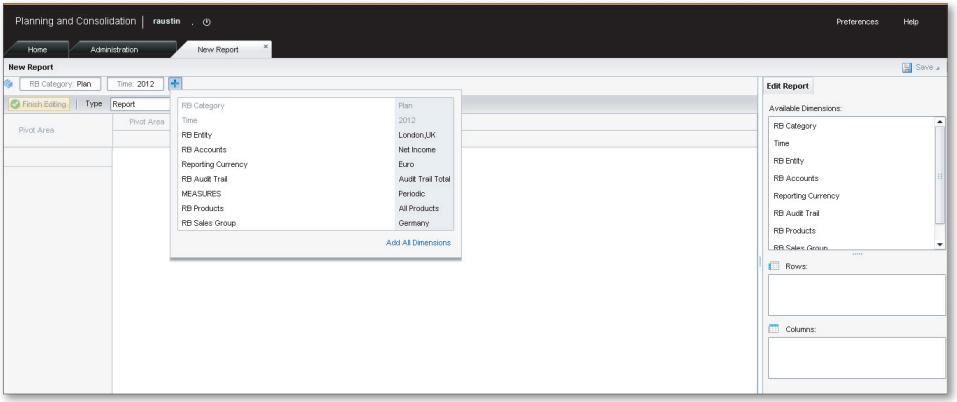


Figure 5.33 Creating a Web Report—Part B

4. Drag the dimensions required to the rows and columns areas. We have selected the RB_ACCOUNTS dimension for the rows (Figure 5.34, ❶). Select 2012.TOTAL in the scope context for the Time dimension (Figure 5.34, ❷), and then drag the TIME dimension to the columns.
5. Highlight the dimension in the row, and click on SELECT MEMBERS to select the members for that dimension (Figure 5.34, ❸).

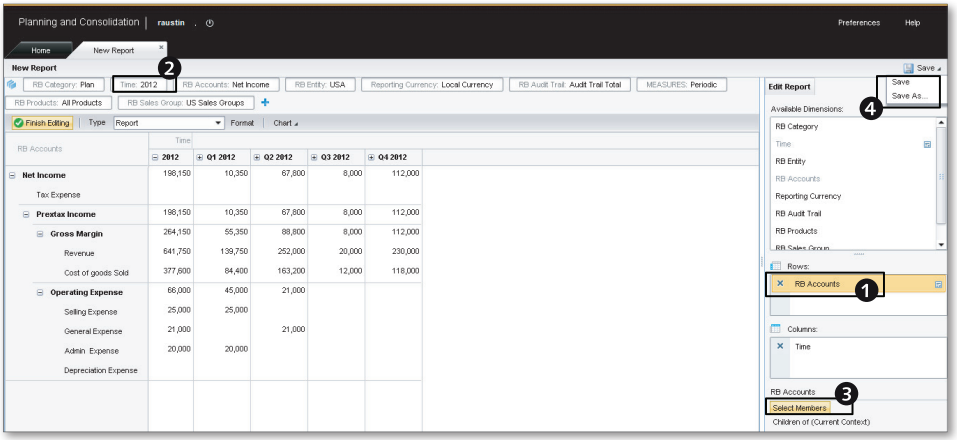


Figure 5.34 Creating a Web Report—Part C

6. This opens the new dialog box Criteria Editor to select the members and relationships (Figure 5.35). Similarly, select the members and relationships for the Time dimension that is displayed in the column.

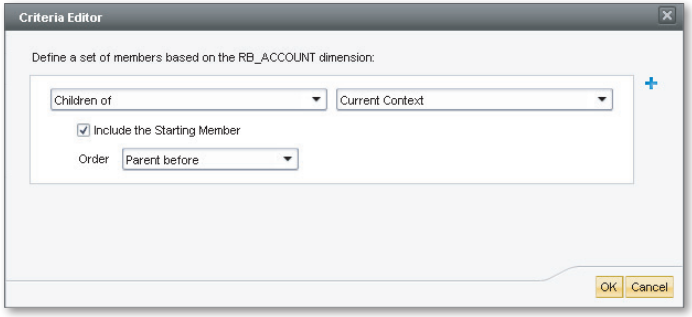


Figure 5.35 Creating a Web Report—Part D

7. Click on Finish Editing to confirm the report settings (Figure 5.36). Figure 5.37 shows the report displayed along with the menu options to add comments, work status, and edit the report.

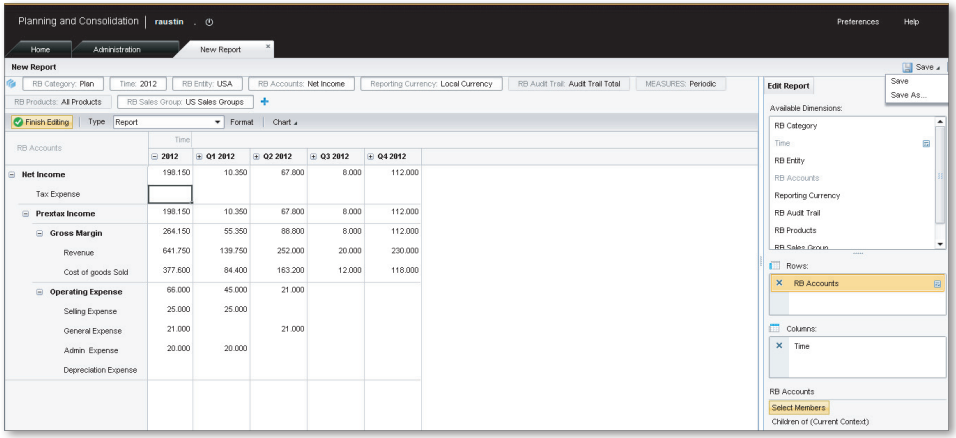


Figure 5.36 Creating a Web Report—Part E

8. When you are ready to save the report, click on Save (refer back to Figure 5.34, ❹).

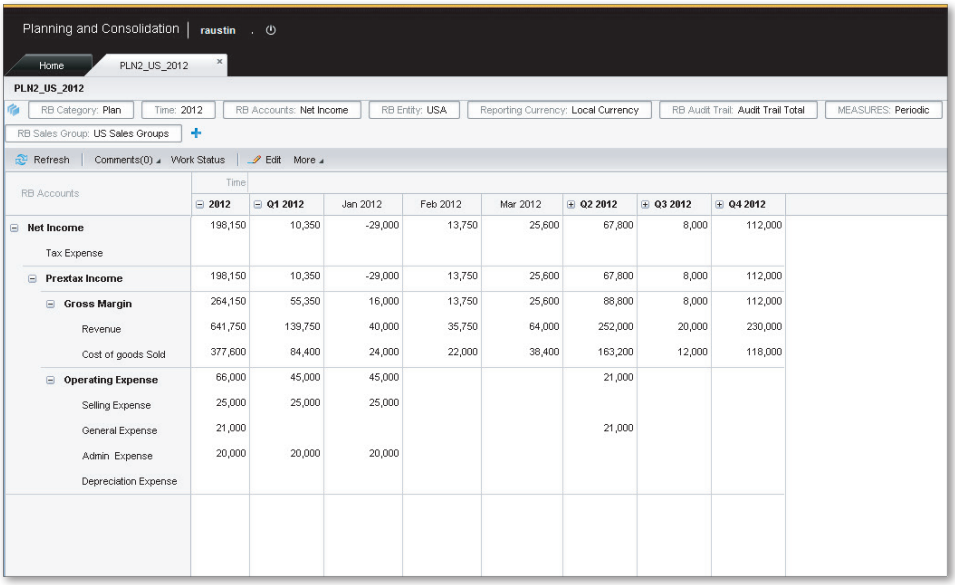


Figure 5.37 Creating a Web Report—Part F

You've now seen how you can develop a report using the SAP BPC Web Client.

Until now, you have seen the features of the EPM add-in reporting tool for Excel and learned how to create a report and use standard templates. We discussed the usage of EPM functions to develop sophisticated reports. You also learned how to develop reports from the Web Client. In the next section, we'll explain how to use input forms to enter and modify data in an application.

5.2 Planning in SAP BPC Using Input Forms

An input form is used for entering and modifying data inside a model. For input forms, data entered in the report can be validated and updated in the database.

The following are some of the uses and features of input forms:

- **Entry of data at the base member level**
  When you use input forms, the dimension member to be updated cannot be a hierarchy member. The rows in the input form should be base members. For example, if the revenue account is a hierarchy member and is a parent of several

base members, it cannot be directly used in the input form. The data needs to be entered only for the base members that make up the hierarchy member.

- **Concurrency lock**
  When data on an input form is sent for updating, the locks on the data are checked. These locks are called *concurrency locks* and are designed to prevent two users from updating the same set of data at the same time. If two users enter data for the same selection and try to send the data to the database at the same time, only the user who obtained the lock first will be able to update the data. The user who sends the request for a lock later will get an error message stating that another user has already obtained the lock and that the update cannot happen at this time.

- **Work status**
  Work status can be configured to prevent changes from being made to data; for example, the plan data for 2012 can be set to the locked status so that no further changes can be made to it. When a user attempts to modify data for a region that is locked, he receives a message indicating that the data cannot be modified due to work status locks. We'll discuss work status in detail in Chapter 7.

- **Validation of data**
  SAP BPC for NetWeaver supports creating validations in the SAP NetWeaver BW system. For example, a validation can be set up to ensure that transaction data entered for an entity located in the United States accepts U.S. dollars as the currency. If data for an entity located in the United States is entered with the Euro as the currency, the validation test fails, and the data is rejected. Transaction UJ_VALIDATION is available for configuring validations in the SAP NetWeaver BW system. This is a new functionality that is available only in SAP BPC for NetWeaver.

  ABAP can be used to create custom logic to enforce validations when you use this transaction. The validation mechanism can be used to configure validation rules that are checked before data is entered or updated. We'll discuss how to use and set up validations in Chapter 6.

5.2.1 Standard Templates for Input Forms

Similar to the standard templates used for reporting, standard templates can be used for entering data. Table 5.5 shows a list of standard templates used for input forms.

Template Name	Description
Trend by Account	The Trend by Account input template is used for displaying and modifying the current trend of data based on the Time dimension. The account members are displayed in the rows, and time data is displayed in the columns.
Consolidating	The Consolidating input template contains account members in the rows and entity members in the columns.
Trend by Entity	The Entity Trend input template is set up similarly to the Account Trend template. The only difference is that the Entity Trend input schedule contains Entity members in the rows, and the Account Trend input schedule contains account members in the rows.
Input Form Nested Row	The Nested Row input template is set up so that more than one dimension can be specified in the rows.
Input Form Comparative	The Comparative input template is defined to display account members in the rows and category members in the columns. The input form also displays the variance between the categories displayed in the column.

Table 5.5 Standard Templates for Input Form

5.2.2 Developing an Input Form

Now let's discuss how to create an input form to manually enter and modify data. The input form we'll create as an example will have ACCOUNT in the row and TIME along the columns. The creation of an input form is similar to the creation of a report. Follow these steps:

1. Create a report in the EPM add-in interface.
2. Select the ACCOUNT dimension in the row and the TIME dimension in the column. In the EPM – MEMBER SELECTOR dialog box, select NET INCOME in the DIMENSION MEMBERS list as the value for the ACCOUNT dimension, and select BASE LEVEL under SELECTION RELATIONSHIP. This will select only base members under the NET INCOME account (Figure 5.38). In the EPM – MEMBER SELECTOR dialog box, select 2012.TOTAL as the value for Time dimension and BASE LEVEL MEMBERS in SELECTION RELATIONSHIP (Figure 5.39).

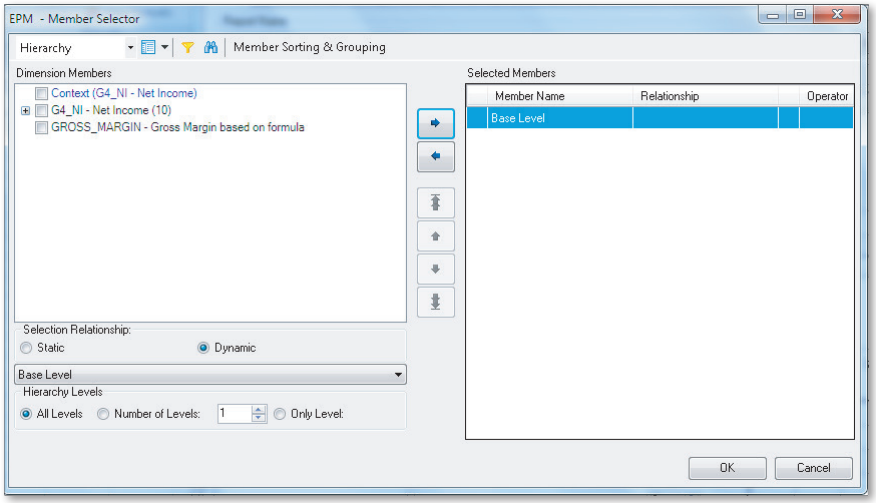


Figure 5.38 Creating an Input Form—Part A

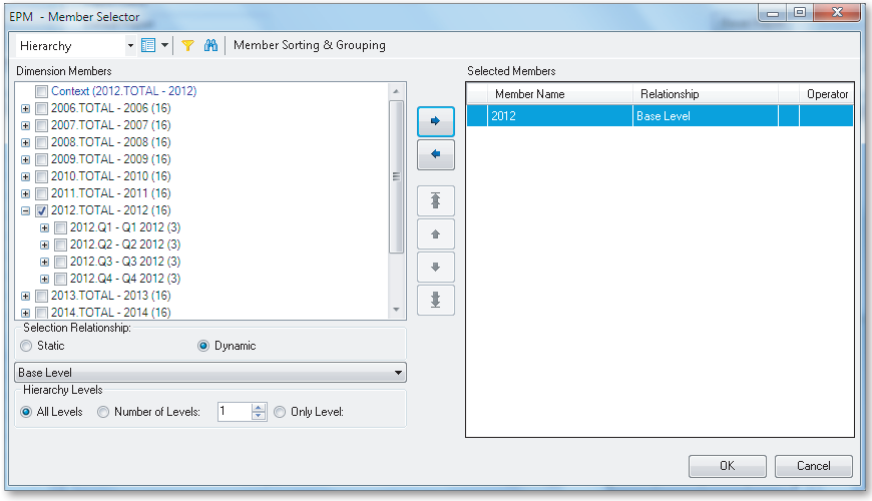


Figure 5.39 Creating an Input Form—Part B

3. In the EPM – Sheet Options dialog box, select Use as Input Form under Data Input (Figure 5.40, ❶). This will allow the user to enter, modify, and save the data.

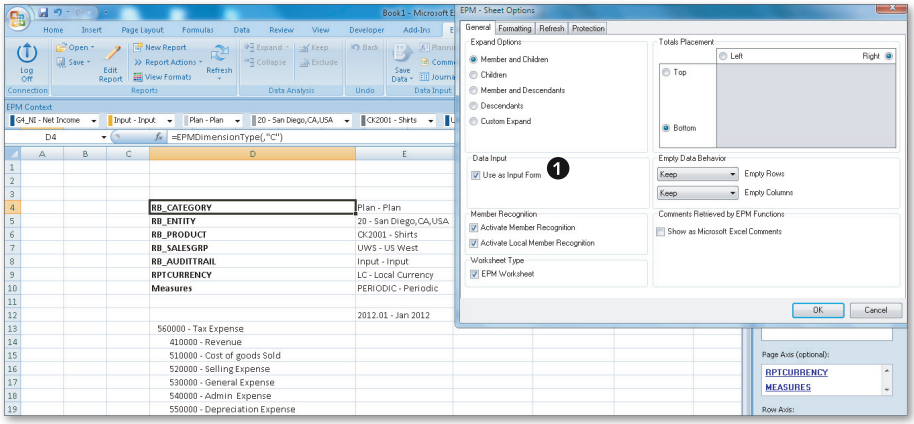


Figure 5.40 Creating an Input Form—Part C

4. Click on the menu option to save the data under the SAVE command. Enter "200" as the amount for JAN 2012. Then click on SAVE DATA, and choose the menu option to save the data (Figure 5.41, ❷). The EPM – SUBMIT DATA dialog box prompts you with the number of records that will be sent and requests a confirmation to continue (Figure 5.41, ❸).

5. Ensure that you have selected base members for all other dimensions in the EPM CONTEXT pane. These values will be used to update data in the model.

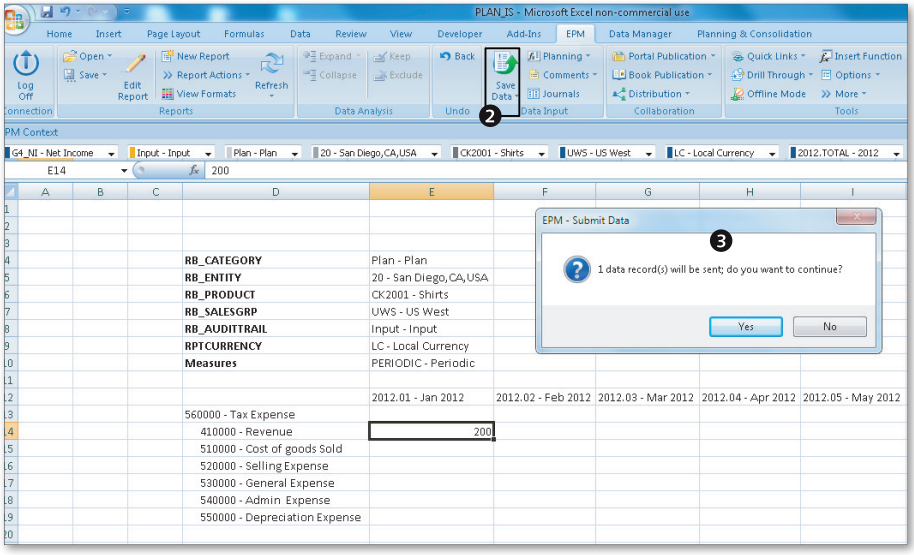


Figure 5.41 Creating an Input Form—Part D

6. After you confirm the action to continue, a message log displays the details of the records updated in the database (Figure 5.42).

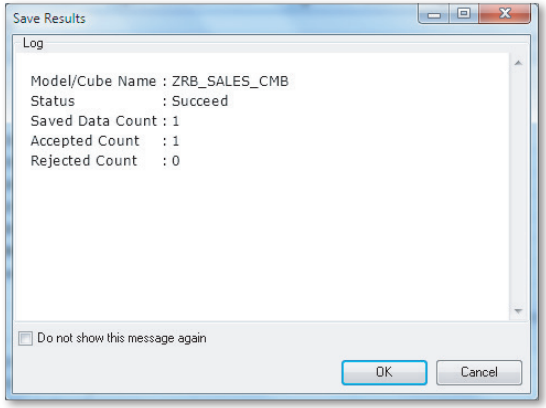


Figure 5.42 Creating an Input Form—Part E

You now know how to create an input form. Next, we'll move on to how to use the `EPMSaveData` function to update data in a model.

5.2.3 Using the EPMSaveData Function to Save Data

In the previous section, we discussed the development of a standard EPM input form. In this section, we will use the `EPMSaveData` function to send data from an input form to the database. You use the `EPMSaveData` function to save data entered in a single cell in Excel based on the selection of dimension members. The `EPMSaveData` function syntax includes the connection, the values of dimension members, and the reference cell that contains the value to be updated.

The syntax for using the `EPMSaveData` function is as follows:

```
EPMSaveData(<ValueReference>,<Connection name>,[dim:]member1,[dim:]
member2,...[dim:]memberN)
```

`ValueReference` is the reference cell that contains the signed data.

You can send the data only for a base member. If a member value is not specified in the function, then the value for that dimension is taken from the EPM Context pane.

The `EPMRetrieveData` and `EPMSaveData` functions provide a lot of flexibility in reports and input forms to retrieve and send data. They also provide the ability to

bring data from different models and different environments in the same report. However, they are inefficient in the way they are run because each function costs a call to the database when the data is refreshed or sent. Please exercise caution when using these functions.

5.3 Other Features

Let's briefly consider some additional features available for reporting using the EPM add-in tool.

5.3.1 Drill-Through Feature

The drill-through feature allows SAP BPC users to directly access information in a source system via a URL. For users who have used the report-to-report interface (RRI) in SAP NetWeaver BW, the drill-through features in SAP BPC will be familiar.

There are two types of drill-through in SAP BPC 10.0: drill-through to table (available only for the Microsoft version) and drill-through to URL.

The configuration for drill-through to URL is defined in the Drill-Through view under the Features Domain inside the Administration workspace in the Web Client for the SAP NetWeaver version. You identify the ID, Description, Target URL, and Model from where the drill through can be launched. In addition, you can specify the parameters you want to pass from your report to the URL.

After a drill-through is defined for a model, it can be accessed from the EPM add-in interface in Excel for a connection that is associated to that model. Select a cell in your report, and click on the Drill-Through command to access the URL.

5.3.2 Integration with SAP Business Objects Dashboards

SAP BusinessObjects Dashboards, formerly called Xcelsius, is an SAP BusinessObjects tool used for creating dashboard applications. SAP BPC 10.0 is integrated with Dashboards and offers two important benefits. First, the data in an SAP BPC model can be used to create dashboards and visually compelling reports. Second, the write-back feature from Dashboards to an SAP BPC model allows data input in Dashboards to be updated to SAP BPC using this front-end tool.

5.4 Summary

In this chapter, we explained the steps for reporting, planning, and analyzing data in SAP BPC. We discussed examples of how to use dynamic templates and create a custom report using EPM functions, and we explained how to create and deploy reports on the web. We then went on to create an input form, and explained how to use it for entering and modifying data in a model. Finally, we saw how the drill-through feature can be used inside SAP BPC and how Dashboards can integrate data in SAP BPC for dashboard reporting and data updating.

In Chapter 6, we'll introduce different types of business logic that can be configured in SAP BPC and see how it automates several tasks related to planning and consolidation. You'll learn how to configure business rules, logic script, and validation.

Dimension formulas, logic script, business rules, and BAdIs can be used to define business logic and perform calculations in planning and consolidation models.

6 Developing Business Logic in SAP BPC

Every organization has rules and policies that guide its business logic and must be incorporated into IT applications; this is no different for planning and consolidation models. In this chapter, we'll discuss the options available in SAP Business Planning and Consolidation (SAP BPC) for defining logic and automating the process of deriving data and applying business rules.

Section 6.1 covers using dimension formulas to perform calculations in SAP BPC models.

In Section 6.2, we'll discuss the usage of code-based logic—also called logic script—to meet complex business requirements in SAP BPC, and you'll learn how to use it to perform functions related to data management.

In Section 6.3, we'll discuss business rules and their usage in SAP BPC. We'll briefly look at how they are used to perform certain common tasks related to planning and consolidation. (We'll also discuss business rules in detail in Chapter 9, when we discuss the process of consolidating financial data for an organization.)

In Section 6.4, we'll explain how to perform currency translation, which is used in the context of planning and consolidation. Currency translation is a requirement for an organization that does business in more than one country or has transactions in different currencies.

In Section 6.5, we'll discuss allocation. Organizations commonly use allocation to apportion expenses incurred by a common entity, such as a corporate entity, to other units based on certain drivers.

In Section 6.6, we'll review the usage of BAdIs to define business logic in SAP BPC. BAdIs are coded using the ABAP language, which provides additional flexibility

when there is a need to write custom code. It also helps to leverage ABAP expertise in the organization when implementing SAP BPC models.

In Section 6.7, we'll discuss validation of data in SAP BPC, explaining how to use driver dimensions to define validation rules and enforce data integrity. This validation rule is used to validate combination of value of dimensions.

In Section 6.8, we'll discuss a new rule type called *controls* that has been introduced in the SAP BPC 10.0 version for NetWeaver. This replaces the validation business rule that existed in the SAP BPC 7.x versions. The controls business rule is used to ensure consistency of data in SAP BPC model (for example, that assets is the sum of liabilities plus an owner's equity).

Let's start this chapter by discussing how to use dimension formulas to automatically calculate values in SAP BPC models.

6.1 Dimension Logic

In the past few chapters, we discussed how to use dimensions in SAP BPC. We went through the process of defining a dimension and loading data into it, as well as how to integrate the dimension into a model and use it in reporting. In Chapter 3, we explained that a dimension can be enabled to include a formula, which means that it can aid in the automatic calculation of data. This process is referred to as *dimension logic*.

6.1.1 Dimension Formulas

To use dimension formulas, you must enable the dimension to contain formulas. This setting is available when editing the structure of the dimension in the Web Client. After this option is enabled, dimension formulas can be defined for a dimension. For example, in the Account dimension, there could be a Gross Margin dimension member that is defined based on a formula to calculate the gross margin. This formula can be then used to display the gross margin in reports.

The following rules apply to the formula property of a dimension:

- When defining formulas, you can use multi-dimension expressions (MDXs), which are especially useful for complex formulas. The list of available MDXs can

be displayed by executing the BAPI_MDPROVIDER_GET_FUNCTIONS function module using Transaction SE37 in the SAP NetWeaver Business Warehouse (BW) system. The ENVIRONMENTSHELL comes with a list of standard MDX formulas, available in the file *Mdxlib.lgf*. When you reference MDX functions in a dimension member formula, specify the name of the library file that contains the MDX functions in the options area of the member formula maintenance view. This is the file the system will reference to resolve the MDX functions defined in the formula. You can also create your own MDX functions and place them in the *\ROOT\WEBFOLDERS\<ENV>\SYSTEMLIBRARY\<LOGIC LIBRARY\<MDXFILE_NAME>.LGF* folder using Transaction UJFS in the BW system.

- When a formula is defined for a dimension member, you cannot directly enter any data for that member.
- In the formula, the name of the dimension should be used to refer to dimension members, but the dimension name is not required for the Account dimension. For example, the formula property of dimension member Product 3 can be set as follows:

  ```
  [PRODUCT].[Product1] + [PRODUCT].[ Product2]
  ```

  Note the usage of the technical name of the PRODUCT dimension in the formula.
- The dimension formula for the dimension member applies to all models containing the dimension in the environment.
- SAP BPC supports the use of addition, subtraction, multiplication, and division operators when defining formulas.
- The syntax of the formula is checked when dimension members are activated using the Process dimension option.
- When a formula is defined for a dimension member, the system calculates the result for the dimension member based on the formula at runtime. Including formulas for a dimension member can cause data retrieval to take more time depending on the amount of data on which the formula has to work.

6.1.2 Solve Order

In the context of defining dimension formulas, it is important to understand the *solve order keyword*. In some reports, you may display more than one dimension member that is calculated based on a dimension formula, and the solve order function helps

resolve conflicts between two formulas used in a report. In this situation, it sets the priority for which formula should be used.

For example, let's look at a scenario where you have the Account dimension with members Account 1, Account 2, and Account 3 in an environment. Account 3 is defined as a calculated member based on the following formula:

```
Account3 = [ACCOUNT].[Account1]/[ACCOUNT].[Account2]
```

Assume that the same environment includes another dimension, Product, with the members Product 1, Product 2, and Product 3. The Product dimension is defined as a calculated member based on the following formula:

```
Product 3 = [PRODUCT].[Product1] + [PRODUCT].[Product2]
```

Here, we have two calculated members: Account 3 for the Account dimension and Product 3 for the Product dimension. For these dimension formulas, we can set the solve order function to set which formula should take precedence over the other. A formula that has the solve order defined as a lower value takes priority over others with a higher value; a formula that has the solve order defined as a value of 0 takes the highest priority. SAP BPC evaluates the member with the highest solve order first, and calculates it last. We'll define the formulas to include the solve order function as follows:

```
Account3 = [ACCOUNT].[Account1]/[ACCOUNT].[Account2];SOLVE_ORDER=5
Product 3 = [PRODUCT].[Product1] + [PRODUCT].[ Product2];SOLVE_ORDER=50
```

Consider a report that contains the accounts in rows and the products in columns. Table 6.1 shows the format of such a report.

	Product 1	Product 2	Product 3
Account 1	120	130	250
Account 2	5	5	10
Account 3	24	26	50

Table 6.1 Demonstration of Solve Order

Now look at the value of the cell that intersects Account 3 and Product 3. The result is 24 + 26 = 50, not 250/10 = 25. This is due to the fact that the Account formula

takes precedence over the Product formula because the value of the solve order for the Account formula is less than that of the Product formula. Because the Product formula has the highest solve order, it is calculated last.

So let's create a Gross Margin dimension member in the Account dimension. We'll calculate the gross margin by subtracting the cost of goods sold (COGS) from the revenue. Recall from Chapter 3 that we set the gross margin as a hierarchy node of total revenue and total cost. In this section, we will demonstrate how to calculate the gross margin using a dimension formula.

We'll now create a dimension formula for the Account dimension in our planning environment. Follow these steps:

1. Log in to the SAP BPC Web Client, and connect to the ZRB_PR_PLAN environment.
2. In the Administration workspace, select the Dimension view of the Dimensions and Models domain in the left pane, highlight the RB_ACCOUNT dimension, and click on the Edit Structure menu option. The structure and the settings for the dimension are displayed. Confirm that the Allow Member Formulas box is checked. If not, check that box, save the changes, and process the dimension.
3. Highlight the RB_ACCOUNT dimension, and click on Edit Members in the Dimension view to create a new member. Create a new dimension member called GROSS_MARGIN. We will define a formula to derive the value of this dimension member a little later. After adding the dimension member, save the changes, and process the dimension.
4. You are now ready to define a formula for the new member GROSS_MARGIN that you created in the previous step. In the Administration workspace, under the Rules domain, select the Member Formulas view. You will see the list of dimensions that are enabled for using formulas. Highlight the RB_ACCOUNT dimension, and click on the Open menu option (Figure 6.1, ❶). This will open the interface to create and maintain formulas for the members in the RB_ACCOUNT dimension. You will now create a formula for the GROSS_MARGIN dimension member. Click on New to open the Edit Formulas dialog box (Figure 6.2, ❷).

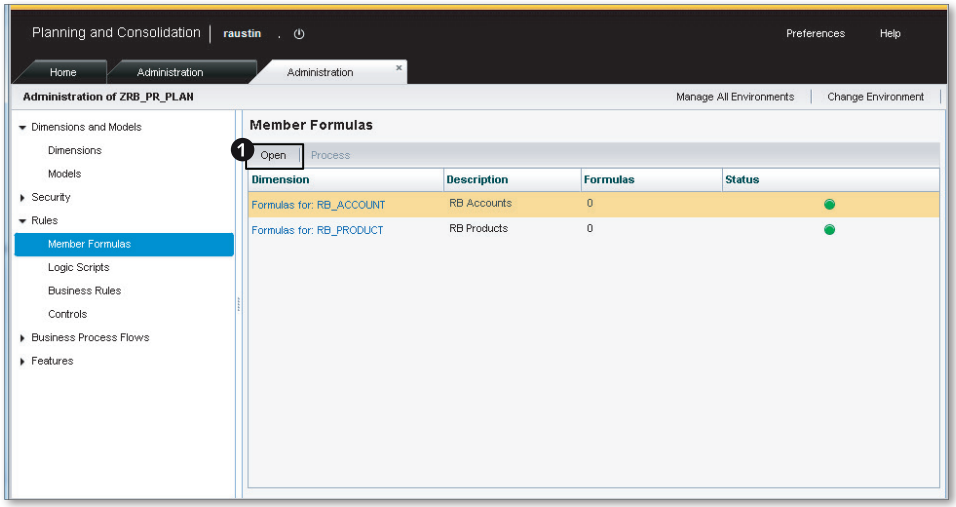


Figure 6.1 Creating a Formula for the Gross Margin Account—Part A

5. Select the dimension member GROSS_MARGIN from the dropdown box for creating the formula (Figure 6.2, ❸). Highlight the revenue account 410000 from the list of dimensions members displayed, and click on INSERT IN FORMULA (Figure 6.2, ❹ and ❺). The member is transferred to the formula area (Figure 6.2, ❻). Then manually enter the "-" sign in the formula area (Figure 6.2, ❻). Now select the COGS account 510000 from the list of members displayed, and click on INSERT IN FORMULA (Figure 6.2, ❹ and ❺). The member is added to the formula area (Figure 6.2, ❻). (You can also manually enter the formula in the formula area.) Save the changes (Figure 6.2, ❼).
6. To use the new formula you created, click on PROCESS (Figure 6.2, ❽). You have defined a formula that will dynamically calculate the value for the GROSS_MARGIN dimension member in your reports.
7. The new dimension member can now be used in reports to display the gross margin data. The model uses the formula defined to display the value for the Gross Margin dimension member (Figure 6.3).

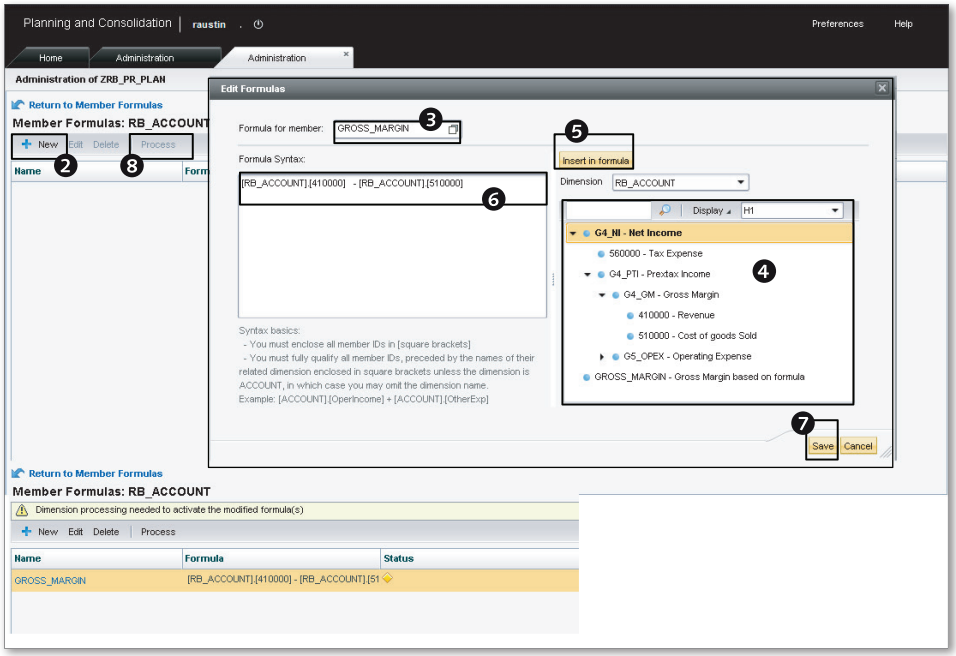


Figure 6.2 Creating a Formula for the Gross Margin Account—Part B

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data										
3	**RB_CATEGORY**	Plan	Plan										
4	**RB_AUDITTRAIL**	BWLOAD	Load from BW system										
5	**RB_ENTITY**	US	USA										
6	**RB_PRODUCT**	ALL_PRD	All Products										
7	**RB_SALESGRP**	US_SG	US Sales Groups										
8	**RPTCURRENCY**	LC	Local Currency										
9													
10													
11		Jan 2012	Feb 2012	Mar 2012	Apr 2012	May 2012	Jun 2012	Jul 2012	Aug 2012	Sep 2012	Oct 2012	Nov 2012	Dec 2012
12	Revenue	40000	35000	64000	92000	140000	20000			20000		150000	80000
13	Cost of goods Sold	24000	22000	38400	55200	96000	12000			12000		70000	48000
14	Gross Margin based on formula	16000	13000	25600	36800	44000	8000			8000		80000	32000
15													
16													

Figure 6.3 Reporting Using Dimension Formula

In this section, we explained how to use dimension formulas to automatically calculate the value of a dimension member in reports. We also discussed the importance of using solve order when defining a formula and how to use it when multiple formulas are in conflict.

Next we'll explain the use of logic script in SAP BPC and see how it can help satisfy detailed business requirements that cannot otherwise be met using a standard SAP BPC configuration.

6.2 Logic Script

Sometimes a business process may be so complex that a standard configuration is not enough to satisfy a requirement, or it may not be feasible to use dimension formulas. In these cases, you can use logic script to meet the requirement.

Consider a business scenario where you're planning sales for a future period. You copied last year's actual sales data as the basis of planning for the current year; however, due to changes in current demand, it is now necessary to adjust revenue based on the realities of the market by, say, accepting a percentage value from the user to either increase or decrease revenue. This is a good candidate for logic script because it is not easy to meet this requirement using the standard SAP BPC model or dimension logic. Logic script, on the other hand, can be used to increase or decrease revenues by a certain percentage for certain selections.

The script language provides full-fledged language support to write custom logic to meet a specific business requirement. The language constructs are similar to 4GL and can be easily included in an SAP BPC model. They are powerful and efficient, especially when a lot of data needs to be processed.

A script file is used to create the code and is stored with the extension .lgf. We'll now look at some of the common instructions used in logic script.

6.2.1 Logic Script Construct

In this subsection, we'll discuss some of the commands used when developing logic script.

Pass Specific Dimension Members

It may be necessary to pass specific members to be used for selection in the logic script. You can achieve this using the `*XDIM_MEMBERSET` command, which is used to scope data records that need to be processed by the script. When executing a logic

script, the selections in EPM context pane does not matter. The system uses the *XDIM_MEMBERSET command to scope data that needs to be processed by the script.

This will be helpful if you want to use only a specific member list in your logic script; you can specify more than one member when using this statement:

```
*XDIM_MEMBERSET {dimension name} = {member set}
```

The following instruction will use the member set ACTUAL,BUDGET for the dimension category when processing the script file:

```
*XDIM_MEMBERSET CATEGORY = ACTUAL,BUDGET
```

You can also use the *XDIM_MEMBERSET instruction with the not equal to (<>) operator, as follows:

```
*XDIM_MEMBERSET CURRENCY<>USD
```

If you want to read all of the members of a dimension, you can use the keyword <ALL>, as follows:

```
*XDIM_MEMBERSET INTCO = <ALL>
```

Adding Members to Existing Member Sets

A *member set* is a set of dimension member values that is selected for processing from a model in a script. You can add members to an existing member set using the following instruction:

```
*XDIM_ADDMEMBERSET {dimension} = {member set}
```

This will add the members specified in the instruction to the list of existing selections:

```
*XDIM_ADDMEMBERSET CURRENCY = LC
```

For example, if the user has selected the USD and EUR currency dimension members, the preceding instruction will also add the local currency (LC) to the list of dimension members. This is helpful when you want to make sure a dimension member is always included in the list of members for that dimension.

Threshold for Maximum Number of Dimension Members Processed for a Dimension

In some cases, when the number of dimension members selected for processing is too high, the execution performance of the logic script may be impacted. In these

cases, it may be necessary to break the queries into more than one to improve performance. The following instruction can be used to specify the threshold for the maximum number of dimension members before the query has to be split:

```
*XDIM_MAXMEMBERS {dimension} = {max number of members}
```

In the following instruction, if the number of entities to be processed exceeds the limit of 20 members, the query processor will break the query into multiple queries of no more than 20 entities each:

```
*XDIM_MAXMEMBERS Entity = 20
```

Parallelize Script Execution

The new keyword `*XDIM_PACKAGEBY` lets you parallelize the script execution. This powerful feature improves the performance of logic script. The SAP NetWeaver BW system should be enabled for parallel processing to leverage this feature. In addition, the SAP BPC global parameter `PARALLEL_SERVER_GRP` has to be set in Transaction SPRO to alert the server group that should be used for parallel processing. The following statement is used in the logic script when using this keyword:

```
*XDIM_PACKAGEBY {dimension}
```

Another option is available to parallelize script execution: the `RUNLOGIC` keyword. In fact, a how-to guide is available in the SAP SDN that describes the process to execute a BAdI via logic script using the `RUNLOGIC` keyword. This how-to guide was originally developed for the SAP BPC 7.5 for NetWeaver, but a new version of the how-to guide is being developed for SAP BPC 10.0 for NetWeaver. Because the document is in the form of a how-to guide, it is not officially supported by SAP.

The URL to the how-to guide is *http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/b0ad38cf-9e0e-2e10-9d9a-fbf57e69cd40*.

In addition to offering the advantage of parallel execution, this BAdI provides the ability to call the logic script that exists in a model different from where it is called from.

If you are using this keyword in the logic script, use Transaction SPRO in the SAP NetWeaver BW system, and confirm that the SAP BPC parameter `PARALLELSCRIPT` is set to specify the number of parallel processes to use in the environment where you are using this BAdI.

Storing and Retrieving Variable Values

You can build a dynamic set of members and store it in a variable using the following instruction, which enables the values of a particular dimension member ID or property value to be stored in a variable. The values stored in the variable can be used later in the logic.

```
*SELECT ({variable}, {what}, {from}, {where}
```

The following example shows how to store data in a variable using the `*SELECT` instruction:

```
*SELECT(%REPORTING_CURRENCIES%, ID, CURRENCY, [GROUP] = 'REP')
```

In this instruction, the user can retrieve the `ID` of all members in the `CURRENCY` dimension, where the `GROUP` property has the value `REP`. The `<variable>` is the `%REPORTING_CURRENCIES%`, the `<what>` is the `ID` that will be stored in the variable, the `<from>` represents the dimension to use, and the `[GROUP] (<where>)` represents the property to pull the data from the dimension specified in the `<from>`.

The values stored in the variable using the instruction can be used to set the member values for a dimension, as follows:

```
*XDIM_MEMBER_SET CURRENCY=%REPORTING_CURRENCIES%
```

In this case, `%REPORTING_CURRENCIES%` is the variable containing the values.

Members Passed to a Dimension

You can use the `%SET%` keyword to determine the members that are passed for a dimension. Add additional members using the following instruction:

```
*XDIM_MEMBERSET ENTITY = %ENTITY_SET%, SPECIAL_ENTITY
```

In this example, the `SPECIAL_ENTITY` entity is added to the Entity dimension list.

Looping—WHEN/ENDWHEN Statement

You use the `WHEN/ENDWHEN` structure when it is necessary to update the current record or generate new records. The syntax of this structure is as follows:

```
*WHEN {criteria}
*IS {valid condition1}[,{valid condition2},…]
*REC[([FACTOR|EXPRESSION={Expression}[,{dim1}={member},{dim2}=…])] [*RE
C[([FACTOR|EXPRESSION={Expression}[,{dim1}={member},{dim2}=…])]] ….
```

```
[*ELSE] … …
*ENDWHEN
```

An example of usage is as follows:

```
*WHEN ACCOUNT.RATETYPE
*IS "AVG", "END"
*REC(FACTOR=-1, ENTITY=INTCO.ENTITY)
*ENDWHEN
```

In this example, the logic script reads all records that have the `RATETYPE` property equal to `AVG` or `END` and generates a new record that has a value equal to the current value, times the `FACTOR`. In addition, the value of the `ENTITY` dimension in the new record is set to the `ENTITY` property value associated with the `INTCO` dimension member of the current record. Values of all other dimensions are the same as that of the current record.

When there is no criterion to test, you can use the following instruction:

```
*WHEN *
*IS *
*REC(…)
*ENDWHEN
```

Control What Data Is Selected in WHEN/ENDWHEN Loops

The `WHEN_REF_DATA` keyword provides two options to control what records are processed in the `WHEN/ENDWHEN` loop:

```
*WHEN_REF_DATA=MASTER_DATA
*WHEN_REF_DATA=TRANS_DATA
```

Using the `WHEN_REF_DATA=MASTER_DATA` option sets the mode of execution as master data and causes it to loop through all of the master data of the dimension in the subsequent `WHEN/ENDWHEN` statements. For example, all of the members in the Account dimension will be looped through in the upcoming example. Be aware that using the `MASTER_DATA` option may cause increased runtimes because all of the master values in the dimension will be processed when using this statement.

```
*WHEN_REF_DATA=MASTER_DATA
*WHEN ACCOUNT
*IS *
*REC(FACTOR=1, CATEGORY="PLAN")
*ENDWHEN
```


Using the WHEN_REF_DATA=TRANS_DATA option will limit execution to only the scoped data for the dimension. When an explicit WHEN_REF_DATA statement is not specified in the logic script, the transaction data mode is used by default.

```
*WHEN_REF_DATA=TRANS_DATA
*WHEN ACCOUNT
*IS *
*REC(FACTOR=1, CATEGORY="PLAN")
*ENDWHEN
```

Looping—FOR/NEXT Statement

FOR/NEXT is a looping statement that allows you to apply logic to a specific set of records. The logic module supports any level of nesting of FOR…NEXT loops in the body of the logic files. The *FOR/NEXT structure can also be included inside a *WHEN/ENDWHEN structure to create new records.

The syntax for the FOR/NEXT loop is as follows:

```
*FOR {variable1} = {set1} [ AND {variable2={set2}]
{text}
{text} …
*NEXT
```

The FOR/NEXT loop also supports up to two variables iterating on two independent sets of members. Consider this example:

```
*FOR %MYTIME% = %TIME_SET%
// logic content
*NEXT
```

COMMIT Statement

You can use the COMMIT instruction anywhere in the code to commit or update the data to the database.

```
*COMMIT
```

Revaluing Using an MDX Statement

You can use the following MDX statement in an expression to revalue revenues to increase them by 10%:

```
[Account].[#Revenue] = [Account].[Revenue] * 1.10
```

The # sign is used to redefine the value posted to an account member. All calculated members should have an ID preceded by the number sign (#).

LOOKUP/ENDLOOKUP Statement

You can use the `LOOKUP/ENDLOOKUP` statement in conjunction with a `WHEN/ENDWHEN` structure to retrieve ("lookup") values that may be needed either to calculate a new value or to define criteria to be evaluated. The lookup can be performed in the current model or in a different model.

The lookup mechanism defines a relationship between the current record being processed and another record in a corresponding user-defined record set. For example, when performing currency translation, you may want to look up, in the RATES model, the value of the rate based on current entity, category, and period.

The syntax is the following:

```
*LOOKUP {App}
*DIM [{LookupID}:] {DimensionName}="Value" | {CallingDimensionName}
[.{Property}] [*DIM …]
*ENDLOOKUP
```

In this syntax, `{App}` is the name of the model from which the values are searched; `{DimensionName}` is a dimension in the lookup model; `{CallingDimensionName}` is a dimension in the current model; and `{LookupID}` is an optional identifier of the "looked-up" amount. This is only required when multiple values must be retrieved. Refer to the following example:

```
*LOOKUP RATES
*DIM R_ENTITY="Global"
*DIM SOURCECURR:INPUTCURRENCY=ENTITY.CURRENCY
*DIM DESTCURR1:INPUTCURRENCY="USD"
*DIM DESTCURR2:INPUTCURRENCY="EURO"
*DIM R_RATE=ACCOUNT.RATETYPE
*ENDLOOKUP
```

In this example, three values are retrieved from the `INPUTCURRENCY` dimension (the rate of the currency of the current entity, the rate of the EURO currency, and the rate of the USD currency). Each of these values has been assigned a specific identifier (`SOURCECURR`, `DESTCURR1`, and `DESTCURR2`) that will be used somewhere in the `WHEN/ENDWHEN` structure.

Any dimension not specified in the lookup instruction is assumed to match a corresponding dimension member in the source model data.

Time Offset

A new keyword, `TMVL`, is available to calculate offset time values from a given time period. The format for this parameter is as follows:

```
TMVL(offset, base_period)
```

For example, you could use the following statement to offset the time period by four months, starting in January 2012:

```
TMVL(4, 2012.JAN)
```

Note the following about offsets:

- Offsets can be either negative or positive.
- The base period can be a hard-coded value, as in the previous example. You can also use a time script variable, `%TIME_SET%`, or a Data Manager variable prompt variable such as `$NEWPER$`.
- You can use `TMVL` in the following:
  - `FACTOR/EXPRESSION` within `REC`
  - `FOR/NEXT` loops
  - `IS` conditions inside `WHEN/ENDWHEN`

  Note that nested `TMVL` parameters such as `TMVL(-1, TMVL(-3, 2009.JAN))` are not supported.

Copy Data across Models

The keyword `DESTINATION_APP` is available to copy data from a source model to a destination model. The destination model may not contain all of the dimensions available in the source model. In these cases, you must add the following keyword to skip those dimensions:

```
*SKIP_DIM= {dimension name}[,{dimension name},…]
```

If the destination model contains dimensions that are not available in the source model, they can be added by using the `ADD_DIM` keyword and specifying a dimension name and value:

```
*ADD_DIM {dimension name}={value}[,{dimension name}={value},…]
```

If the dimension names in the destination model are named differently from their names in the source model, they can be renamed using the RENAME_DIM keyword. You use the RENAME_DIM keyword as follows:

```
*RENAME_DIM {dimension name}={value}[,{dimension name}={value},…]
```

You can add multiple dimension names to the instruction, separated by commas, when using the SKIP_DIM, ADD_DIM, and RENAME_DIM keywords.

When the DESTINATION_APP keyword is defined in a script, all of the subsequent statements used for writing data are directed to the model referenced in the DESTINATION_APP keyword.

The upcoming example of using the DESTINATION_APP keyword in a script file was taken from SAP Help. To explain DESTINATION_APP with SKIP_DIM, ADD_DIM, and RENAME_DIM, let's say that another model DETAIL_PLAN is created by copying the Planning model from the delivered ENVIRONMENTSHELL. So it is necessary to do the following:

1. Create anew the dimensions Product and Market, and add these to the DETAIL_PLAN model.
2. Replace P_ACCT with P_ACCTDETAIL.
3. Remove the P_ACTIVITY dimension from the DETAIL_PLAN cube.

The following is the syntax for using the DESTINATION_APP command in your script file:

```
*XDIM_MEMBERSET TIME = 2006.AUG
*XDIM_MEMBERSET CATEGORY=ACTUAL
*DESTINATION_APP=DETAIL_PLAN
*SKIP_DIM = P_ACTIVITY
*ADD_DIM P_AUDITTRAIL=INPUT, PRODUCT = NO_PRODUCT, MARKET = NO_MARKET
*RENAME_DIM P_ACCT=P_ACCTDETAIL
*WHEN CATEGORY
*IS "ACTUAL"
*REC(EXPRESSION=%VALUE%)
*ENDWHEN
```

In this example, DETAIL_PLAN is the target model, which has all dimensions of PLANNING except for P_ACCT. This dimension is replaced with P_ACCTDETAIL. Also,

DETAIL_PLAN has two additional dimensions: PRODUCT and MARKET. The logic script sets the member values for those dimensions in the DETAIL_PLAN mode as follows:

- PRODUCT and MARKET are set to NO_PRODUCT and NO_MARKET, respectively.
- The value of P_AUDITTRAIL is set based on INPUT.
- The value of P_ACCT is copied to P_ACCTDETAIL.

You now know a few commands you can use when developing logic script. In the next section, we'll explain how to execute logic script.

6.2.2 Executing Logic Script

You can execute logic script two ways: either by including it in the file *DEFAULT.LGF* and automatically executing it when users enter or load data to a model, or executing it using a Data Manager package.

DEFAULT.LGF is a special type of logic script file that is executed when you load data or enter data using an input template. The SAP BPC system runs the default logic indicated in the *DEFAULT.LGF* file of the model.

You can also execute a logic script file from the Data Manager. Using a Data Manager package, you can develop logic script to perform specific data management tasks and execute the logic script.

Let's now explore how our case study company, Rich Bloom, uses logic script to revalue plan data.

6.2.3 Creating Logic Script to Revalue Plan Data

Rich Bloom wants to revalue the plan data for specific products, so we'll explain how this is done using logic script. In this example, we'll create logic script that will allow a user to select specific products and specify a percentage by which to revalue the plan. We'll first create a logic script file in SAP BPC that contains the logic to perform revaluation. Then we'll create a process chain in BW that can be used to execute the logic script. Finally, we'll configure a data management package in SAP BPC that is associated with the logic script file and process chain and can be executed as needed.

1. Log in to the SAP BPC Web Client, and connect to the ZRB_PR_PLAN environment.

2. In the Administration workspace, under the Rules domain, select the Logic Scripts view. Select the scripts for the ZRB_SALES_CMB model, and click on the Open option (Figure 6.4, ❶). This opens the scripts used in the ZRB_SALES_CMB model. Click on New to create new logic file (Figure 6.5, ❷).
3. In the Create a New Logic Script dialog box, enter "REVALUE_PLAN" in the Logic Script File Name text box, and click on Create (Figure 6.5, ❸ and ❹).

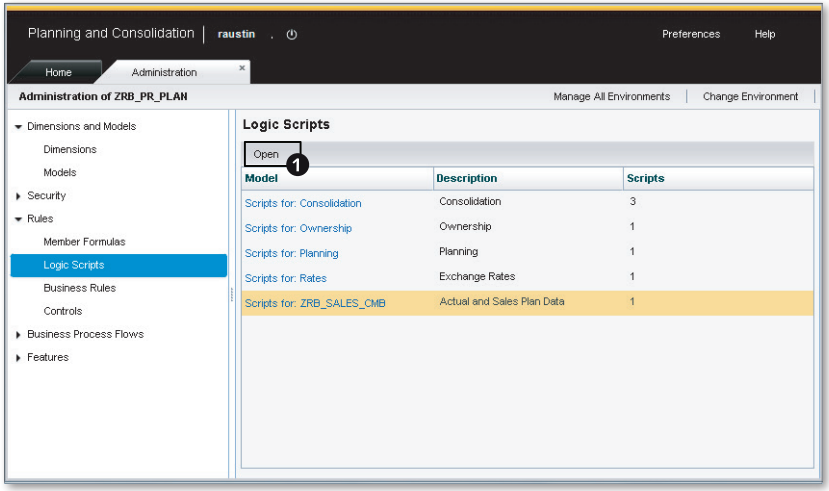


Figure 6.4 Creating Logic Script to Revalue Plan Data—Part A

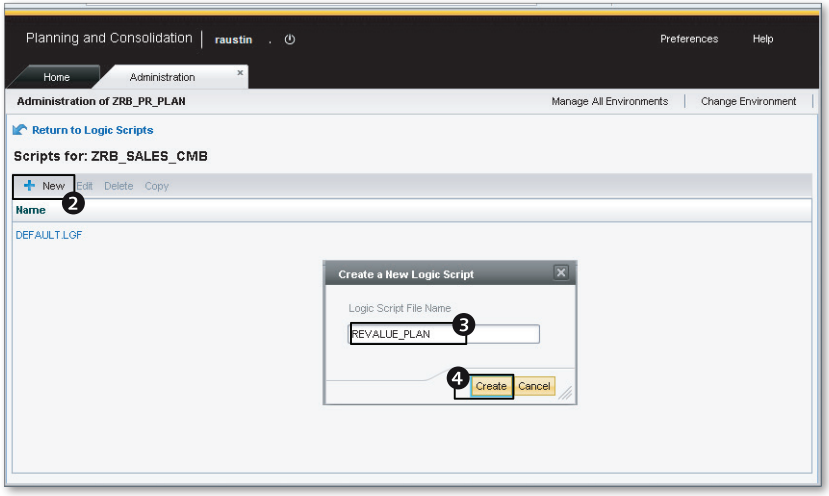


Figure 6.5 Creating Logic Script to Revalue Plan Data—Part B

4. Next, you can enter the logic script for revaluing the plan data. Enter the code as shown in Figure 6.6, ❺. The script is used to prompt the user to enter a percentage for revaluing revenue, and the increase or decrease in revenue is posted back to the revenue account (410000); it is calculated by multiplying the percentage increase/decrease (entered by the user) to the revenue amount. After entering the code, click on VALIDATE, and then click on SAVE (Figure 6.6, ❻ and ❼). You will get an error message when you validate the logic script stating that there is an unreferenced variable $Z_INCREASE_REV$. This error message is okay because this variable will be used in the Data Manager package prompt.

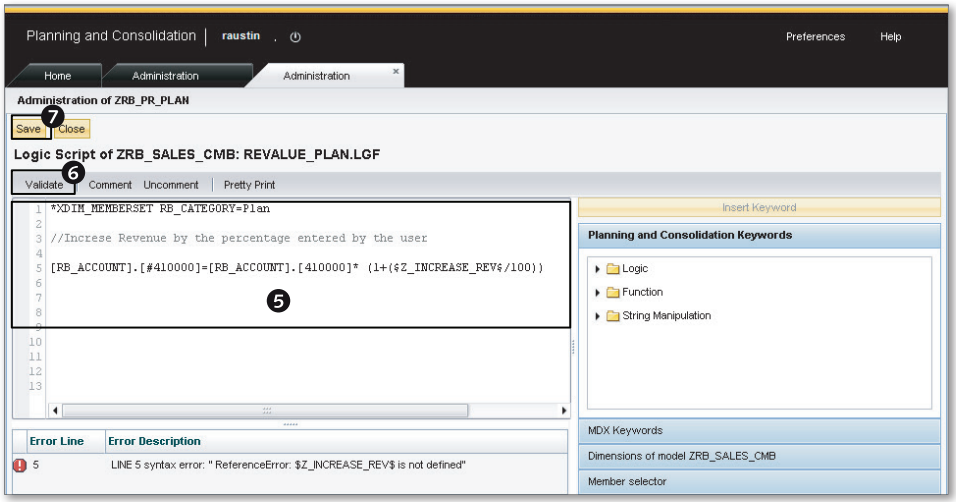


Figure 6.6 Creating Logic Script to Revalue Plan Data—Part C

5. Create a new process chain using Transaction RSPC in the SAP NetWeaver BW system, as shown in Figure 6.7. In this step, we'll show you how to create a process chain for running logic script. But you do not have to create a separate process chain for running logic script; you can use process chain `/CPMB/DEFAULT_FORMULAS`, which is delivered with SAP BPC for NetWeaver. The process chain will include the START process type and the following standard process types that are specific to SAP BPC:

   - MODIFY DYNAMICALLY
   - RUN LOGIC
   - CLEAR BPC TABLES

Note

The CLEAR BPC TABLES process type is used in SAP BPC process chains to clear temporary tables during processing.

6. Make sure you include the process types in the same order as shown in Figure 6.7. Every process type will be associated with a variant, and the variant for the START process type has to be unique for every process chain in the system. Create a new process variant for the START type.

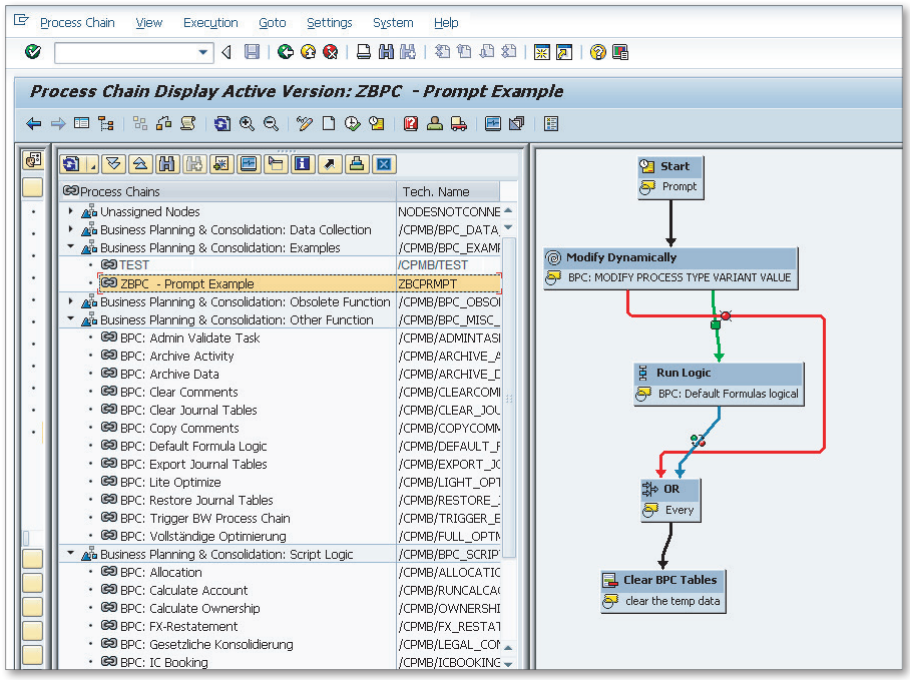


Figure 6.7 Creating Logic Script to Revalue Plan Data—Part D

7. Now, associate this process chain with a data package in SAP BPC by going to the EPM interface for Excel and connecting to the ZRB_SALES_CMB model in the ZRB_PR_PLAN environment. In the EPM DATA MANAGER tab, select the ORGANIZE PACKAGE LIST menu option to create a new data package, and associate the process chain you created in the previous step with this package (Figure 6.8, ❽).

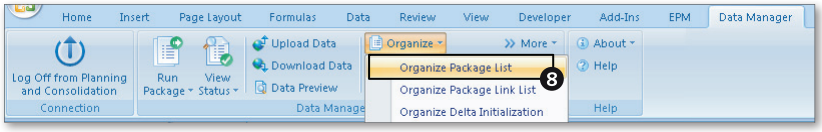


Figure 6.8 Creating Logic Script to Revalue Plan Data—Part E

8. This will open a new dialog box where you can maintain and create new package groups and Data Manager packages. First we'll create a new package group called RB CUSTOM. Click on the ADD PACKAGE GROUP option under the PACKAGE menu (Figure 6.9, ❾). In the ADD PACKAGE GROUP dialog box, enter "RB CUSTOM", and click on OK (Figure 6.9, ❿ and ⓫).

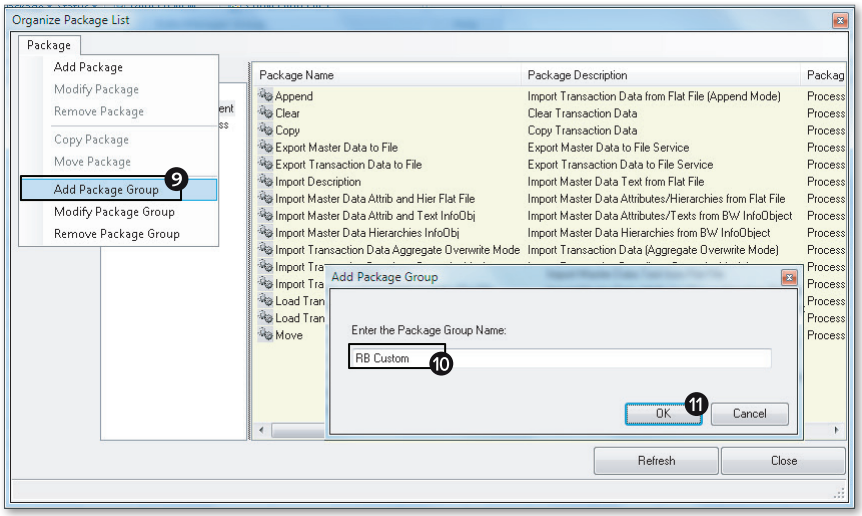


Figure 6.9 Creating Logic Script to Revalue Plan Data—Part F

9. We will now create a new package under the RB_CUSTOM group we created in the previous step. (This is just an example; you can place the package into any group you want.) Click on the ADD PACKAGE GROUP option under the PACKAGE menu (Figure 6.10, ⓬). In the ADD PACKAGE dialog box, click on CHANGE, and select the ZBCPRMPT process chain you created earlier in the SAP NetWeaver BW system (Figure 6.10, ⓭ and ⓮). This process chain is run when this data package is executed. Enter "Increase Plan Revenue" as the package name (Figure 6.10, ⓯). Select RB CUSTOM as the group where this package should belong using the dropdown box, and enter "Package to Increase Plan Revenue" for the description (Figure 6.10, ⓰ and ⓱).

10. To give appropriate access to the package, specify whether users will be authorized to execute it. If you select User Package, users can execute the package. If you select Admin Package, only administrators can execute it (Figure 6.10, ⓲). After specifying the selection, click on Add (Figure 6.10, ⓳).

    After you add the package, you'll see the new package named Increase Plan Revenue (Figure 6.11).

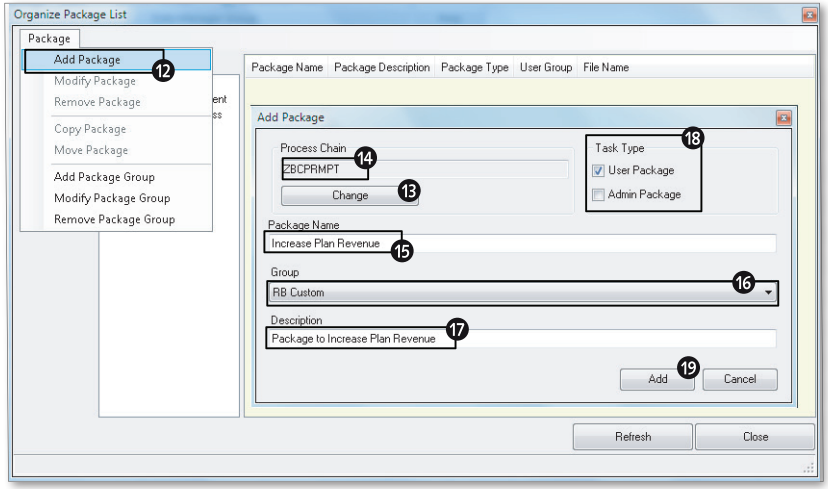


Figure 6.10 Creating Logic Script to Revalue Plan Data—Part G

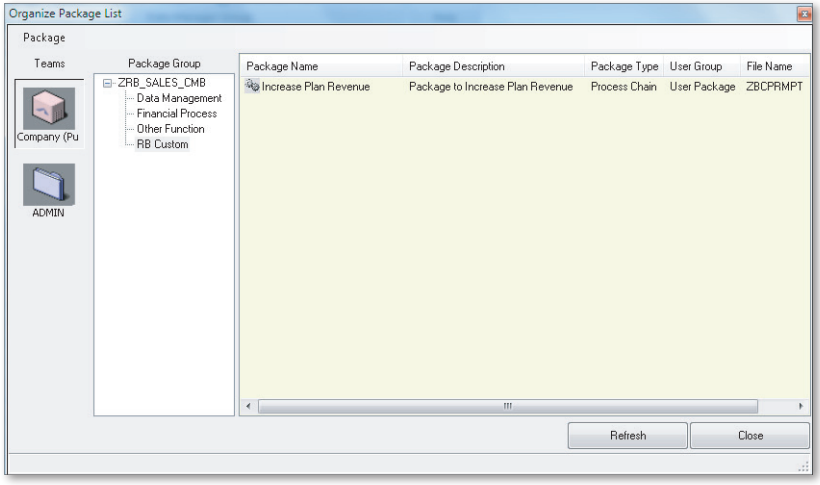


Figure 6.11 Creating Logic Script to Revalue Plan Data—Part H

11. Next, you need to associate the logic script file with the data management package you created. In the EPM DATA MANAGER tab, select the ORGANIZE PACKAGE LIST to modify the data package you created. Select the INCREASE PLAN REVENUE package, and right-click on the MODIFY option. The MODIFY PACKAGE dialog box displays; click on MODIFY SCRIPT (Figure 6.12, ⓴).

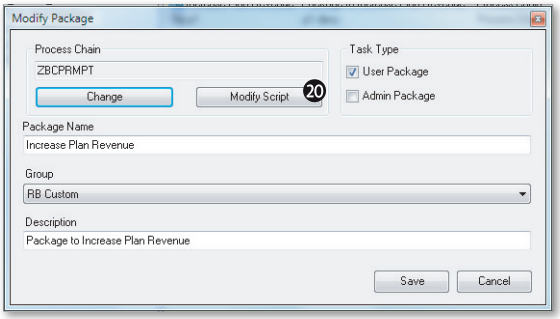


Figure 6.12 Creating Logic Script to Revalue Plan Data—Part I

12. The MODIFY SCRIPT dialog box displays. Click on ADVANCED to specify the package settings (Figure 6.13, ㉑).

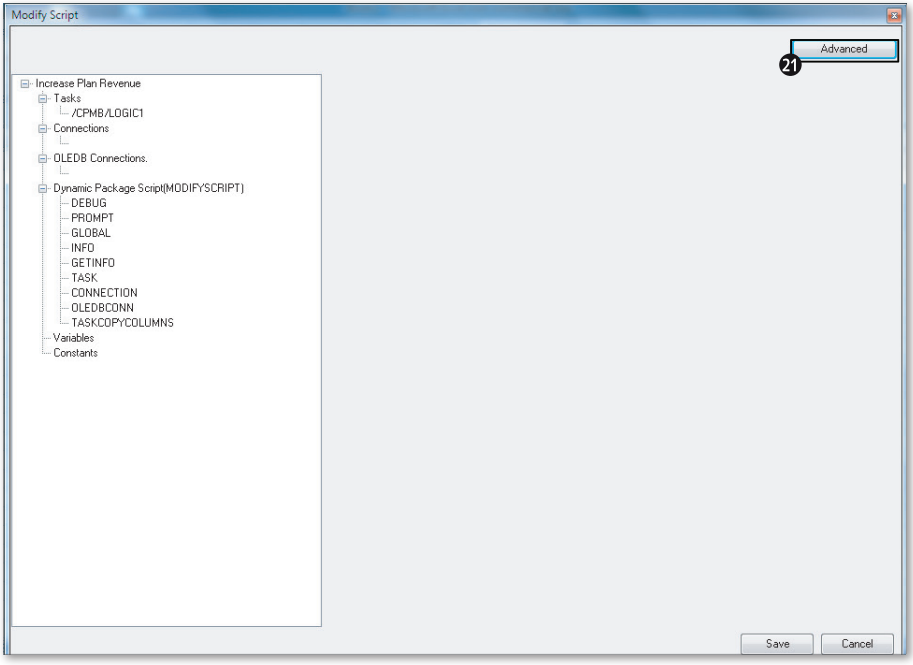


Figure 6.13 Creating Logic Script to Revalue Plan Data—Part J

13. Enter the following code, which is also shown in Figure 6.14, ㉒.

```
PROMPT(SELECTINPUT,,,,"%CATEGORY_DIM%,%CURRENCY_DIM%,%TIME_
DIM%,%DATASRC_D
IM%,%ENTITY_DIM%,RB_PRODUCT,RB_SALESGRP")
PROMPT(TEXT,%Z_INCREASE_REV%,"Input Revenue Increase in Percent",)
INFO(%EQU%,=)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,SUSER,%USER%)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,SAPPSET,%APPSET%)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,SAPP,%APP%)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,SELECTION,%SELECTION%)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,LOGICFILENAME,REVALUE_PLAN.LGF)
TASK(/CPMB/DEFAULT_FORMULAS_LOGIC,REPLACEPARAM,Z_INCREASE_REV%EQU%%
Z_INCREASE_REV%)
```

14. You can also copy the code from a file. Note that in this setting, we're prompting the user to enter selection values for the Entity, Category, Currency, Time, AuditTrail, Sales group, and Product Dimensions fields. We're also prompting the user to enter a percentage in the `%Z_INCREASE_REV%` field for revaluation. The name of the process variant we used in the process chain for the run logic task is `/CPMB/DEFAULT_FORMULAS_LOGIC`. In this code, we're also passing the values entered by the user to the process chain.

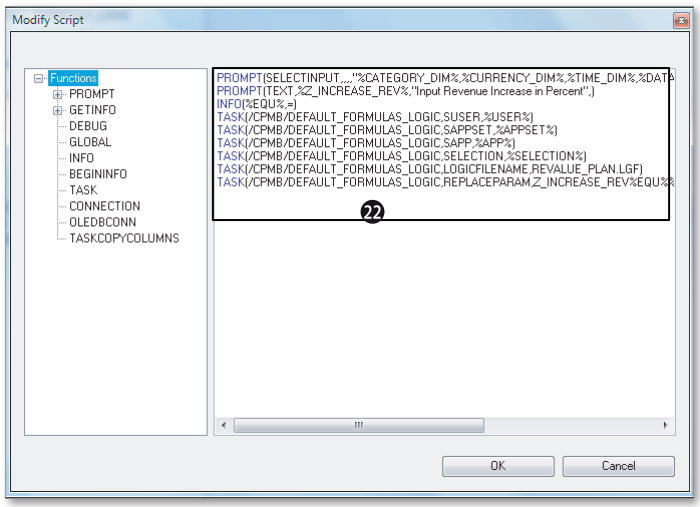


Figure 6.14 Creating Logic Script to Revalue Plan Data—Part K

15. Run a report to see the data before the revaluation is done (Figure 6.15).

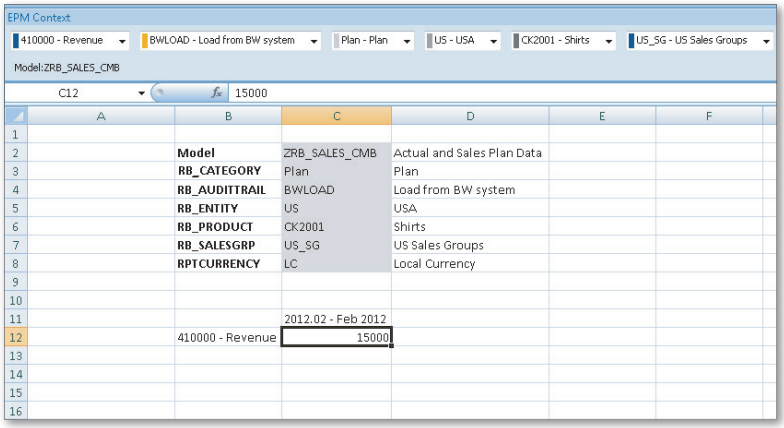


Figure 6.15 Creating Logic Script to Revalue Plan Data—Part L

16. You're now ready to execute the package to increase the plan revenue. From the Data Manager tab, select Run Packages. Select the Increase Plan Revenue package, and click on Run.

17. You're prompted to enter the member value selections for the dimensions. Select Run Now, and click on the Next button. Enter the selections for which you want to increase revenue, and click on Next again to run the report (Figure 6.16). Note that we selected product CK2001 for revaluation.

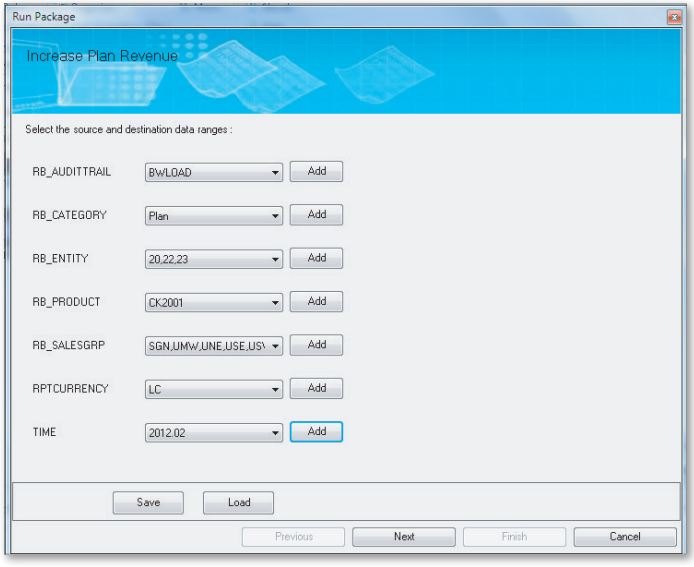


Figure 6.16 Creating Logic Script to Revalue Plan Data—Part M

18. Enter the revenue increase percent you want to effect, and click on NEXT (Figure 6.17).

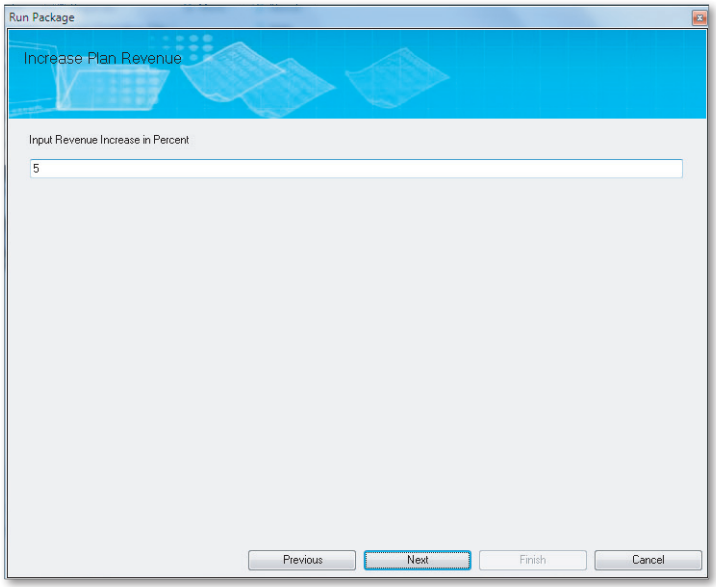
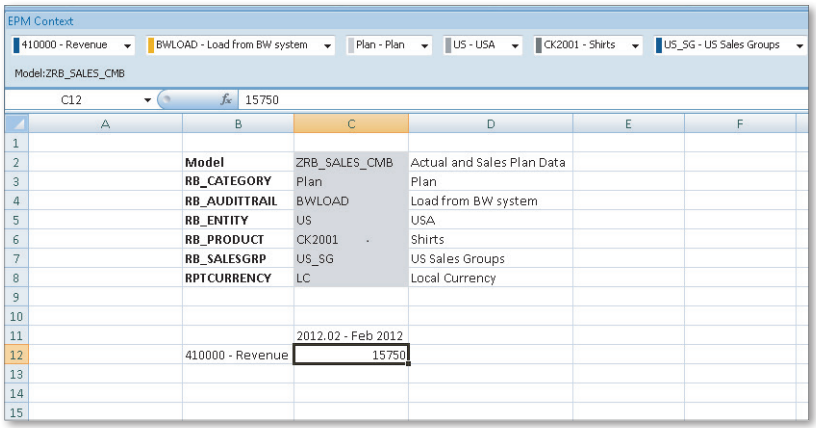


Figure 6.17 Creating Logic Script to Revalue Plan Data—Part N

19. Click on RUN NOW. You'll see that product CK2001 has an amount posted to the revenue account, as shown in Figure 6.18. This amount represents revaluation to the revenue.



	A	B	C	D	E	F
1						
2		**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data		
3		**RB_CATEGORY**	Plan	Plan		
4		**RB_AUDITTRAIL**	BWLOAD	Load from BW system		
5		**RB_ENTITY**	US	USA		
6		**RB_PRODUCT**	CK2001	Shirts		
7		**RB_SALESGRP**	US_SG	US Sales Groups		
8		**RPTCURRENCY**	LC	Local Currency		
9						
10						
11			2012.02 - Feb 2012			
12		410000 - Revenue	15750			
13						
14						
15						

Figure 6.18 Creating Logic Script to Revalue Plan Data—Part O

Now that you understand how to use logic script to revalue revenue for a specified period in a model, we'll cover creating logic script to convert data that is YTD into periodic for PERIODIC models.

6.2.4 Creating Logic Script to Push Data from the YTD to PERIODIC Models

You can have two models in your environment: a YTD model that contains consolidated actual data, and a PERIODIC model that contains plan data. You may need to push data for a single period from the YTD model to the PERIODIC model. The following logic script in Listing 6.1 can be used to meet this objective. For income and expense accounts, the script calculates the periodic values by subtracting YTD values of the previous period from the YTD values of the single period entered by user. For the month of January, the script does not do the subtraction but sets the YTD value as the periodic value. For balance sheet accounts, the script pushes the ending balances for the single period entered by user. In this example, the name of the destination model that contains the plan data is `PLAN_APP`. The logic script is created and executed in the model that contains the YTD consolidated actual data.

```
//Push data for a single period from YTD model to PERIODIC model
*SELECT(%PYEAR%,"[YEAR]","TIME","[ID]=%TIME_SET%")
//Prevents multiple periods in time set
*SELECT(%PPERIOD%,"[ID]","TIME","[ID]=%TIME_SET%")

//Scope-in prior period and selected period
*XDIM_MEMBERSET TIME=TMVL(-1,%PPERIOD%),%PPERIOD%

//Scope-in ACTUAL that needs to be pushed
*XDIM_MEMBERSET CATEGORY=ACTUAL
// Push data to PLAN_APP model
*DESTINATION_APP = PLAN_APP

*WHEN TIME.YEAR
//Used for month of January - Prevent processing of December
//of prior year
*IS %PYEAR%
  *WHEN TIME
  *IS %PPERIOD% // Period selected by user

     *WHEN ACCOUNT.ACCTYPE
      *IS AST,EXP
```

```
        *REC(FACTOR=1)
       *IS LEQ,INC
        *REC(FACTOR=-1)
       *ENDWHEN // ACCOUNT.ACCTYPE

// Period previous to period selected by user
  *IS TMVL(-1,%PPERIOD%)
*WHEN ACCOUNT.ACCTYPE
         *IS EXP
          *REC(FACTOR=-1, TIME=%PPERIOD%)
         *IS INC
          *REC(FACTOR=1, TIME=%PPERIOD%)
         *ENDWHEN // ACCOUNT.ACCTYPE
  *ENDWHEN // TIME

*ENDWHEN // TIME.YEAR
```

Listing 6.1 Push Data for a Single Period from YTD Model to PERIODIC Model

6.2.5 Logic Script Files Provided by SAP

SAP BPC provides a library of standard logic script functions, all of which have the file extension .lgf. Table 6.2 shows a list of logic script files.

	Script File	Description
1.	*Allocation.lgf*	Runs allocation logic
2.	*Calcaccount.lgf*	Runs account transformation business rule
3.	*Consolidation.lgf*	Runs a legal consolidation business rule
4.	*Copy_Opening.lgf*	Runs a balance carry forward business rule
5.	*FX_Trans.lgf*	Runs currency conversion
6.	*ICBooking.lgf*	Runs the IC Booking business rule (based on business rules, this script automatically books mismatches in intercompany entries entered by an entity and its trading partner)
7.	*ICData.lgf*	Performs intercompany reconciliation (the script creates entries that allow users to match intercompany transactions)

Table 6.2 Logic Script Files Provided by SAP

	Script File	Description
8.	*ICElim.lgf*	Runs intercompany elimination
9.	*MDXlib.lgf*	Contains a library of MDX functions
10.	*System_Constants.lgf*	Stores constant values for use within a logic script
11.	*System_Library.lgf*	Includes basic examples of a set of keywords

Table 6.2 Logic Script Files Provided by SAP (Cont.)

Note

You can use Transaction UJKT to debug logic script in the SAP NetWeaver BW system. This transaction allows you to test the execution of a script for a particular data region.

In the next section, we'll discuss business rules and explain how to use them in a model.

6.3 Business Rules Overview

In the previous section, you saw how to code logic script to perform certain data management tasks. Instead of writing code, you can configure *business rules* to perform certain tasks related to planning and consolidation. The code is already available in SAP function modules; you just need to call them in a script and pass the appropriate parameters to perform the preferred function.

In this section, we'll provide a brief introduction to the business rules you can configure in SAP BPC. We'll discuss setting up business rules to perform currency translation. (We also discuss business rules in detail in Chapter 9, when we discuss the process for consolidating data in SAP BPC.)

To begin, let's discuss the different business rules that can be configured in SAP BPC.

6.3.1 Currency Translation

Currency translation is required when an organization executes business in more than one country and uses multiple currencies for business transactions. When an organization uses multiple currencies, it needs to convert transactions in local currency to one or more reporting currencies. Currency translation is used in both

planning and financial consolidation models. Business rules can be set up to perform currency translation. We'll discuss the steps and configuration for setting up business rules for performing currency translation in this chapter.

6.3.2 Carry Forward

The carry forward of balances is an essential step in the creation of a balance sheet for an organization. Carry forward refers to the closing balance of accounts for a fiscal period being transferred as the opening balance of a subsequent fiscal period. The carry forward balance rules can also be used to close current period net income and post to the balance sheet to create a fully balanced balance sheet. The setup for this process is taken care of in SAP BPC with the use of a business rules table. We'll discuss the steps and configuration for setting up this business rule in Chapter 9.

6.3.3 Account Transformation

There are some scenarios where accounts are consolidated or transformed into another account, which is an activity used extensively in cash flow models. Account transformation business rules are used to aggregate values posted to specific combinations of source account, flow, category, and data source and post them to aggregated destination account, flow, category, and data source. We'll discuss the steps and configuration for setting up this business rule in Chapter 9.

6.3.4 Intercompany Matching/Booking

Intercompany matching/booking is applicable for an organization that has entities that do business with each other. When intercompany transactions exist, it is important to ensure that they are matched between entities before running the intercompany elimination process. For example, if an entity within an organization reports an intercompany sales transaction, the trading partner associated with this transaction should post an intercompany expense transaction. The two transactions will need to be matched under a matching currency. This is required to produce accurate consolidated results. The intercompany booking business rule allows you to post/book differences in amounts reported by the entities so that intercompany transactions match. We'll discuss the steps and configuration for setting up this business rule in Chapter 9.

6.3.5 Intercompany Elimination

This is applicable for an organization that has entities that do business with each other. A consolidated balance sheet of the organization shows transactions that relate to external companies. The transactions such as sales and COGS or receivables and payables that result from exchange of goods or services within the organization should not be counted. In these cases, it is essential to eliminate the intercompany transactions. The intercompany eliminations process creates entries that eliminate intercompany transactions. There are two methods of performing intercompany elimination, one using U.S. elimination and the other using elimination and automatic adjustments business rule. Both of these methods use business rules. We'll discuss the steps and configuration for setting up these business rules in Chapter 9.

6.3.6 Eliminations and Automatic Adjustments

We discussed the usage of the elimination and automatic adjustments business rule to perform intercompany elimination. This business rule is also used for consolidation of investments. The elimination and automatic adjustment business rule supports the calculation and generation of these postings. We'll discuss the steps and configuration for setting up this business rule in Chapter 9.

You can access the Business rules view and Add or Remove Rule Types for a model under the Rules domain in the Administration workspace (Figure 6.19).

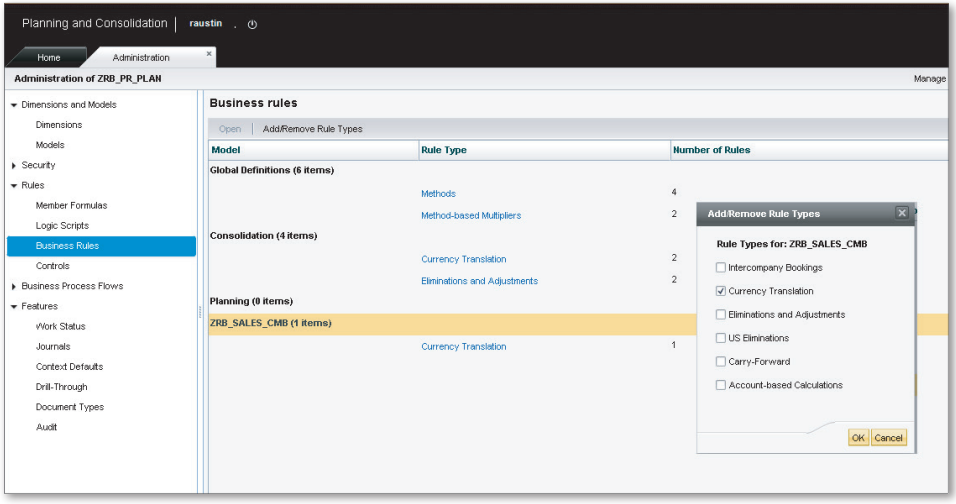


Figure 6.19 Maintain Business Rules for a Model

In the next section, we'll discuss how currency translation is performed in SAP BPC and the steps to configure the business rule to meet this requirement.

6.4 Currency Translation

Let's shift our attention to currency translation using SAP BPC. An organization may have businesses in more than one country and conduct business transactions in many currencies. The currency translation business rule is used to translate all business transactions to one or more reporting currencies.

The currency that is associated with an entity is referred to as the *local* currency. The transaction data in SAP BPC may be stored in local currency. These transactions are converted using the currency translation business rule to one or more reporting currencies or group currencies.

We'll now explain how to use the currency translation business rule to translate all financial transactions to one or more reporting currencies. First, we'll explore the contents of the RATES model that is used in the currency translation process.

RATES Model

SAP BPC provides a model called RATES, which is a supporting model used by financial and consolidation models. It stores exchange rates to support currency conversion. It is important that the Category and Time dimensions in the RATES model are identical to the respective dimensions in the model that uses the RATES model for performing currency translation. The RATES model has the following dimensions:

- CATEGORY
- INPUTCURRENCY
- R_ACCT
- R_ENTITY
- TIME

You can enter exchange rate data into the RATES model either by using input templates or by loading the data.

The value of the CATEGORY dimension specifies the category to which the exchange rate is applicable. Sometimes we may maintain data for several categories in our planning or consolidation models, and it may not be feasible to maintain exchange rates for each of the categories in the RATES model, especially if the exchange rates for all categories for an exchange rate type and period are going to be the same. In this case, we can maintain exchange rates for a generic category in the RATES model. In the planning and consolidation models that contain the source records that will need to be translated, we can set the `RATE_CATEGORY` property of the CATEGORY dimension to this generic value. The currency conversion process will then look for the exchange rate in the RATES model based on this property associated with the CATEGORY of the source record.

The value of the R_ENTITY dimension specifies the entity to which the exchange rate is applicable. We can set this value to `Global` and can use it for currency conversion across all entities. In this case, when we call the program to perform currency translation, we specify `RATEENTITY = Global`.

The value of the R_ACCT dimension is the account rate type for which the exchange rate is applicable for currency conversion. The RATES model contains the currency translation exchange rate to use for different periods and for different exchange rate types. The exchange rate type signifies the type of rate—for example, average rate or month-end rate. The business rules table for currency conversion defines logic for currency conversion based on the account rate type.

The value of the TIME dimension is the period for which the exchange rate is valid. We can use the `RATE_YEAR` and `RATE_PERIOD` properties of the CATEGORY dimension to determine the year and period that the source records would look up during currency conversion to determine exchange rates. If these properties are filled in the CATEGORY dimension, then during currency conversion, based on the CATEGORY of the source record, the system would look up the values in these properties to determine year/period for determining the exchange rate.

The INPUTCURRENCY dimension within the RATES models must include the `MD` property, whose values can be either `M` (to multiply rates) or `D` (to divide rates).

The currency exchange rate is maintained for each currency. In the RATES model, if the exchange is with respect to U.S. dollars, then the rate for USD is set to 1. The rate for EUR is set to 1.25.

Next, we'll discuss the process of currency translation.

Currency Conversion

Currency translation involves the following steps:

1. Data for each entity is either entered or loaded into the SAP BPC system in local currency (`CURRENCY=LC`). If group dimension (`type=G`) is used in the data model, then when loading or entering data, the value of group dimension is set to `NON_GROUP` where `NON_GROUP` represents the dimension member of the group dimension that has the `CURRENCY_TYPE` property set to `N` (non-group). Group dimension is used in legal consolidation models.
2. The reporting/group currencies for which translation is required are specified when running the Data Manager package for curency conversion
3. When the currency translation process is run for a data region, the system reads all records for the data region that were entered or loaded in local currency and determines the local currency of each record based on the `CURRENCY` property associated with the `ENTITY` of the source record.
4. The `RATETYPE` to be used for currency translation is based on the `RATETYPE` property of the account in the record read.
5. When the currency translation process runs, it looks up business rules that pertains to the `RATETYPE` property of the account in the record read. The business rules table has the logic to perform the currency translation. The process creates records that have the translated values with reporting/group currencies in the currency dimension. If currency translation is run for a group, the value of the GROUP dimension is set with the ID of group.

Let's look at some important setup factors that relate to the translation of currency:

- The model that requires currency translation must be created as a financial or consolidation model.
- A model can include only one dimension for indicating currency type. This is the dimension that states whether the transaction is in the local or reporting currency.
- The Entity dimension has a `Currency` property that denotes the local currency associated with an entity.
- Set the `CURRENCY_TYPE` property of the currency dimension—`L` for local currency and `R` for reporting. For example, the `CURRENCY_TYPE` property of local currency (LC) is set to `L` and Euro (EUR) is set to `R`.

- Set the REPORTING property of the currency dimension. If a currency is marked as reporting (REPORTING=Y), currency translation can be used to convert transactions from the local currency to this reporting currency. For example, the REPORTING property of US dollars (USD) is set to Y.
- The Rate Type property of the Account dimension specifies the type of rate to be used for the translation.
- The Time dimension must include the YEAR, PERIOD, TIMEID, and MONTHNUM properties.
- The AuditTrail dimension must include the following properties:

  DATASRC_TYPE, whose values include the following:

  - I: Input
  - M: Manual adjustment
  - A: Automatic adjustment or elimination
  - L: Data source level (use only for consolidation)

  IS_CONVERTED, whose values include the following:

  - N (or blank): These members are ignored in the currency conversion process.
  - Y: These members are converted from the local currency (LC) into the preferred currency.
  - G: These members are copied from the reporting currency of the group that is being translated into the currency member corresponding to the given group.
- The Flow dimension is not mandatory for currency translation, but if it is included, the dimension must include the FLOW_TYPE property. This property is used in the currency translation business rule to perform "force closing." In this case, an additional entry is generated where the destination flow is set to a flow whose FLOW_TYPE is equal to CLOSING.
- When we discussed the RATES model, we discussed certain properties of the CATEGORY dimension related to currency conversion such as RATE_CATEGORY, RATE_YEAR, and RATE_PERIOD. In addition, we can use the FX_SOURCE_CATEGORY of the CATEGORY dimension to determine the category of source data to be used for currency conversion. The FX_DIFFERENCE_ONLY property, if set to Y, is used to post differences between the calculated (default) value and the source (simulated) value.

You can run the currency translation program in two modes: reporting currency mode and group currency mode. In reporting currency mode, transactions are translated from the local currencies to the reporting currencies. You should include the following script:

```
*RUN_PROGRAM CURR_CONVERSION
CATEGORY = %C_CATEGORY_SET%
CURRENCY = %RPTCURRENCY_SET%
TID_RA = %TIME_SET%
RATEENTITY = Global
......
*ENDRUN_PROGRAM
```

The group currency mode is used in an organization that is composed of a group of subsidiaries and where the subsidiary data should be converted to one or more group currencies.

A consolidation group represents the relationship of entities for a given consolidation result. The relationship between the entities is stored in the OWNERSHIP model. In this case, the group dimension is added to the data model. We will study the OWNERSHIP model in detail in Chapter 9. The following properties must be set in the group dimension to enable currency translation:

- Set the `CURRENCY_TYPE` property for members in the group dimension: `G` for group currency, `N` for non-group (group input), and `L` for local currency.
- Set the `GROUP_CURRENCY` property for the groups (e.g., USD, EUR, etc.): The currency translation process translates data in local currency to group currency that is specified in the `GROUP_CURRENCY` property of the Group dimension for the group.

Note

For a consolidation model, it is recommended to have a GROUP dimension (type `G`) and a CURRENCY dimension (type `R`) to be able to report in multiple group currencies.

```
*RUN_PROGRAM CURR_CONVERSION
CATEGORY = %C_CATEGORY_SET%
GROUP= %GROUPS_SET%
TID_RA = %TIME_SET%
RATEENTITY = Global
......
*ENDRUN_PROGRAM
```


In addition, you maintain the currency conversion business rules table to define the rules for currency translation. When the currency translation Data Manager package is run, the rules defined in this table are used during the currency translation process.

The following keywords are used in the FORMULA field of the business rules table. These keywords are used to determine the amount to post during currency translation.

- `AS_IS`: Do not change existing value in target currency.
- `AVG`: Average rate as stored in the RATES model.
- `COPYLC`: Translate the local currency to reporting/group currency with `factor=1`.
- `END`: Month-end rate as stored in the RATES model.
- `OPEEND`: `END` rate of last period of prior year.
- `OPEAVG`: `AVG` rate of last period of prior year.

Table 6.3 explains the details for setting up the business table for currency conversion.

Field Name	Description
Account Rate Type	Specify the account rate type to use in the currency translation (e.g., `AVG`, `END`, `HIST`).
When the rate type property value of the source account is the same as defined in the rule, the criteria for using the business rule is met.	
Source Flow	Specify the source flow as additional criteria to use in the translation. The value specified in this field can be one of the following: ▸ Base member ▸ Parent member A list of members is defined and filtered using a value of the `DIMLIST` property in the Flow dimension. Balance sheet accounts, such as inventory accounts, have an opening balance at the beginning of a period, additions and transfers for the period, and a closing balance for end of the period. The Flow dimension helps break the account into various flow types such as opening, additions, transfers, and closing

Table 6.3 Currency Translation Business Rules

Field Name	Description
Destination Account	Specify a base member as the destination account. If no value is specified for the destination account, the value of the source account is used.
Destination Flow	Specify a base member for the destination flow. If no value is specified for the destination flow, the value of the source flow is used.
Formula	Specify an arithmetic expression to define calculation of the amount during currency translation. In the Formula field, we can specify formulas using `END-AVG`. In this case, the difference between the month-end rate and average rate is calculated and used by the currency conversion process.
Force Closing	When the Force Closing checkbox is selected, an additional entry is generated where the destination flow is the closing balance. This is applicable for members in the Flow dimension where the property `FLOW_TYPE` is equal to the value `CLOSING`.
Apply to Periodic	When the model stores data in YTD and must perform the currency translation based on periodic values, select the Apply to Periodic checkbox. When this checkbox is selected, the difference between the current period and prior period amounts is calculated, and the formula is applied on the resulting amount. The result is added back to the prior period's value as written in the current period.
Entity FX Type	When a value is specified in Entity FX Type, the rule will be applied to all of the entity members that have a matching value in the `FX_TYPE` property of the Entity dimension.
Remark	A brief description of the business rule.

Table 6.3 Currency Translation Business Rules (Cont.)

Table 6.4 displays the business rules.

Account Rate Type	Source Flow	Destination Account	Destination Flow	Formula	Force Closing
AVG				AVG	
ENDFLOW	F_OPE		F_DIFF	END-OPEEND	

Table 6.4 Currency Translation Business Rules (Sample)

The first business rule would look for all accounts in the source data that have RATE_TYPE property = AVG and perform currency translation based on the exchange rate for R_ACCT = AVG in the RATES model.

The second business rule would look for all accounts in the source data that have RATE_TYPE property = ENDFLOW and Flow = F_OPE and perform currency translation. The business rule would post the difference between the END (current month-end) rate and OPEEND (END rate of last period of prior year) to the same account and flow F_DIFF. In the example that we're going to illustrate, we do not have the FLOW dimension. However, this rule helps us understand the usage of the business rules if we use FLOW dimension in our data model.

We'll now use an example to demonstrate how currency translation is set up and performed in SAP BPC. One of the required dimensions for currency translation is AuditTrail. The planning model we use for our model company, Rich Bloom, includes the AuditTrail dimension. The objective is to understand how to convert currency data from local currency of U.S. entities—U.S. dollar (USD)—into the Euro (EUR).

We would normally load exchange rate data provided by a third-party vendor to the RATES model. But for our illustration, we'll enter this information directly into the RATES model.

1. Log in to the EPM interface for Excel, and connect to the RATES model in the ZRB_PR_PLAN environment. The Category dimension in the RATES model is modified to use the RB_CATEGORY dimension. The reason for this is that the Category and the Time dimension used in the RATES model and the ZRB_SALES_CMB model on which current translation is performed should be the same. Create an input form schedule, and enter the exchange rate for Jan 2012, as shown in Figure 6.20. The average rate (AVG) for Jan 2012 is entered for the USD and EUR currencies. We want to translate the data from USD into EUR, so we enter a factor of "1" for USD and a factor of "1.25" for EUR. Save the exchange rate data.

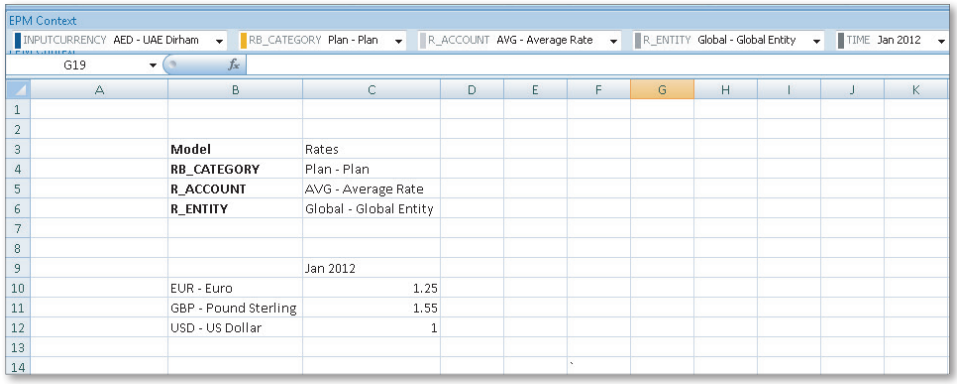


Figure 6.20 Performing Currency Translation—Part A

2. Log in to the SAP BPC Web Client, and connect to the ZRB_PR_PLAN environment. Select the Logic Script view under the Rules domain. Select the ZRB_SALES_CMB model, and click on the Open menu option to view and maintain logic scripts. Create a new logic script file with the name "FXTRANS" (Figure 6.21), and enter the following code:

```
*RUN_PROGRAM CURR_CONVERSION
CATEGORY = %RB_CATEGORY_SET%
CURRENCY = %RPTCURRENCY_SET%
TID_RA = %TIME_SET%
RATEENTITY = Global
OTHER = [ENTITY=%RB_ENTITY_SET%]
*ENDRUN_PROGRAM
```

Save and validate the logic file.

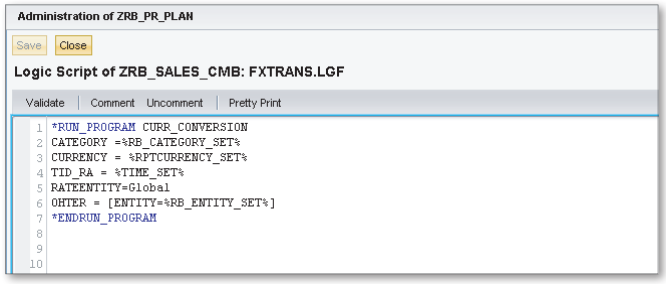
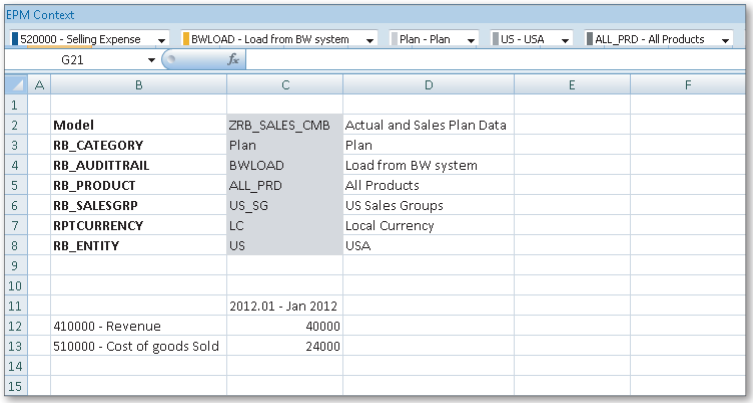


Figure 6.21 Performing Currency Translation—Part B

3. Return to the EPM interface for Excel, and connect to the ZRB_SALES_CMB model in the ZRB_PR_PLAN environment. Execute a report to see the data before the currency translation is executed. Figure 6.22 shows the data associated with all of the entities in the United States, in local currency. The data for accounts 410000 (REVENUE) and 510000 (COGS) are displayed.



	A	B	C	D	E	F
1						
2		**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data		
3		**RB_CATEGORY**	Plan	Plan		
4		**RB_AUDITTRAIL**	BWLOAD	Load from BW system		
5		**RB_PRODUCT**	ALL_PRD	All Products		
6		**RB_SALESGRP**	US_SG	US Sales Groups		
7		**RPTCURRENCY**	LC	Local Currency		
8		**RB_ENTITY**	US	USA		
9						
10						
11			2012.01 - Jan 2012			
12		410000 - Revenue	40000			
13		510000 - Cost of goods Sold	24000			
14						
15						

Figure 6.22 Performing Currency Translation—Part C

4. Execute the FX RESTATEMENT data package (Figure 6.23). The system will prompt you to enter parameter values (Figure 6.24). Our objective is to translate transactions in local currency to EUR currency. Select EUR for the report currency.

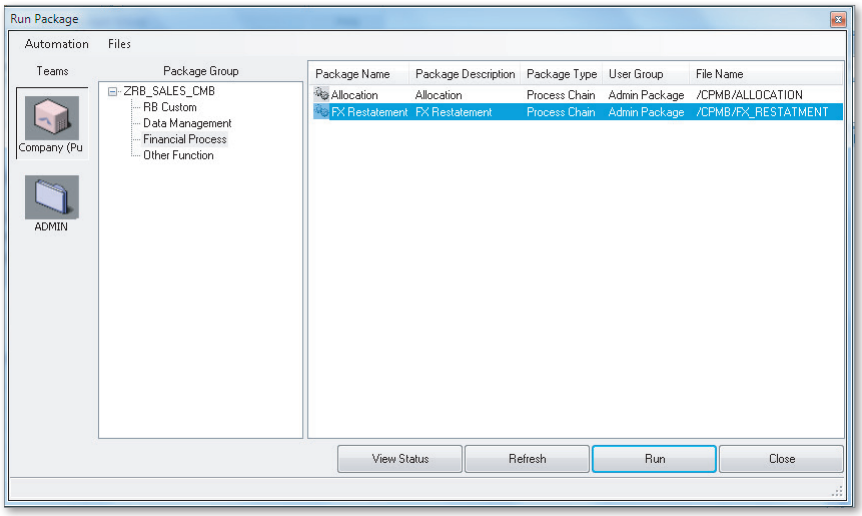


Figure 6.23 Performing Currency Translation—Part D

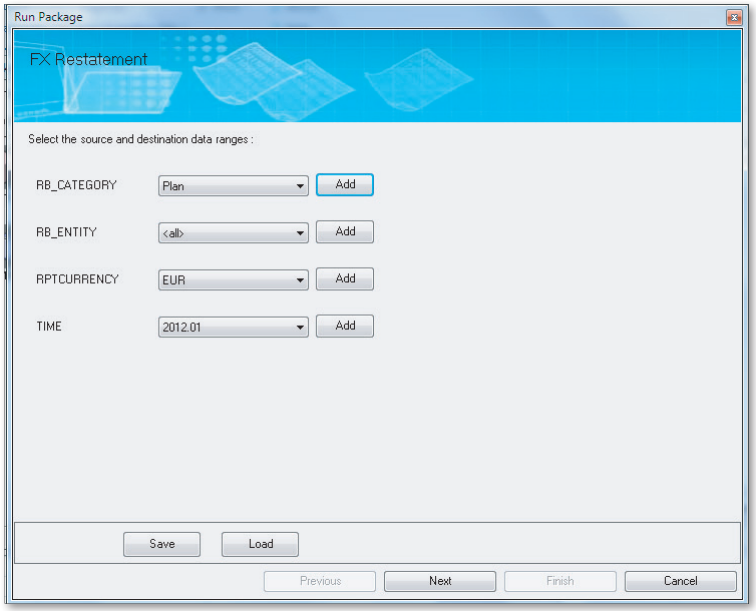
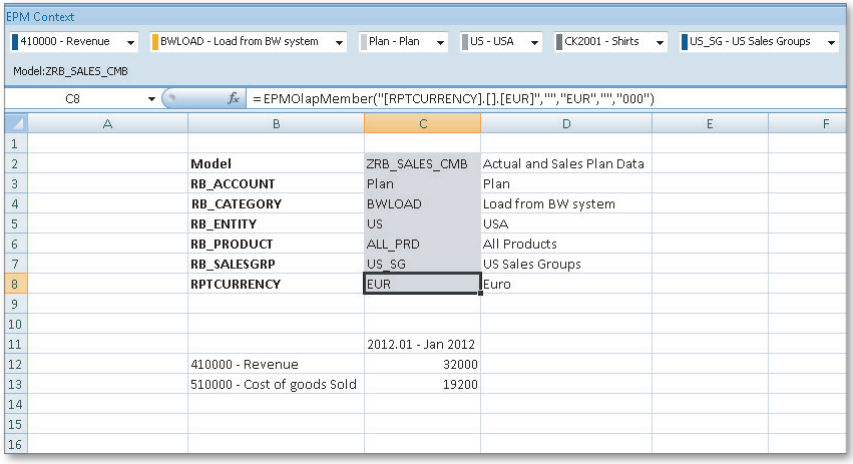


Figure 6.24 Performing Currency Translation—Part E

After you execute the data package, you'll see that the data from the local currency has been translated to EUR currency, based on the exchange rate specified in the RATES model (Figure 6.25).



	A	B	C	D	E	F
1						
2		**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data		
3		**RB_ACCOUNT**	Plan	Plan		
4		**RB_CATEGORY**	BWLOAD	Load from BW system		
5		**RB_ENTITY**	US	USA		
6		**RB_PRODUCT**	ALL_PRD	All Products		
7		**RB_SALESGRP**	US_SG	US Sales Groups		
8		**RPTCURRENCY**	EUR	Euro		
9						
10						
11			2012.01 - Jan 2012			
12		410000 - Revenue	32000			
13		510000 - Cost of goods Sold	19200			
14						
15						
16						

Figure 6.25 Performing Currency Translation—Part F

Note

Currency translation in SAP BPC 10.0 can be done using either a Data Manager package or consolidation monitor. We will cover the consolidation monitor in Chapter 9.

You now know how to translate financial transactions that are maintained in a local currency to reporting/group currency. In this section, you've seen the different types of business rules you can set up in SAP BPC. In the next section, we'll explain how you can perform allocations. Allocation is a powerful feature available to distribute data inside SAP BPC.

6.5 Allocations

Several business reasons drive the need for using allocation logic. Let's say, for example, that you have an organization with a corporate office and additional locations where it does business. When the corporate office incurs an expense, such as overhead, it has to be allocated to the other office locations based on criteria that depend on usage or work done by the corporate office for the other offices.

SAP BPC supports a robust interface for allocating data. The allocation definition is coded as a script file and can be called from a Data Manager package. The script for allocating data is based on a table interface for the developer to specify how the allocation should be performed. This process provides the developer with an intuitive interface through which he can satisfy the business requirements for allocation.

An allocation is always made up of the following components:

- What needs to be allocated
- Where the results of the allocation must be written
- What driver should be used to perform the allocation
- How the allocation driver should be used—that is, what factor must be applied to the source amounts during the allocation process

You should have a clear understanding of these components before you begin coding the allocation logic. When you know what needs to be allocated, you can use the following syntax to code the allocation:

```
*RUNALLOCATION
*FACTOR={expression}
```

```
*NAME={allocation name}
*APP [WHAT={app name};] [ WHERE={ app name };] [USING ={ app name }]
*DIM {dim name} WHAT={set}; WHERE={set};[USING ={set};] [TOTAL={set}]
*DIM …
*ENDALLOCATION
```

Now let's discuss the following instructions in the code: `*FACTOR` and `*DIM`. The `*FACTOR` instruction can be used to define any arithmetic expression (written in the `{expression}` parameter) and may contain operands, parentheses, constants, and one or both of the keywords `USING` and `TOTAL`, representing, respectively, the amount coming from the `USING` region (i.e., the amount of the driver) and the amount coming from the `TOTAL` region (i.e., the sum of the drivers):

```
*FACTOR=USING/TOTAL
```

Another keyword supported by this parameter is `COUNT`, which represents the number of members into which one amount must be allocated. For example, when evenly allocating a yearly value into all months of a year, the administrator may use just the `COUNT` keyword.

```
*FACTOR=1/COUNT
```

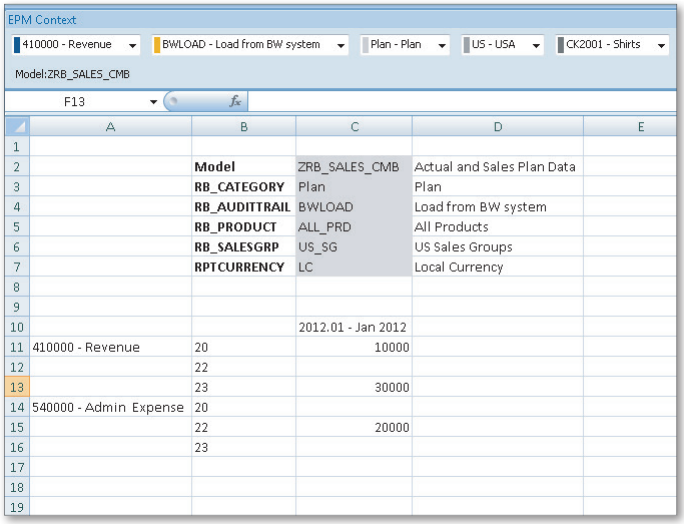
With the `*DIM` keyword, you can define the set of members that each dimension should read for each specific region of the allocation (the `WHAT`, the `WHERE`, the `USING`, and the `TOTAL` regions).

Using the same logic, if you had to allocate rent expenses incurred by a corporate entity called Admin to other entities based on the respective percentage usage, you would have to code the following in your logic script. Note that the percentage is maintained in the model for each entity.

```
*RUNALLOCATION
*FACTOR=USING/100
*DIM ENTITY WHAT=ADMIN; WHERE<>ADMIN; USING<>ADMIN
*DIM ACCOUNT WHAT=RENTAL; WHERE=RENTAL; USING=PERCENTAGE
*ENDALLOCATION
```

To demonstrate this concept, we'll perform allocations so that Rich Bloom can allocate the overhead expenses incurred at its corporate office (entity 22) to other entities in the United States, based on the revenue the company generated for the period.

1. Looking at the plan data for Jan 2012, Rich Bloom's corporate office (entity 22) has planned administrative expenses for $20,000 (Figure 6.26). This has to be allocated to the other entities in the United States, based on their total revenue for the same period.



	A	B	C	D	E
1					
2		**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data	
3		**RB_CATEGORY**	Plan	Plan	
4		**RB_AUDITTRAIL**	BWLOAD	Load from BW system	
5		**RB_PRODUCT**	ALL_PRD	All Products	
6		**RB_SALESGRP**	US_SG	US Sales Groups	
7		**RPTCURRENCY**	LC	Local Currency	
8					
9					
10			2012.01 - Jan 2012		
11	410000 - Revenue	20	10000		
12		22			
13		23	30000		
14	540000 - Admin Expense	20			
15		22	20000		
16		23			
17					
18					
19					

Figure 6.26 Performing Allocations—Part A

2. Enter the following code in the *ALLOCATION.LGF* script file for the model:

```
*RUNALLOCATION
*FACTOR = USING/TOTAL
*DIM RB_ACCOUNT WHAT=540000; WHERE=<<<; USING=410000; TOTAL=<<<;
*DIM RB_ENTITY WHAT=22; WHERE=>>>; USING=>>>; TOTAL=BAS(US);
*DIM RB_CATEGORY WHAT=Plan; WHERE=<<<; USING=<<<; TOTAL=<<<;
*DIM TIME WHAT=2012.01; WHERE=<<<; USING=<<<; TOTAL=<<<;
*DIM RB_PRODUCT WHAT=PRN; WHERE=<<<;
*DIM RB_SALESGRP WHAT=SGN; WHERE=<<<;
*ENDALLOCATION
```

Save and validate the script file (Figure 6.27). The logic allocates the overhead data for the period Jan 2012 from entity 22 to the other entities in the United States, based on their sales revenue for that period.

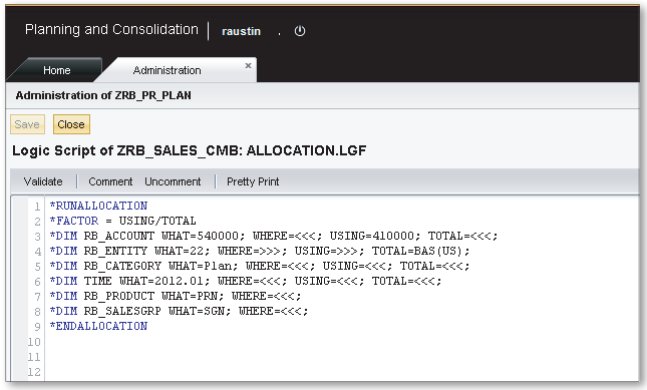


Figure 6.27 Performing Allocations—Part B

You're now ready to execute the ALLOCATION Data Manager package (Figure 6.28).

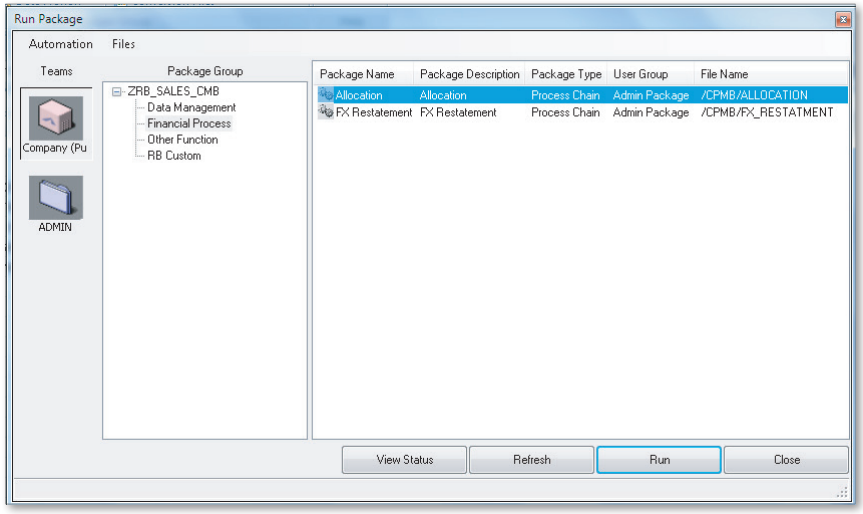
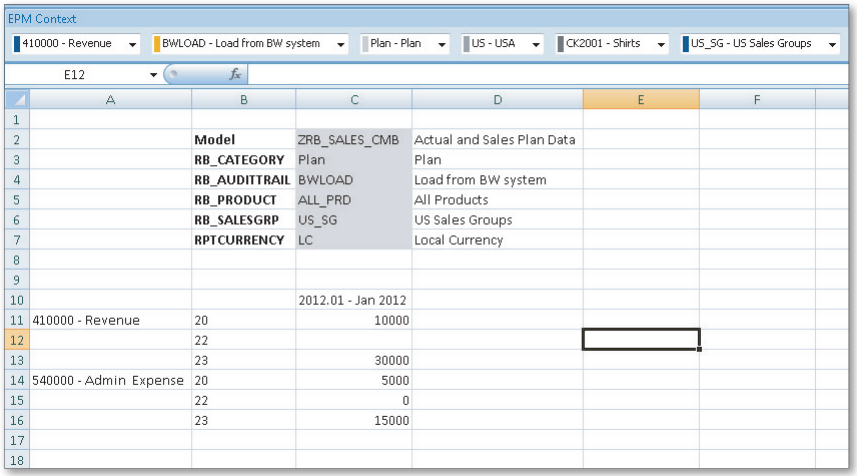


Figure 6.28 Performing Allocations—Part C

3. After executing the package, run the report to see how the allocation logic has worked (Figure 6.29). The overheads have been allocated from the corporate entity to the other entities, based on revenues.



	A	B	C	D	E	F
1						
2		**Model**	ZRB_SALES_CMB	Actual and Sales Plan Data		
3		**RB_CATEGORY**	Plan	Plan		
4		**RB_AUDITTRAIL**	BWLOAD	Load from BW system		
5		**RB_PRODUCT**	ALL_PRD	All Products		
6		**RB_SALESGRP**	US_SG	US Sales Groups		
7		**RPTCURRENCY**	LC	Local Currency		
8						
9						
10			2012.01 - Jan 2012			
11	410000 - Revenue	20	10000			
12		22				
13		23	30000			
14	540000 - Admin Expense	20	5000			
15		22	0			
16		23	15000			
17						
18						

Figure 6.29 Performing Allocations—Part D

Note

The allocation process does not normally zero-out the amounts from the source accounts, but in this example, the source account was set to zero because of the driver that was used to perform the allocation. We allocated admin expense to all base entities in the United States based on each entity's revenue, and because the corporate entity did not have any revenue, the allocation process calculated the admin expense account of the corporate entity to zero.

In this section, you've seen how to perform allocation in SAP BPC. In the next section, we'll discuss areas where you can write ABAP code to meet specific business requirements.

6.6 Using BAdIs to Code Logic

Some business processes require the use of additional logic to complement the existing functionality in the SAP BPC framework. You can use SAP Business Add-Ins (BAdIs) for this purpose and create them using ABAP programming language in the SAP NetWeaver BW system. The rich language constructs in ABAP make it excellent for use in SAP BPC models. ABAP may also provide performance benefits when used in certain business scenarios.

There are different types of BAdIs available in SAP BPC for NetWeaver, and they serve distinct requirements. They are referred to as enhancement spots. You define a BAdI for SAP BPC for NetWeaver in the BW system. The following enhancement spots are available to enhance the logic to meet a custom requirement:

- `UJD_ROUTINE`
  This enhancement spot provides the ability to enhance the master and transaction data coming from a flat file or SAP NetWeaver BW system. You can reference a BAdI in the transformation file when executing a Data Manager package. The reference to the BAdI implementation is made in the `STARTROUTINE` or `ENDROUTINE` in the Options section of the transformation.
- `UJ_CUSTOM_LOGIC`
  A BAdI implementation created using this enhancement spot can be referenced in the logic script. A Data Manager package that uses a logic script file can be used to execute the code in the BAdI. This is used, for example, when custom logic is needed to copy data from one model to another model in the same or a different environment.
- `UJR_WRITE_BACK`
  The write-back BAdI is called before the data is written to the database. This BAdI is useful when it is necessary to disaggregate the data coming from an input form in top-down planning. For example, when data is planned at a product group level—and if that data needs to be broken down at the product level based on some rules—then this BAdI can serve that purpose.
- `UJD_RETRACT`
  This BAdI can effectively be used to retract the data in SAP BPC back to the SAP ERP system.
- `UJ_VALIDATION_RULE_LOGIC`
  This BAdI can enforce specific rules on data that is updated into a model.
- `UJQ_SHARED_QUERY`
  This BAdI is used to filter or modify the data that is displayed in a report.
- `UJJ_CALCULATED_AMOUNT`
  This BAdI is used to perform calculations or add data when using journals.

You'll need a basic understanding of object-oriented ABAP before you can develop BAdIs in SAP BPC models, but for now, we'll just provide the high-level steps to create a BAdI. An ABAP programmer in your organization will be able to help in this area.

1. Use Transaction SE19 to create a BAdI, which should be based on one of the enhancement spots (e.g., `UJ_CUSTOM_LOGIC`) and is provided for use in SAP BPC models. You're prompted to create a transport when you create a custom BAdI.
2. Specify the filter values for the BAdI implementation. These are the values that passed to the BAdI.
3. The `UJ_CUSTOM_LOGIC~EXECUTE` method is automatically created for the BAdI because it is based on the `UJ_CUSTOM_LOGIC` enhancement. Enter the code for the BAdI here. The internal Table CT_TABLE contains the data passed from the SAP BPC model, which can be modified based on the business requirement.
4. Save and activate the objects created.
5. Create a script file in SAP BPC to call the BAdI you've created. Refer to the following example:

   ```
   Syntax:
   *START_BADI <BADI_Name>
   <key1> = <value1>
   <key2> = <value2>
   ..
   *END_BADI
   ```

   In this example, `<BADI_Name>` is the name used in the BAdI. You can also pass the additional parameters `<key1>` and `<key2>`.

   The execution of the BAdI performs a default query and automatically writes back to the model. Set the `QUERY` parameter to `OFF` if you want to perform your own read inside the BAdI implementation; set `WRITE` to `OFF` to turn off the automatic write back of query results. You may code to write to another model from the BAdI.
6. Create a Data Manager package and assign process chain `/CPMB/DEFAULT_FORMULAS` to the package. This Data Manager will be used to run the logic script described in the last step. In the Data Manager package, adjust the prompts for the script to be executed.
7. You're now ready to execute the Data Manager package. This will execute the package and prompt for any parameters you specified in the BAdI.

You now know the steps for developing logic using BAdIs in SAP BPC; you can also use BAdIs to write logic in `Start Routine` and `End Routine` that are associated with data loading. `Start Routine` is logic that is executed before transformation is

executed during data loading. `End Routine` is logic that is executed after transformation is executed during data loading.

First, create a BAdI implementation for `UJD_ROUTINE`. Develop the logic in separate filters—one for `Start Routine` and another for `End Routine`. In the OPTIONS section of the transformation file associated with a data load, specify the filters of the BAdI that contain the logic as follows:

```
STARTROUTINE=<BAdI filter name1>
ENDROUTINE=<BAdI filter name2>
```

SAP SCN offers users a how-to guide for creating a BAdI for SAP BPC in the SAP NetWeaver BW system, although the guide is not supported by SAP. The how-to guide's URL is *http://scn.sap.com/docs/DOC-4165*.

In the next section, we'll discuss rules you can configure to ensure data accuracy.

6.7 Validation

With the introduction of SAP BPC for NetWeaver, a powerful feature is available to enforce rules and to prevent incorrect records from being entered or updated in the system. This ensures the accuracy of data in an SAP BPC model. The validation is used to validate combinations of values of dimensions. For example, we can set up a validation rule to ensure that corporate expenses are only posted to corporate entity dimension members.

The following rules apply to validation:

- Validation is performed in the SAP NetWeaver BW system using Transaction UJ_VALIDATION. You'll need access to this transaction to access this interface.
- Only one dimension can be marked as the Driver dimension.
- Any dimension in a model can be specified as the Driver dimension for validation. Different models in a given environment can use any dimension as the Driver dimension.
- In the planning and consolidation models, the Account dimension is usually marked as the Driver dimension.
- Rules are defined so that only those records that meet the criteria can be saved to the model. Every time a record is updated for the model, including a Driver dimension, the rules are checked before the data is updated.

- When data is updated in SAP BPC, if the data does not comply with the rule defined, the record is rejected, and an error message displayed.
- If the same dimension is used as the Driver dimension by more than one model in the environment, any rules created for that Driver dimension apply to all of the models that share it.
- You can turn off the validation's functionality for a model using Transaction UJ_VALIDATION. When the validation is turned off, the rules for the model are not checked when data is updated. Validations can be set to apply to journals, manual updates, and Data Manager packages.
- Setting up validation rules does not require knowledge of ABAP, but the BAdI implementation is available for setting up advanced logic using code. This provides additional ABAP functionality when a standard configuration for validation does not fully meet the requirement or when the process is complex enough to require the use of BAdIs.
- Validation rules configured in the development system can be transported across the landscape.

We'll now look at an example of setting up validation rules in the SAP NetWeaver BW system:

1. Log in to the SAP NetWeaver BW system, and execute Transaction UJ_VALIDATION. Identify the environment for which you want to set validation rules, and then click on Assign Driver Dimension (Figure 6.30, ❶ and ❷).

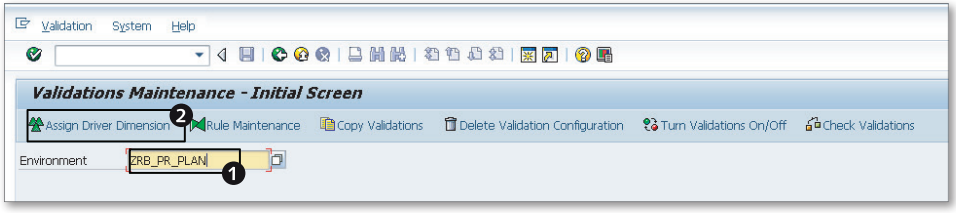


Figure 6.30 Creating a Validation Rule—Part A

2. All of the models associated with the environment are displayed. Enter the dimension for the model for which you want to set the Driver dimension, and click on Save Driver Dimension Settings (Figure 6.31, ❸ and ❹). In our example, we're setting RB_ACCOUNT as the Driver dimension for our planning model.

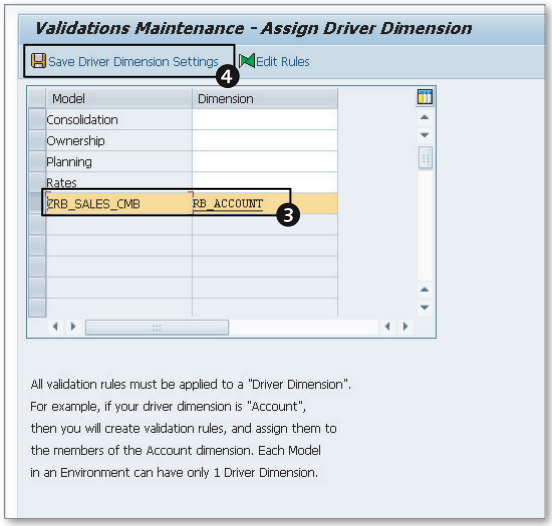


Figure 6.31 Creating a Validation Rule—Part B

3. After you have saved the Driver dimension settings, click on the back arrow. Here, you can create a new rule. Click on CREATE RULE to create a new validation definition (Figure 6.32, ❺).

Figure 6.32 Creating a Validation Rule—Part C

4. We will define validation for admin expense account 540000 and then allow administrative expenses to be posted only to the corporate entity (entity 22). Select 540000 as the member of the Account dimension (Figure 6.33, ❻). Then click on ADD DIMENSION (Figure 6.33, ❼). Enter the following values in the first row for the columns:
   - DIMENSION: "RB_ENTITY"
   - OPERATOR: "="
   - MEMBERS: "22"

This defines the member value for the entity that can post values for this account. Entity 22 is the corporate office of Rich Bloom under which corporate administrative expenses are recorded (Figure 6.33, ❽). Other entities should not be able to charge corporate admin expense amounts.

5. Click on SAVE RULE, and click on the back arrow to go back to the previous screen (Figure 6.33, ❾).
6. You'll now see the validation rule you created (Figure 6.34).

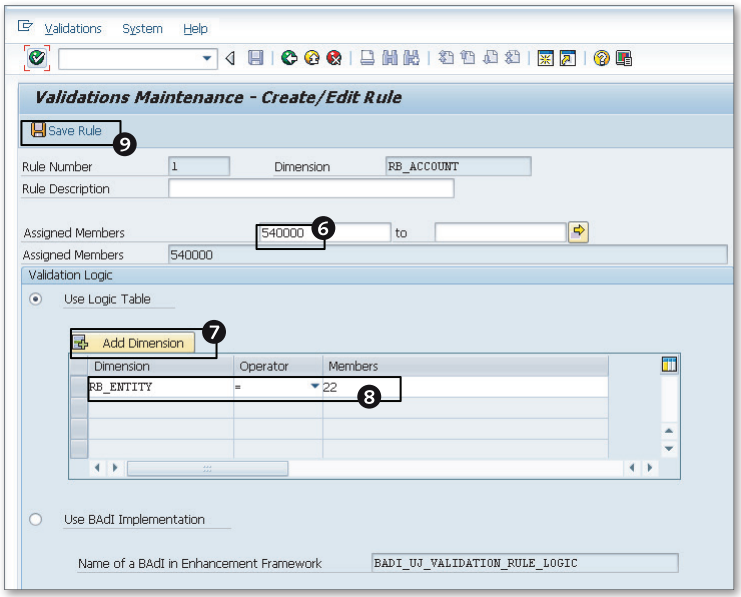


Figure 6.33 Creating a Validation Rule—Part D

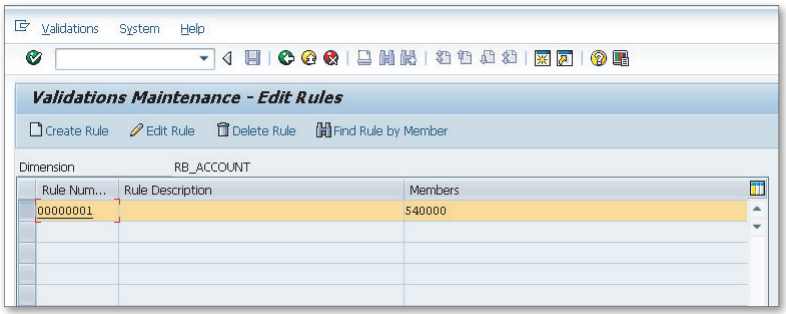


Figure 6.34 Creating a Validation Rule—Part E

7. Click the green arrow to go back to the initial screen of Transaction UJ_VALIDATION, and click on the turn validations On/Off buttons to activate the validation rules (Figure 6.35, ⑩).

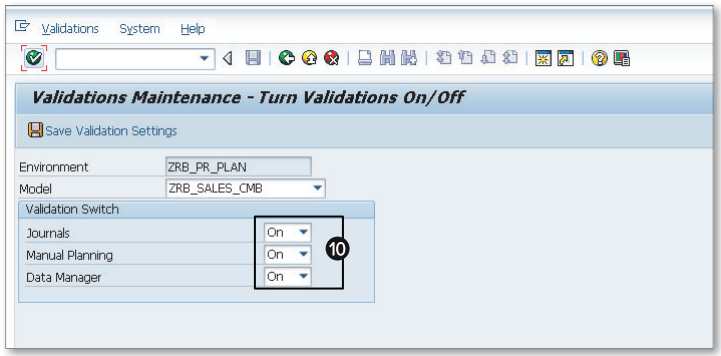


Figure 6.35 Creating a Validation Rule—Part F

8. Create an input schedule. Select 20 as the entity and 540000 (Admin Expense) as the account member. Enter a value for 540000 (Admin Expense), and send the data to the database (Figure 6.36). The system should disallow this transaction based on the validation rule created earlier.

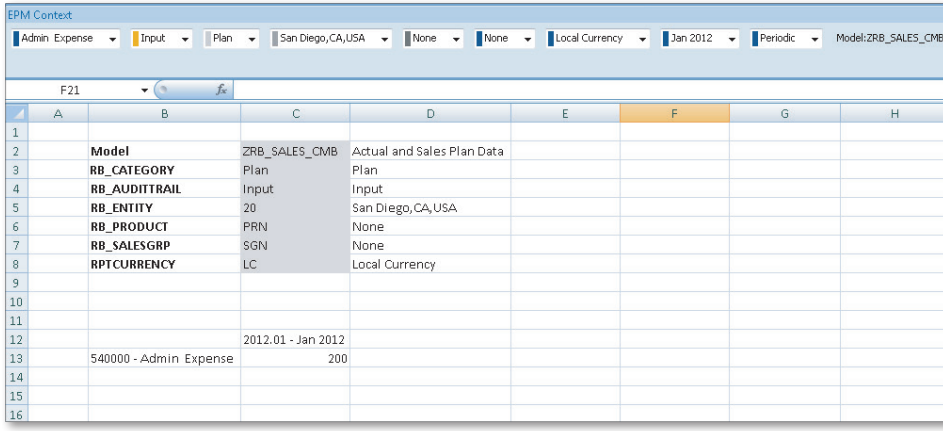


Figure 6.36 Creating a Validation Rule—Part G

You'll receive an error message saying that the data failed the validation rule because the admin expense can only be posted to corporate entity 22 (Figure 6.37).

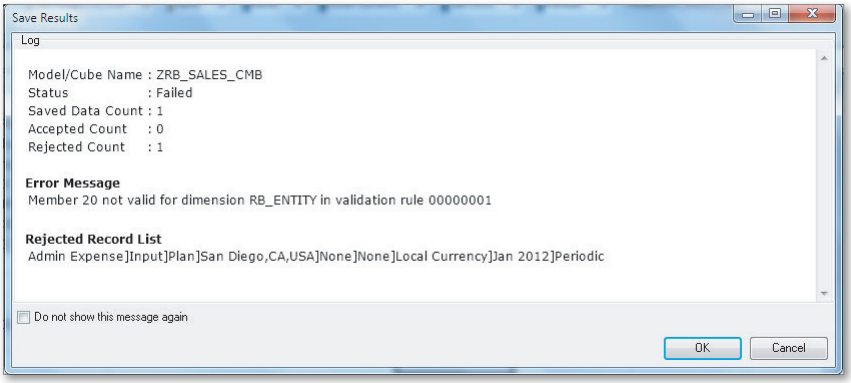


Figure 6.37 Creating a Validation Rule—Part H

You now know how to set up validation rules in the SAP NetWeaver BW system. This powerful functionality enforces data integrity in SAP BPC applications. Next, we will understand the controls business rule which is also a validation mechanism to ensure consistency and accuracy of data.

6.8 Controls

Controls is the name of a new rule type introduced in the SAP BPC 10.0 version for NetWeaver. It replaces the validations functionality that was available under business rules in the SAP BPC 7.x versions. Controls provide enhanced features to confirm the consistency and accuracy of data in a model and are more enhanced than the validations functionality from prior versions.

Controls are enabled at the model level and ensure that the data inside a model is reliable. As consolidation data is used for external reporting, controls are used to verify that the data maintained is accurate and suitable for reporting. For example, a control can be defined to check that the total of the assets equals the liabilities plus the owner's equity. Controls can also be used in planning to check that the data meets the business requirements of the model.

We will look at an example of how a control is defined and used inside a model in Chapter 9, when we discuss consolidating data inside an SAP BPC model.

6.9 Summary

In this chapter, you learned about the options available to define logic in an SAP BPC model. First, you saw how to define formulas to automatically calculate values inside a dimension. Then, you learned about logic script and how it provides a rich collection of statements to handle complex logic inside models.

We then discussed business rules to execute common planning and consolidation tasks and explained how to use them in place of developing detailed logic script. We showed an example of performing currency translation using the RATES model. We also showed how to perform allocations based on certain drivers.

Next we explored how BAdIs can be used in SAP BPC for NetWeaver. This is especially useful for coding complex business requirements using the ABAP programming language.

We also saw how to define validation rules in the SAP NetWeaver BW system to ensure data accuracy.

Finally, we saw the new controls feature introduced in SAP BPC 10.0 for NetWeaver and how it provides enhanced features to ensure the consistency of data maintained in a model.

In the next chapter, we'll review the tools available in SAP BPC to support collaboration. Specifically, we'll review the *work status*, which you use to monitor the planning and consolidation process and protect data from further changes. We'll also discuss options available to post comments, publish books, distribute and collect data, and develop a user interface using menus.

Process management and collaboration are key requirements for users of planning and consolidation applications in order to perform their tasks efficiently. The SAP BPC tool allows users to collaborate with the other users in the system and facilitates the dissemination of information to other users who are outside of BPC.

7 Process Management and Collaboration

This chapter introduces the collaboration tools and features available in SAP Business Planning and Consolidation (SAP BPC) that facilitate the sharing and exchange of data. The objective of using these tools is to promote team dynamics and to enable users to make informed decisions on different business aspects. Understanding the implementation of work status, adding and viewing comments, distributing data in reports and input templates, and publishing books can facilitate the planning and consolidation process.

Section 7.1 discusses how to configure work status to monitor and restrict changes to the data in a model. In particular, we'll explain how you can use work status in a scenario where the current plan data needs to be locked against further changes.

Section 7.2 explains how to add and view comments in SAP BPC. Comments can provide context to information and share perception and reasoning behind information in a model.

Section 7.3 explains how you can distribute data or input forms in SAP BPC to users as offline reports and how these reports can then be uploaded back into SAP BPC after users make changes to them.

Section 7.4 explains how to use the new feature introduced in SAP BPC 10.0 for NetWeaver to publish books. The ability to create books is a powerful feature to publish one or more reports with varied selections as a single book for analysis.

SAP BPC can be used not only with Excel but also with PowerPoint and Word. Section 7.5 shows the steps involved in integrating SAP BPC data into these documents.

We'll begin this chapter by explaining the use of work status and how to configure it to restrict changes to the data in a model.

7.1 Work Status

In SAP BPC, work status serves the important function of protecting changes to data stored in a model. It enables a subset of data—also referred to as a region of data—to be assigned a work status and helps you monitor the status of the data throughout the planning cycle.

Work status can be set for any region of data, based on the dimensions selected for maintaining work status for a model. (If you're familiar with SAP planning tools such as SAP NetWeaver Business Planning and Simulation and SAP NetWeaver BW Integrated Planning, you may have heard the term *data slices* used in those environments to describe the same thing.)

Work status helps you meet the objective of protecting and monitoring data for several reasons. It gives an organization more control over its data and defines the tasks that each user can perform. Additionally, it differs from standard data access in SAP BPC in that, after the work status is set as locked, even users who have write access to the data are restricted from performing data updates. For example, the data can be restricted from being updated by anyone in the organization after the plan data is approved by management.

For each environment, you can define different work statuses, such as unlocked, submitted, approved, and so on, depending on the process flow the organization uses. All of the models inside an environment use the work status defined for the environment. You can also create a new work status, reorder the work status, edit the description of the work status, or delete a work status for an environment.

Work status can be applied to a wide range of activities: Data Manager (DM), journals (JRN), manual entry (MAN), comments (COMM), and documents (DOCS). It applies to a set of data in a model, which is called a data region. For each model in the environment, you configure a set of dimensions to use for setting the work status. Only the dimensions selected in this configuration can be used to define the data region for setting the work status.

To use work status, you have to select a minimum of three and a maximum of five dimensions in each model. When you're defining the dimensions for work

status in a model, the system requires one dimension to be specified as the Owner dimension, which should include the OWNER property. The members listed in this dimension should be updated with a user ID or team ID in the OWNER property so that the IDs can be used for setting work status.

You can select a region of data for setting the work status. For example, if you want the plan data for all of 2012 to be locked from any changes in a model, you can set this region of data as locked.

For a combination of work status and activity, you can define the following types of access:

- All: Any user who has write access to the data region for the model can update data for the data region.
- Locked: No user can update data for the data region.
- Owner: Only the user ID or team ID listed under the OWNER property of the Owner dimension can update the data for the data region.
- Manager: Only the user ID or team ID associated with a manager can update the data for the data region. Hierarchy of the Owner dimension is used to identify the manager. The owner associated with a node in the hierarchy is the implied manager of owners associated with children of the node.

You can set multiple owners as values in the OWNER property of a dimension; be sure to separate owner names with a comma. The owner name can be a user ID or team ID. The user ID or team ID specified in the OWNER property of a dimension should be a valid user ID or team ID listed for that environment. When you specify a team ID, enclose it as [<TEAMID] where <TEAMID> is the team ID.

SAP BPC for NetWeaver supports selecting the Account dimension as the Owner dimension for work status settings. Work status also lets you monitor updates to unstructured data such as documents that are associated to a model.

Administrators at both the environment and model levels set work status configuration, so administrators should be familiar with the business process and the work status steps to perform this role. Administrators define the work statuses and the types of users who can update data for a given work status. This serves the purpose of defining the users who can update data for a given work status. Changes to the definition of work status (say, list of dimensions to administer work status) remove any existing work statuses set by users for all models in the environment.

Before you can use the dimension identified as the Owner dimension to set work status, you have to specify a hierarchy for it. This hierarchy is specified when configuring the work status for a model. The hierarchy is used to identify implied managers of entities. The owner associated to a node in the hierarchy is the implied manager of owners associated with children of the node. When data is written back to an SAP BPC model the work status is checked for each record that is updated. If, for example, one of the records in the update failed the work status check, that record is rejected. The other records are updated successfully.

The EPM function `EPMWorkStatus` in the EPM add-in interface retrieves the work status for a set of dimension member values in a model.

You can turn on the feature to send email notification when there is a change to a work status. The set-up for email functioning in the SAP NetWeaver BW system is a prerequisite for this feature to be used.

So let's take a look at a common scenario that requires the use of work status, using our sample company. In this scenario, the sales managers of Rich Bloom enter the sales plan for the next year, which is then submitted to management for approval. After the plan has been approved, it should be protected from changes by anyone other than the owner of the entity, as follows:

1. Connect to the ZRB_PR_PLAN environment, and click on the Work Status view under the Features domain in the Administration workspace (Figure 7.1, ❶).
2. In the upper portion of the view, you'll see the work statuses in the rows and the activities in the columns. The intersection identifies who can update the data for the combination of work status and activity (Figure 7.1, ❷). The work status defined here applies to all of the models in the environment.
3. Above the work statuses displayed, you can see the menu options to maintain work status for an environment (Figure 7.1, ❸). You can also create a new work status using the options listed here.
4. Click on Approved work status to set who can update data for an activity (Figure 7.1, ❹). Select Owner for Manual Entry (MAN) to ensure that only the owner of an entity can modify the data when the work status is set to Approved (Figure 7.2, ❺).

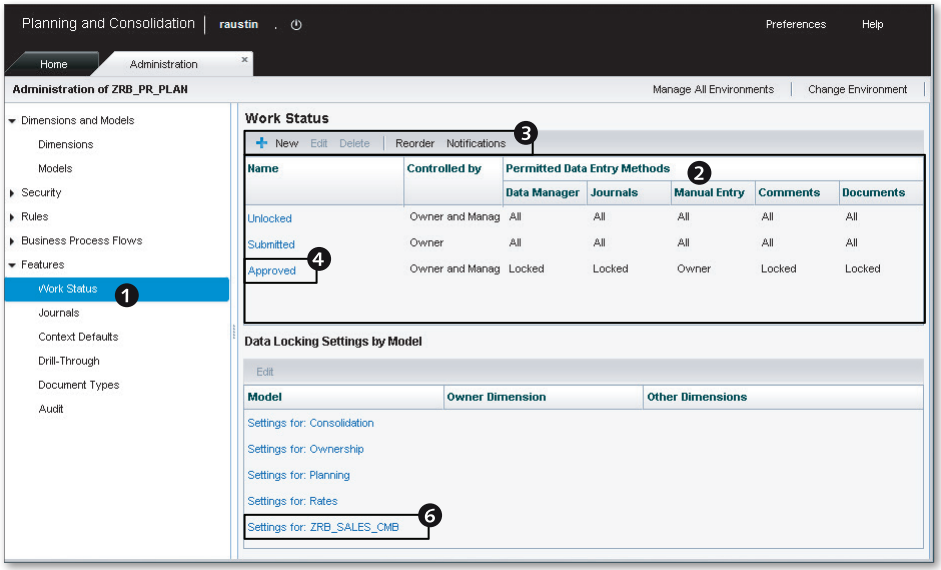


Figure 7.1 Configuring Work Status to Lock Data—Part A

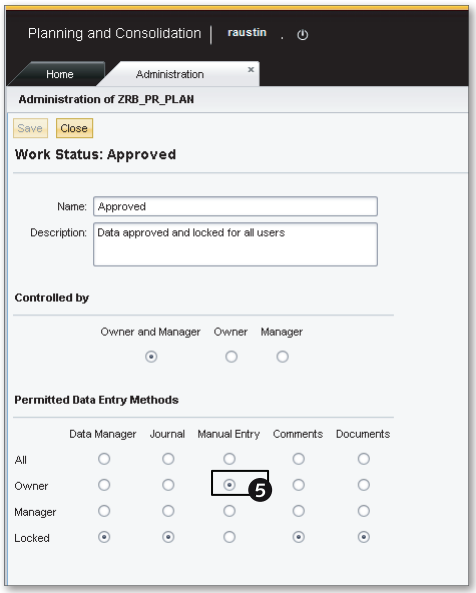


Figure 7.2 Configuring Work Status to Lock Data—Part B

5. You will now set the work status of a model. Click on the sales model, ZRB_SALES_CMB, and identify the Category, Entity, and Time dimensions as the dimensions to be used for setting work status for the model (refer back to Figure 7.1, ❻, and Figure 7.3). Check Enable Data Locking to enable the work status for the model. The hierarchy to use for the Owner dimension is set here. In this case, hierarchy H1 of the RB_ENTITY dimension is used to determine the implied manager of entities. Remember that you can select a maximum of five dimensions to set work status, and you are required to set one dimension as the Owner dimension, which should include an `OWNER` property. The property ID of the dimension should read "OWNER". We selected the Entity dimension as the Owner dimension in our example.

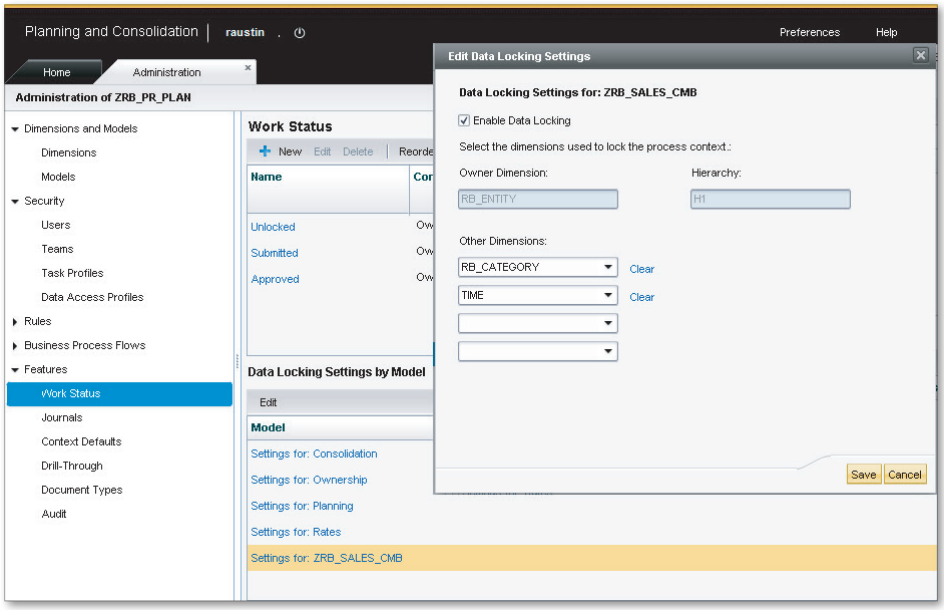


Figure 7.3 Configuring Work Status to Lock Data—Part C

6. Select the RB_ENTITY dimension under Dimensions in the left pane, and enter the owner value for the members in the dimension. The OWNER column identifies who the owner of the entity is. In our example, SHILL (Shawn Hill) is identified as the owner (Figure 7.4, ❼). You can also enter a team in the OWNER column or multiple owners separated by commas.

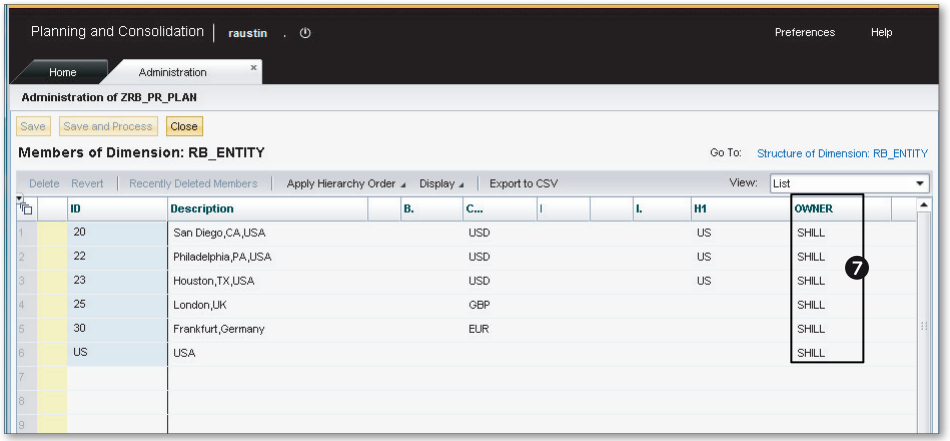


Figure 7.4 Configuring Work Status to Lock Data—Part D

7. Let's say you have different entity members that fall under a higher-level hierarchy node. The OWNER property value associated with the higher-level entity node is the implied manager of the entities that fall under the node. Work status uses the hierarchy set in the EDIT DATA LOCKING SETTINGS dialog box to identify the owner and manager of the Owner dimension.
8. From the EPM add-in for Excel interface, connect to the ZRB_SALES_CMB model in the ZRB_PR_PLAN environment.
9. Select the CHANGE WORK STATUS menu option under the SAVE DATA command in the EPM tab to choose the member values for specifying the work status, and make the selections shown in Figure 7.5 in the EPM – CHANGE WORK STATUS dialog box as follows:
   - RB_ENTITY: "USA"
   - RB_CATEGORY: "Plan"
   - TIME: "Feb 2012"
   - NEW STATUS: APPROVED
10. Include all of the descendants for the entity member selected by selecting INCLUDE ALL DESCENDANTS (Figure 7.5, ❽). This will apply the work status selection for all of the descendants of the entity member selected. Select APPROVED as the work status, and click on APPLY. You will see a confirmation that the work status has been set to APPROVED for the selections.

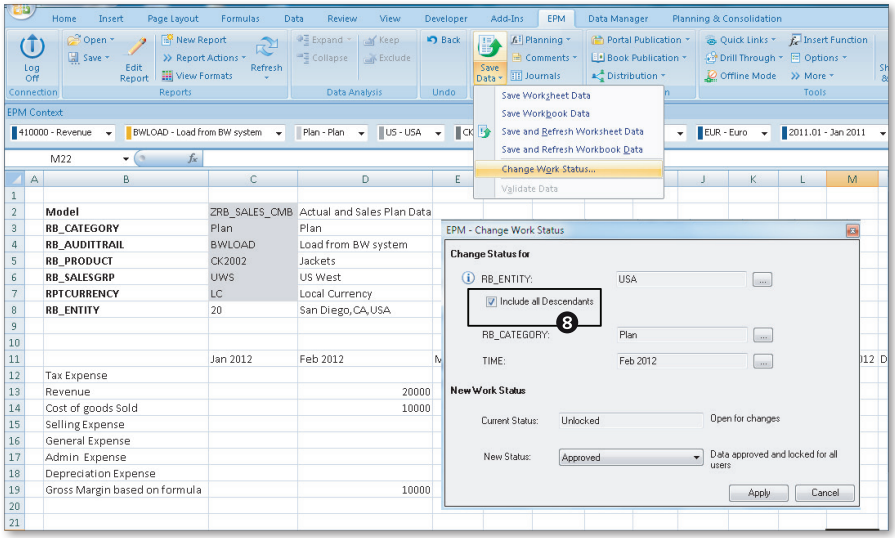


Figure 7.5 Configuring Work Status to Lock Data—Part E

11. Next, we will test to make sure that the work status setting that we just configured works as desired. Create an input form, and enter or modify plan data for entity 20 (San Diego, CA) and Feb 2012 period (Figure 7.6, ❾). Then send the data to the database by selecting the Save Worksheet menu option under the Save Data command.

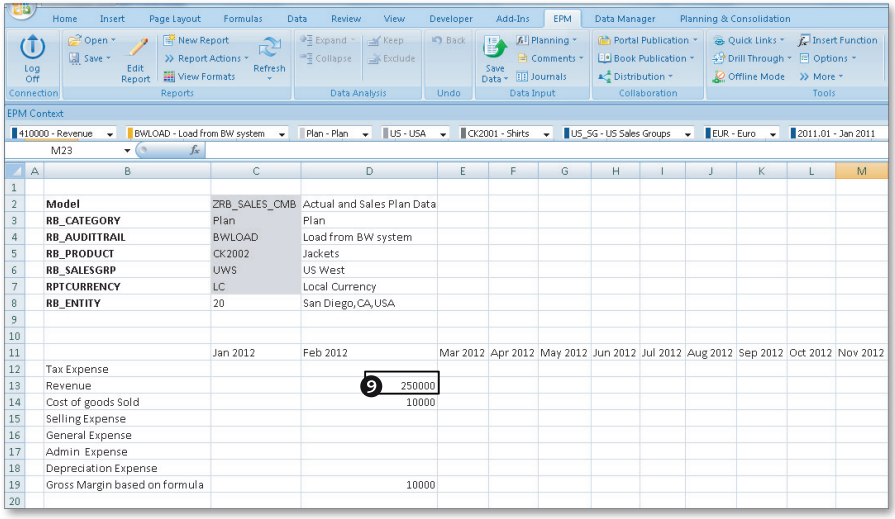


Figure 7.6 Configuring Work Status to Lock Data—Part F

You'll notice that the update fails with an error message indicating that the work status check has failed. Because the work status is APPROVED, only the owner of the entity, Shawn Hill, can modify the data (Figure 7.7). In this example, because we are logged in as user RAUSTIN, the updated failed.

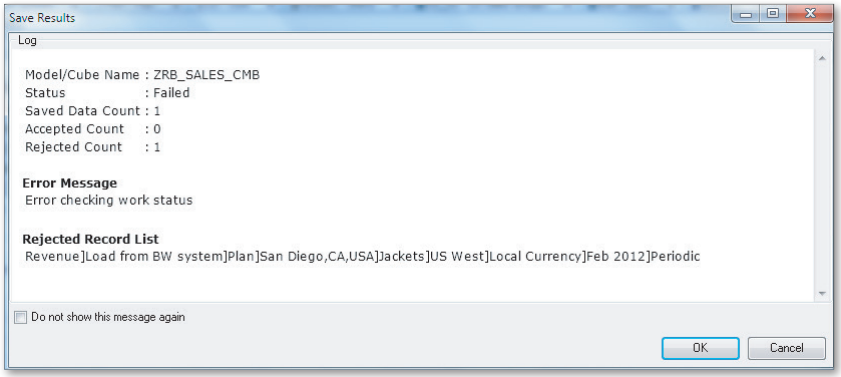


Figure 7.7 Configuring Work Status to Lock Data—Part G

In this section, we explained how to define work status types and select the dimensions to be used when assigning work status for a model. We also reviewed the steps for setting the work status for a data region and explained how to use work status to lock data from being changed in an SAP BPC model.

In the next section, we'll discuss how to use another collaboration tool, comments, in SAP BPC.

7.2 Comments

Comments play the important role in SAP BPC of enabling users to enter free-form text about the data in an application. Information users enter as comments can be viewed by others and aid in decision making. Let's take a look at some important features of comments.

You can access, add, and search comments from the EPM add-in for Excel, Word, and PowerPoint interface; users should have authorization to add, view, or delete comments.

The checkbox to use comments for a model is enabled in the Models view under the Dimensions in the model domain in the Administration workspace of the Web Client.

You can set a comment for a specific selection of dimension members from the SAP EPM add-in for Excel interface.

A comment can be up to 1,332 characters. The length of the comment is defined in the Model view under the Dimensions in the model domain in the Administration workspace of the Web Client. When you create a comment, you can assign a keyword and priority to it. You can then search for comments using either partial or full criteria based on these two parameters. You can also run a report to get a detailed list of comments by history.

In the EPM add-in for Excel interface, the `EPMSaveComment` function sends comments to a database for a specific set of member values. The `EPMCommentFullContext` and `EPMCommentPartialContext` functions retrieve comments based on a specific set of member values. We will discuss these functions in this section.

7.2.1 Adding Comments

There are two ways to add comments from the EPM add-in for Excel interface. You can add a comment either manually or use the EPM function `EPMSaveComment` for a set of dimension member values. Follow these steps to enter a comment manually:

1. In the SAP BPC Web Client, connect to the ZRB_PR_PLAN environment, and click on the Models view of the Dimensions in the model domain in the Planning and Consolidation workspace. Select the ZRB_SALES_CMB model, and click the Edit menu option. Set the checkbox to Enable Comments for the model, and set the length of the comment (Figure 7.8, ❶ and ❷).

2. To add a comment manually, open a report or input form from the EPM add-in for Excel interface, and connect to the ZRB_SALES_CMB model. Click on the Add Comment menu option under the Comments command (Figure 7.9).

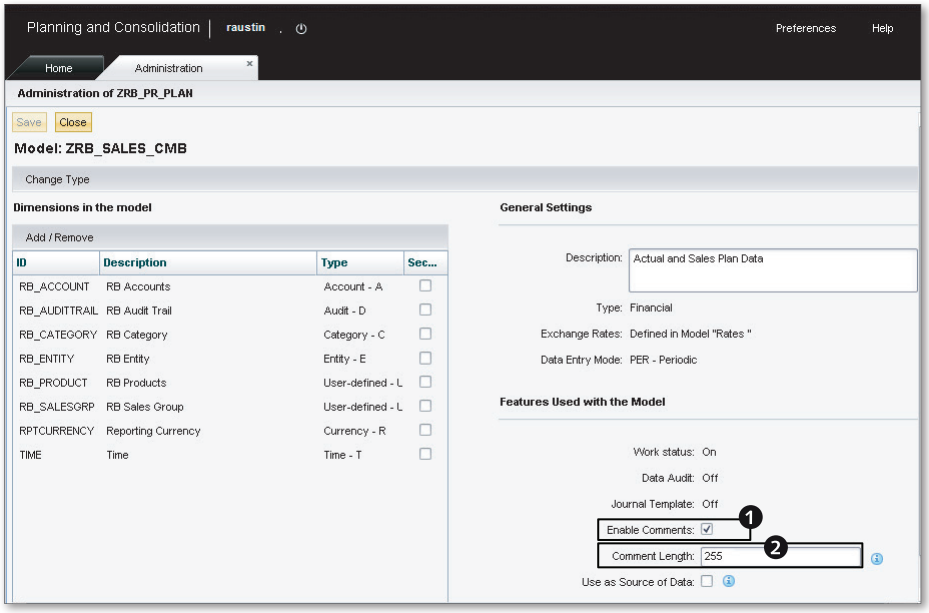


Figure 7.8 Adding Comments—Part A

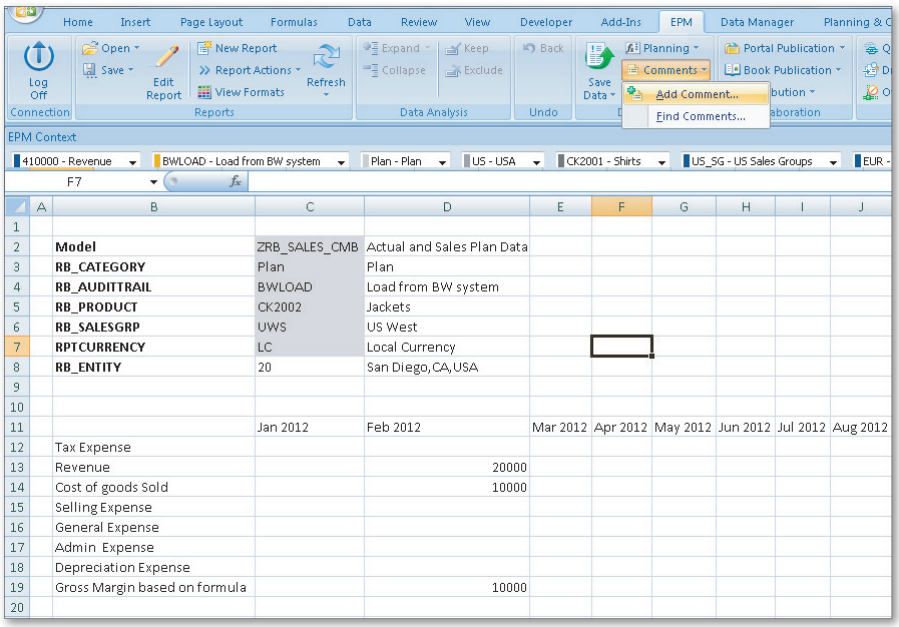


Figure 7.9 Adding Comments—Part B

The system displays a dialog box, EPM – Add Comment, with options to create a comment (Figure 7.10).

3. (Optional) Select the Priority of the comment: High, Medium, Low, Critical or No Priority. Select High from the list in the dropdown box. Setting the priority will allow us to retrieve the comments later by searching comments based on priority.
4. Enter a comment description. For our example, we have entered the description "We are approving the plan data for 2012".
5. (Optional) Select a keyword for the comment so you can retrieve the comments later by searching comments based on keyword. We have used the keyword "PLAN2012_APP".

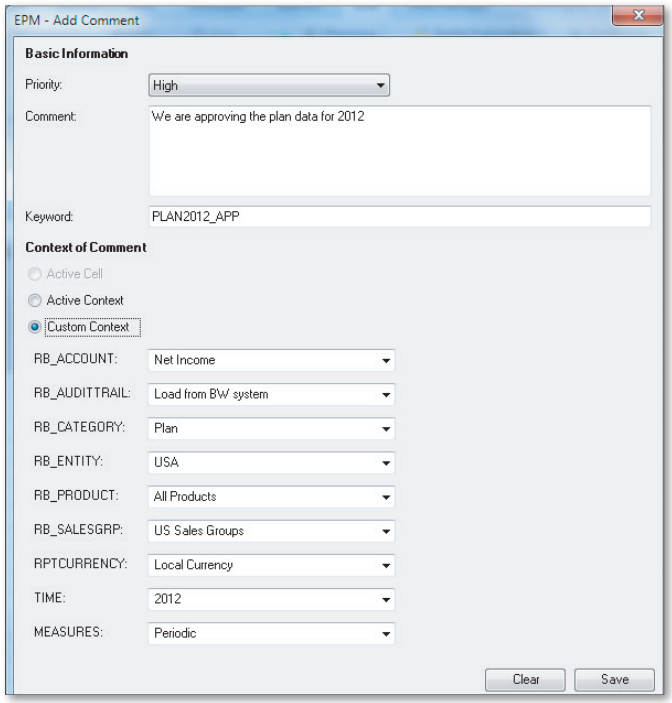


Figure 7.10 Adding Comments—Part C

6. Select one of three options for Context of Comment:
   - Active Cell: You can select this option if a report or input form is open. Otherwise, this option is grayed out.

- Active Context: You can select this option when the comment is associated with the values defined in the context view.
- Custom Context: You can select this option if you want to create a comment based on a custom selection. If this option is selected, the system prompts you to select the dimension member values to which the comment should apply. If you do not want to use a dimension for your comment, select the value None for that dimension.

For our example, select Custom Context as the Context of Comment. Select the following member values for entering the comment:

- RB_ACCOUNT: Net Income
- RB_AUDITTRAIL: Load from BW system
- RB_CATEGORY: Plan
- RB_ENTITY: USA
- RB_PRODUCT: All Products
- RB_SALESGRP: US Sales Groups
- RPTCURENCY: Local Currency
- TIME: 2012
- MEASURES: Periodic

Then click on the Save button.

We've now added a comment for a combination of dimension member values. Next, we'll explain how to view comments.

7.2.2 Viewing Comments

Follow these steps to view comments:

1. Open a report or input form from the EPM add-in for Excel interface. Connect to the ZRB_SALES_CMB model. Click on the Find Comment menu option under the Comments command.
2. The dialog box EPM - find Comments is displayed (Figure 7.11). You can search for comments based on selections defined here.
3. Select a Keyword for the comment. This is useful when you know the keyword used when creating the comment. You can also use a partial keyword to search for comments.

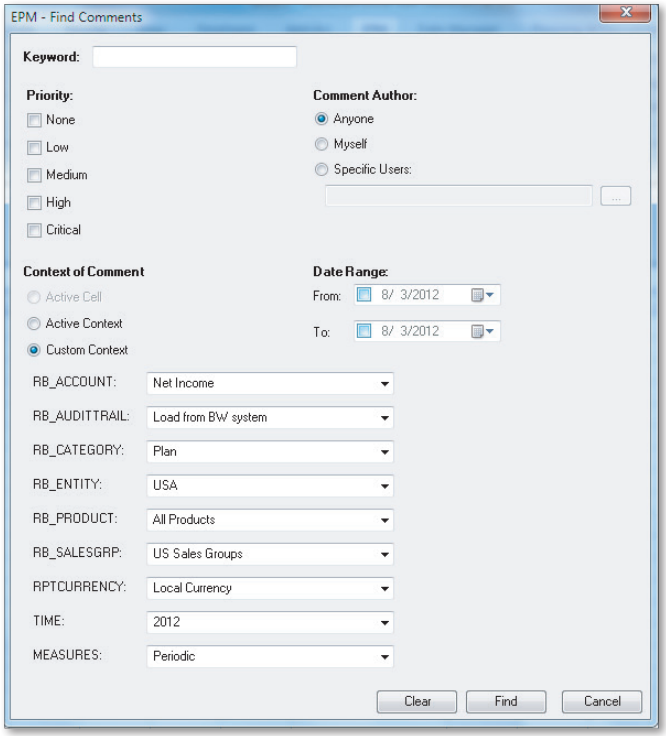


Figure 7.11 Viewing a Comment—Part A

4. Select the priority level of the comments you want to view (either NONE, LOW, MEDIUM, HIGH, OR CRITICAL).
5. Select the context view of the comment under CONTEXT OF COMMENT. There are three options available to select the context of a comment:
   - ACTIVE CELL: You can select this option if a report or input form is open. Otherwise, this option is grayed out.
   - ACTIVE CONTEXT: You can select this option for the system to return a comment associated with the selections in the context area of the EPM pane.
   - CUSTOM CONTEXT: You can select this option if you want to view a comment based on dimension values that are different from what is shown in the context area. If this option is selected, the system prompts you to select the dimension values to retrieve the comments. If you want to include all of the values for a dimension or if you had selected NONE for a dimension member when creating a comment, select the value ALL for that dimension.

For our example, select Custom Context. Select the following member values for entering the comment:

- RB_ACCOUNT: Net Income
- RB_AUDITTRAIL: Load from BW system
- RB_CATEGORY: Plan
- RB_ENTITY: USA
- RB_PRODUCT: All Products
- RB_SALESGRP: US Sales Groups
- RPTCURENCY: Local Currency
- TIME: 2012
- MEASURES: Periodic

6. Select the origin for comments from one of the following options:
   - Anyone: This selection returns all comments that were entered by all users. For our example, use this option.
   - Myself: This selection returns only the comments you entered (based on your user ID).
   - Specific Users: This selection returns comments added by a specific user. When you select this option, you have to select the particular users whose comments you want to view.
7. You can specify a date criteria if you want to search comments created based on a date range. When the date criteria is entered, it restricts the comments displayed to the date specified.
8. After making your selections, click on the Find button, which launches the Manage Comments workspace in the Web Client and displays the comments for the selections (Figure 7.12).

   The Manage Comments workspace displays the comments for the selection. In addition, it provides the menu options to manage the comments displayed. You can edit and delete a comment, display the history of a comment, or specify a new criteria for displaying the comment. You can highlight a comment and select the desired menu option. When you are editing a comment, the comment description, priority, and keyword can be changed.

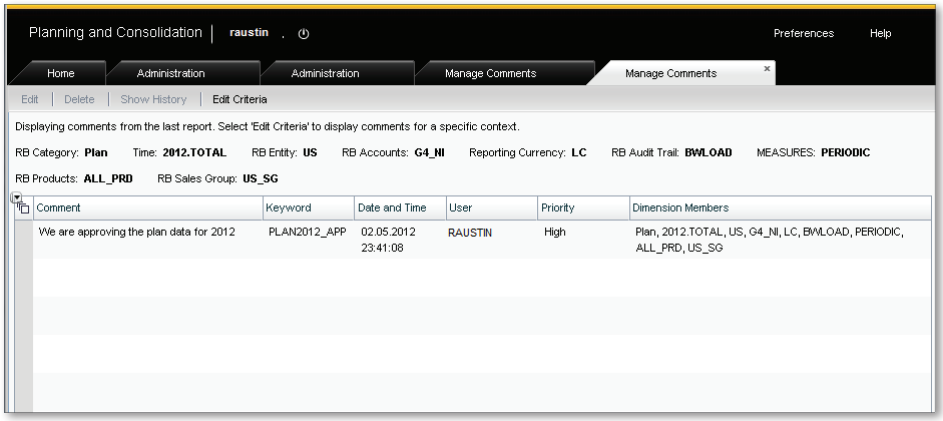


Figure 7.12 Viewing a Comment—Part B

You now know how to add, view, and maintain comments from the EPM add-in interface. In the next section, we'll explain how to report on comments.

7.2.3 Reporting on Comments

You can also view comments by selecting the COMMENTS link from the AUDIT view in the HOME workspace inside the SAP BPC Web Client. The menu option to run comment reports is located as a link under the GENERAL section. Click on the COMMENTS link (Figure 7.13, ❶). The option to specify the criteria for retrieving comments opens in a new workspace (Figure 7.14).

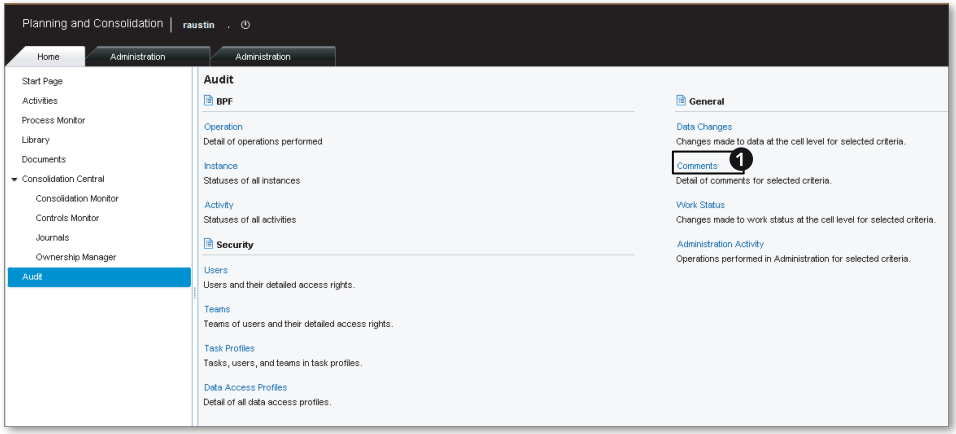


Figure 7.13 Reporting a Comment—Part A

After you select the model you want to see comments for from the Models drop-down list, you can generate a report of comments based on the following selections:

- Dimension Members
  Specify the dimension member values for which you want to display comments. Select All Members for a dimension member when you want all of the comments for a dimension to be reported or if you had selected None when selecting a value for the dimension when creating the comment.
- History
  In this section, select Only the Most Recent Comments or, alternatively, Full History of Comments if you want the report to include not only the latest comment but also a history of all of the comments for a selection.
- Date and Time
  This section enables you to select all of the comments or comments created based on a start and end date. The two options are Anytime and Start Time/End Time.
  - Anytime: Select this option when you want to report comments irrespective of the time when it was created.
  - Start Time: Specify the start date and time when the comment was created. A selection here restricts the comments created after the start time.
  - End Time: When you specify the start time, you can specify the end time for comments to be included between the Start Time/End Time.
- Additional Criteria
  This area offers the options to select a comment based on the priority of the comment, the user, and the keyword:
  - Priority: Select the priority of comments to be included in the report.
  - User: If you want the report to include comments for only a particular user, you can specify the user's ID.
  - Keyword: Select the keyword of the comment to be included in the report.

After you make your selections, click on the Display Report button to display the comments for the selections (Figure 7.14, ❷).

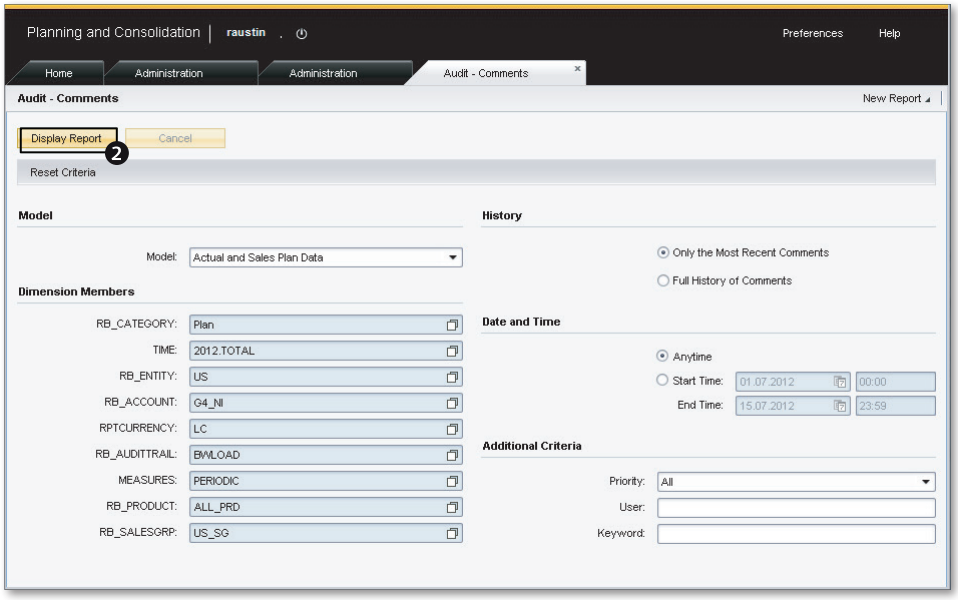


Figure 7.14 Reporting a Comment—Part B

The comments are output for the selections in Figure 7.15. Here, you can use the EDIT CRITERIA button to define a new selection for displaying comments.

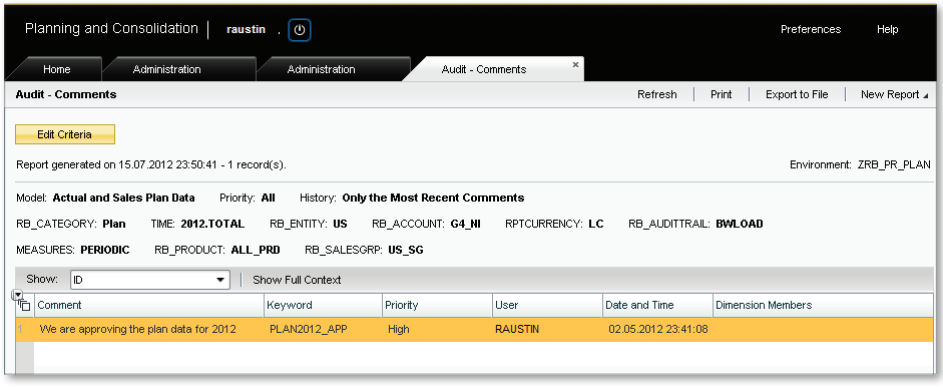


Figure 7.15 Reporting a Comment—Part C

You now know how to display comments. Next, we'll review some of the EPM functions that are available to maintain and view comments.

7.2.4 EPM Functions for Comments

You can use the following standard EPM functions in SAP BPC for Excel to create, update, or display comments:

- `EPMSaveComment`
  You can use the `EPMSaveComment` function to send a comment to the database. The values specified in this function should reference a connection name and dimension values with which this comment value must be associated. If a member value is not specified, the value for the dimension is automatically taken from the context view of a report or schedule.
- `EPMCommentFullContext`
  A comment with full qualified context is one that has all dimension member values specified. In other words, the comment does not have None for any dimension member value. You can use the `EPMCommentFullContext` function to retrieve a comment from the database for a model and dimension member values. If a dimension member value is not specified in the function to retrieve comments, the value for the dimension is automatically taken from the context view of the report or form (e.g., `EPMCommentFullContext(<Connection Name>)` or `EPMCommentFullContext(<Connection Name>,<dim_val1>,<dim_val2>,…)`).
- `EPMCommentPartialContext`
  Comments with partial qualified context have one or more dimension members with values equal to None. You can use the `EPMCommentPartialContext` function to retrieve a comment from the database for a model and partial set of dimension member values (e.g., `EPMCommentPartialContext(<Connection name>,<dim_val2>,<dim_val3>…)`).

You should now know how to add and view comments, including how to display comments using the SAP BPC Web Client. In the next section, we'll explain how distribution and collection is used to distribute reports and input forms. We'll also explain how to use the Distribution and Collection interface to update data from an offline input schedule into SAP BPC.

7.3 Distribution and Collection

The Distribution and Collection interface in SAP BPC allows a user to share the data in SAP BPC with other users who may or may not have access to the system.

It also enables the data to be entered in offline mode and be reloaded back into SAP BPC.

You can access the options to distribute the data in SAP BPC and collect the data via the EPM add-in Excel interface. The menu options are available under the Distribution command in the EPM tab. You must have specific tasks granted in your task profile to use this functionality.

The process of sending a report or input form in offline mode is referred to as *distribution* in SAP BPC. The data included in this report can be used by others who may need access to this information for analysis and decision making. Sometimes users who travel may not be able to connect to SAP BPC to access information in a model; distribution facilitates the process of getting data to the necessary users via email and aids in collaboration.

A user can update the data in an input form in offline mode and push the changes back to the SAP BPC system, which is referred to as *collection*.

The report or input form generated using the distribution list is a snapshot of the data at a specific point in time. When a user sends a report or an input form using a distributor, the data can be placed in a folder or sent as an email to users. If the report or input form is sent via email, the Simple Mail Transfer Protocol (SMTP) server settings for email routing should be entered in the Server Configuration tab of User Options in the EPM tab in Excel.

The process of defining reports or input forms for distribution is managed using distribution templates. You can specify more than one report or input form when defining the distribution templates; if you're sending data as an email message, you can list the users who should receive the report and what data should be sent.

After a distribution template is created, it can be executed to generate the reports and input forms defined there. A template can be run immediately or scheduled in the background. This job is executed on the machine of the user who schedules the job.

When data is generated using the distribution option, the access available for the user who schedules the distribution governs the content of the report or input forms. Ensure that the user executing the distribution list has sufficient member access to the data requested in the distribution list. If the user does not have access

to the data, the distribution template fails to execute and returns a message saying the user does not have sufficient authorization to run the reports. A report or input schedule generated using distribution can be viewed directly in Excel; there is no need to use any of the interfaces.

When a file is created or sent as an email after the generation of reports or input forms using the distribution process, anyone who has access to that file or email can view the generated data because the information is offline at that time. This has to be taken into consideration when data is distributed using this process, especially if the generated data is sensitive and should be kept confidential.

Three primary interfaces are used to distribute and collect data:

- Template: This interface is used to create and maintain distribution templates. The list identifies the reports, forms, and dimension member selections.
- Distribute: This interface is used to schedule execution of a distribution template. The job can be scheduled to run immediately, or it can be scheduled as a future job to be run later.
- Collection: This interface is used only for input forms that were distributed earlier. With this interface, data entered or modified in an offline input form can be updated back into a model in SAP BPC.

We'll now explain how to create a distribution template using an input form.

7.3.1 Creating a Distribution Template

Before we begin creating a distribution template using the input form that is displayed in Figure 7.16, recall that we created this input form in Chapter 5.

1. To create a distribution template, from the EPM add-in interface for Excel, connect to the ZRB_SALES_CMB model in the ZRB_PR_PLAN environment. The Edit Book and Distribution Templates, Use Offline Distribution, and Use Offline Collection security tasks provide access to a user to maintain, distribute, and collect data
2. From the EPM tab of the Excel interface, select the New Template menu option under the Distribution menu (Figure 7.16, ❶).

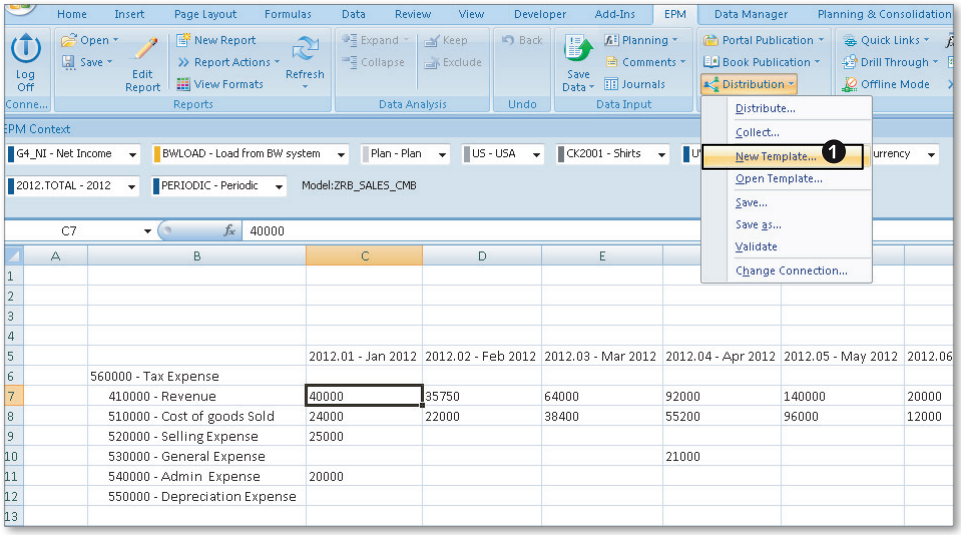


Figure 7.16 Creating a Distribution Template—Part A

3. This opens a distribution template (Figure 7.17), which contains a placeholder to enter the name of a template. This applies to the entire distribution template. A distribution template can include multiple sections, and each section in turn can include multiple reports or input forms. Enter the following parameters for the distribution template:

   - Template Name
     Every distribution template has a name assigned to it. Specify the name of the template (Figure 7.17, ❷).
   - Variable Time
     The time member selected here overrides the value in the context pane and applies the time to all of the sections in the template (Figure 7.17, ❸).
   - Section Name
     A distribution template can contain one or more sections, and a section in turn can include one or more reports and input forms. Enter the name of the section here (Figure 7.17, ❹).
   - Report Workbook
     In this parameter, specify the report or input form you want to use in a section of the distribution template. You can select a report or input form from the SAP BPC server folder or the local desktop (Figure 7.17, ❺ and ❻).

- Workbook Description
  Enter a description for the name of the workbook selected. For every report included in the description, it is mandatory to specify a description (Figure 7.17, ❼).

- Multiple Reports
  Multiple reports can be added to a section by clicking on this icon, which creates a new line to specify a new report and workbook description (Figure 7.17, ❾).

- Select Dimensions
  Click the link to specify the fixed members and variable members for the reports included in the section (Figure 7.17, ❽, and Figure 7.18).

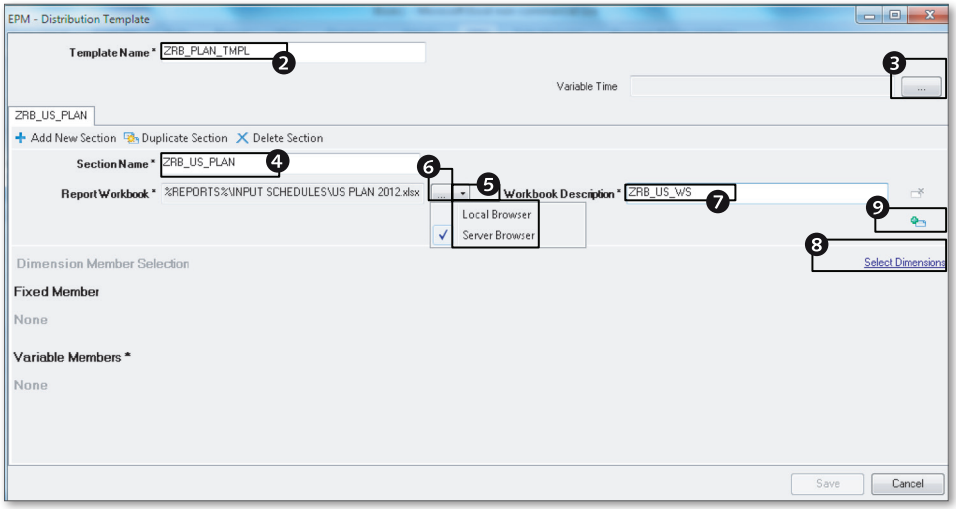


Figure 7.17 Creating a Distribution Template—Part B

- Fixed Member
  This field specifies fixed filters for a template section; specify the static selections. You can include Fixed Member values for the different dimensions in your report; for example, we have selected the Plan category. However, you can specify only one member value as a Fixed Member for a dimension. The field is mapped to the Fixkey field in the template.

- Variable Members
  Select the dimension values to use to execute the report. You can specify more than one dimension in this field. If multiple values are selected for a

dimension, multiple reports are created when executing the report. You can specify a relationship of the dimension members to include in the report. The report is generated corresponding to the authorized dimension members of the user. In the example shown in Figure 7.19, a report will be generated for each entity that is under the USA hierarchy. The field is mapped to the VARKEYS field in the template.

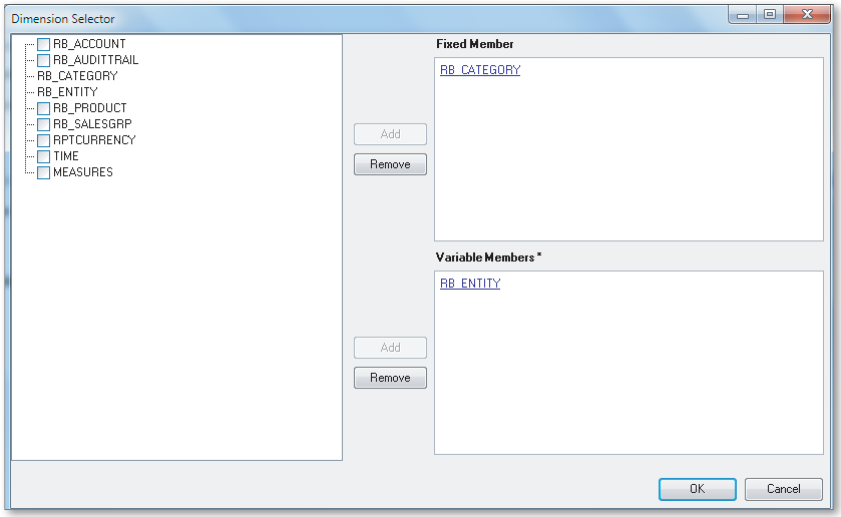


Figure 7.18 Creating a Distribution Template—Part C

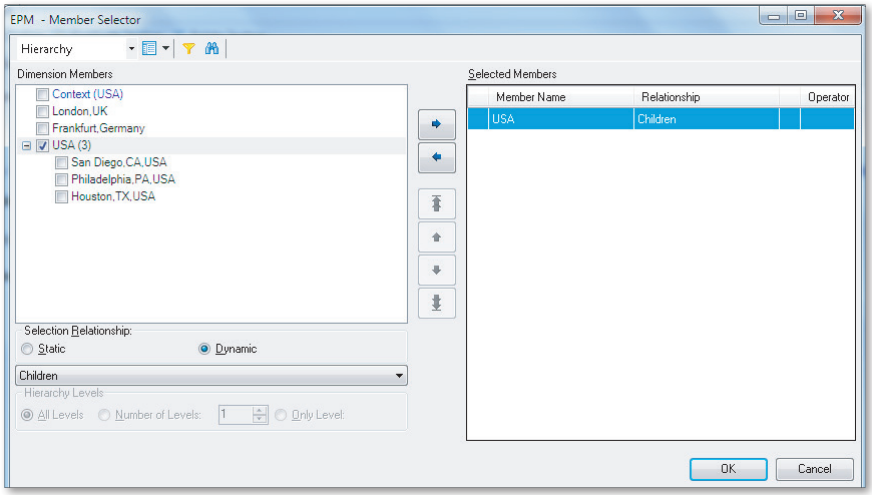


Figure 7.19 Creating a Distribution Template—Part D

- User
  Specify the user(s) for the report. This is useful when it is required to email the reports to a user. The SMTP server settings for email routing should be entered in the Server Configuration tab of User Options in the EPM tab of Excel.

4. The template after the selections have been made is shown in Figure 7.20. Save the distribution template. Select the Save menu option under the Distribution command to save the template. You can edit the template for further changes by selecting the Edit menu option under the Distribution command.

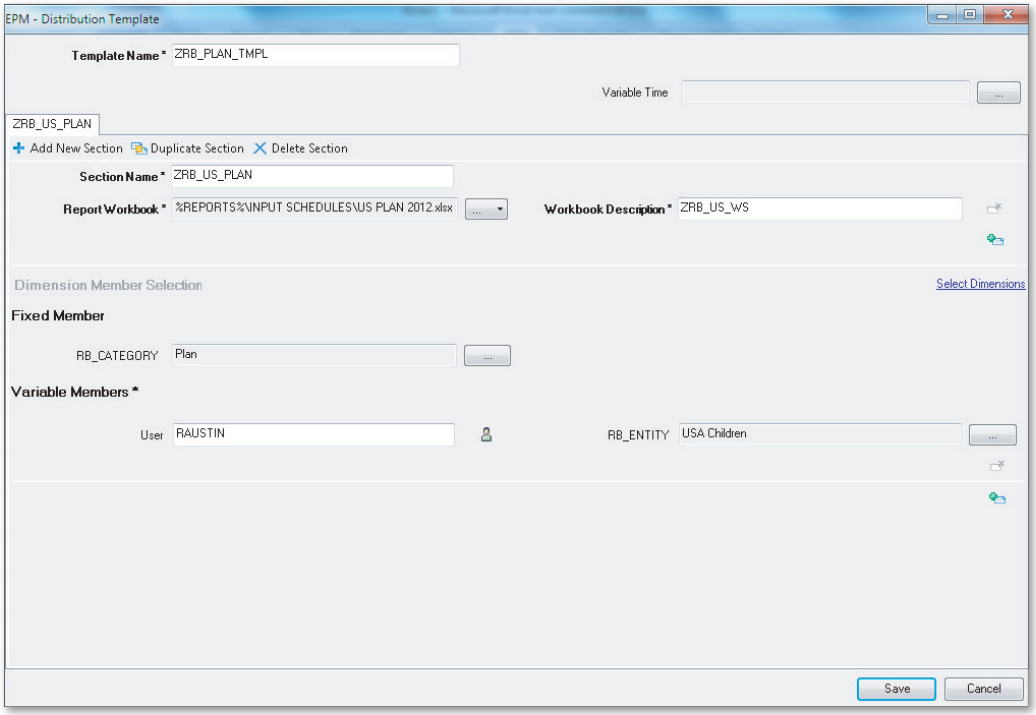


Figure 7.20 Creating a Distribution Template—Part E

Note

If you do not specify a dimension available in your model in the Fixed Member or Variable Members fields, the values for the dimension are taken from the context view of the user who is scheduling the report.

5. Validate the distribution template you just created. Open the template using the Open Server Root Folder menu option under the Open menu, and select the template (Figure 7.21). To validate the distribution list, click on the Validate menu option under the Distribution menu. When the distribution template is validated, the system checks to see whether all mandatory fields are entered.

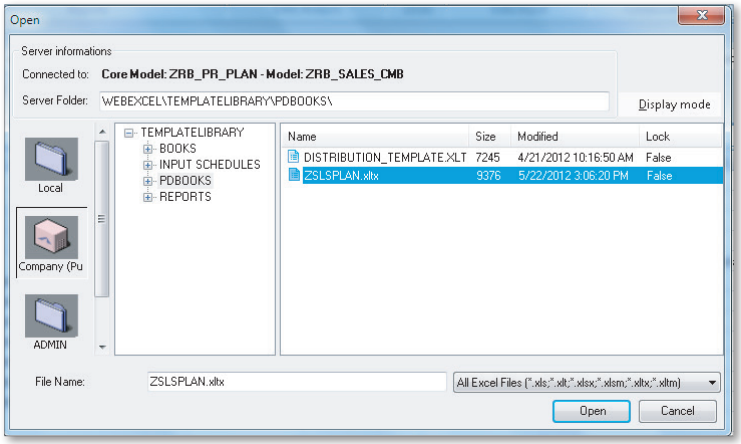


Figure 7.21 Creating a Distribution Template—Part F

Now that we've created a distribution list and validated it, we can execute and distribute it as an offline report or input schedule.

7.3.2 Offline Distribution Wizard

Before using the Offline Distribution Wizard, ensure that you have the necessary access. You'll need access to the Use Offline Distribution task that falls under Collaboration.

1. In the EPM tab of the Excel add-in interface of SAP BPC, select the Distribute menu option under the Distribution menu. You're prompted to select a distribution template for distribution.
2. Select the type of distribution from the dialog screen shown in Figure 7.22. There are two options to distribute the template: Distribute Now and Schedule the Distribution. When the Distribute Now option is checked, the distribution is scheduled immediately. When the Schedule the Distribution option is

checked, additional parameters are prompted to schedule the distribution. Scheduling a distribution results in a windows task being scheduled on the user desktop, so it requires the Windows password. The desktop should be on and connected to the network when the scheduled job is executed. Click on NEXT to proceed with the distribution.

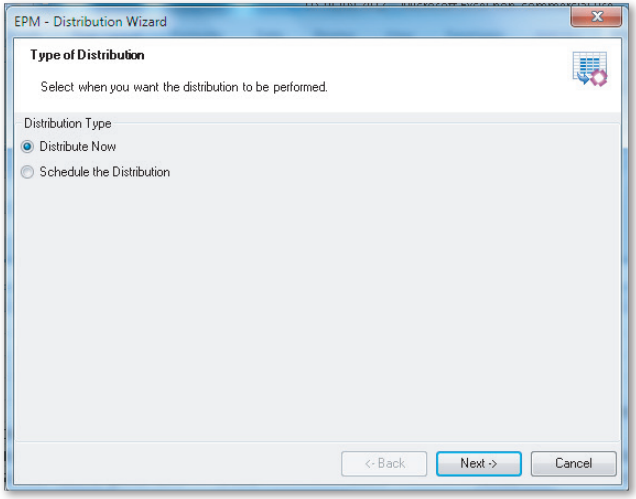


Figure 7.22 Offline Distribution Wizard—Part A

3. The output mode is specified irrespective of the type of distribution selected. One of the following output modes can be selected (Figure 7.23):

   - SWITCH TO OFFLINE MODE AND SEND BY EMAIL
     When this option is selected, the system prompts you to select a distribution template and then requires you to specify the sender, subject, and body details of the email message. Additional options can be set if you want to receive confirmation that the message was received or if you want to send the report as a ZIP file.
   - SWITCH TO OFFLINE MODE AND SAVE TO FOLDER
     When this option is selected, the system prompts the user to select a folder location in the local desktop where the reports or schedules selected in the distribution template should be saved.

   Click on NEXT to proceed with the distribution.

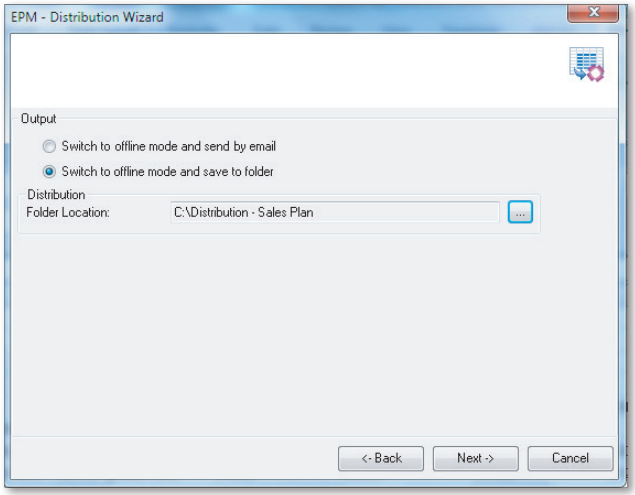


Figure 7.23 Offline Distribution Wizard—Part B

4. You are prompted to select the distribution template and identify the sections to distribute. More than one distribution template can be selected and distributed at once. After making the selections, click on the NEXT button (Figure 7.24).

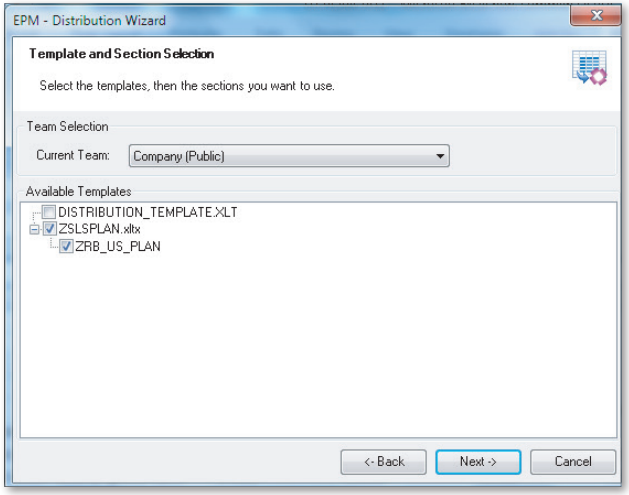


Figure 7.24 Offline Distribution Wizard—Part C

5. The next dialog box confirms the reports that will be created and the selections that will be used in creating the reports (Figure 7.25). Click on the NEXT button to continue.

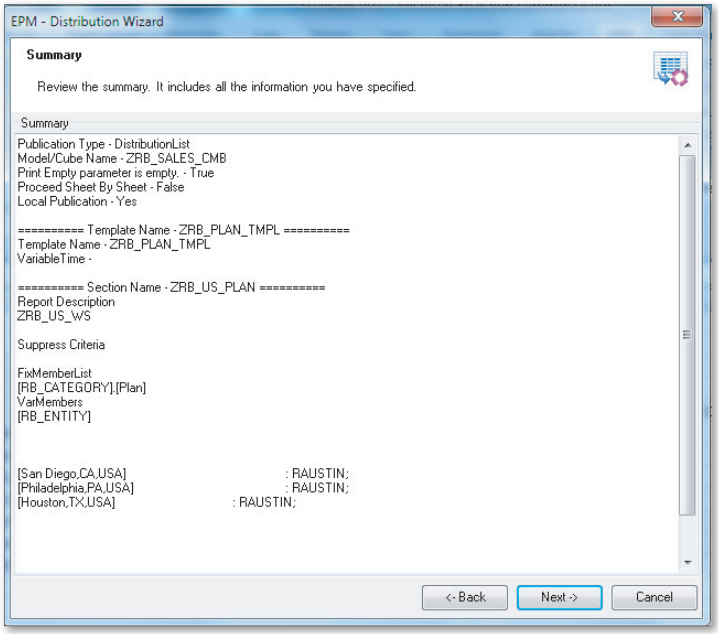


Figure 7.25 Offline Distribution Wizard—Part D

6. The distribution template is now executed on the client machine. In this case, the client is the user who is scheduling this process for execution. The reports and input forms specified in the distribution template are opened, the context view is changed based on the selections defined there, and the data is refreshed and saved in the folder specified. A message confirms that the reports were distributed successfully (Figure 7.26). Click on FINISH to complete the distribution process.

The files are generated in the folder specified in the distribution template (Figure 7.27). The naming convention of the reports or input forms generated by the distributor is as follows:

- The user ID name as specified in the VARIABLE MEMBERS field in the distribution list
- The description of the workbook as specified in the distribution section
- The member selected in the VARIABLE MEMBERS field (which is the name of the .xls file that gets generated)

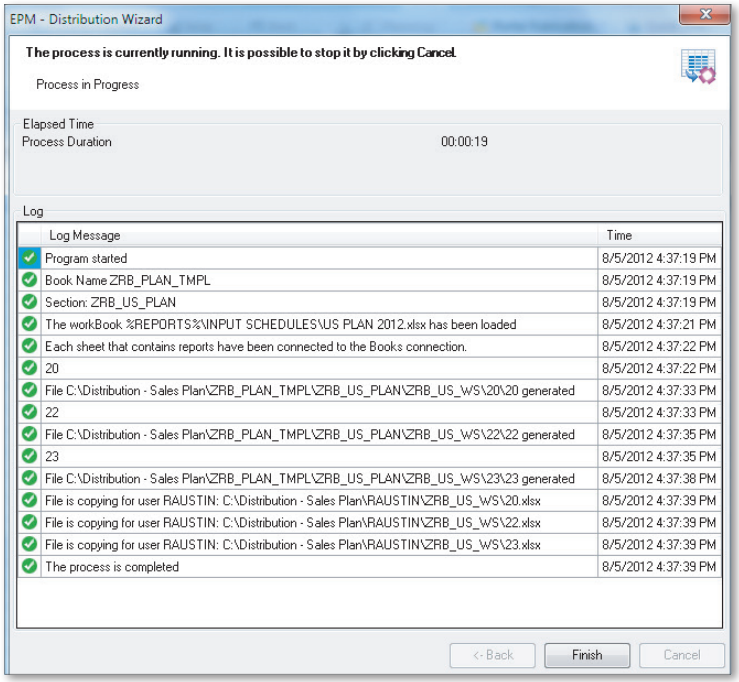


Figure 7.26 Offline Distribution Wizard—Part E

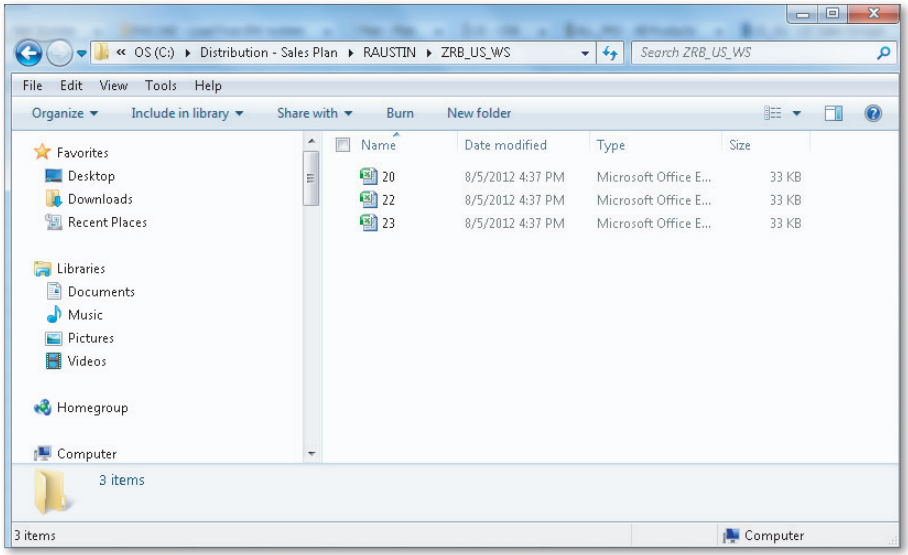


Figure 7.27 Offline Distribution Wizard—Part F

Note

More than one file or email may be generated based on the selections in the VARKEY field of the distribution template.

7.3.3 Collection

Collection applies only to input forms that were distributed using the Offline Distribution Wizard. Input forms that are distributed as offline reports can be updated offline, and then the updates can be pushed back to SAP BPC.

You need sufficient authorization to collect data; specifically, you need access to the USE OFFLINE COLLECTION and USE INPUT FORMS AND SAVE DATA tasks. The distribution list we ran included an input form; we'll now change one of the records in the input form and push the data back to SAP BPC using the COLLECT interface.

1. In the EPM tab of the Excel add-in interface of SAP BPC, select the COLLECT menu option under the DISTRIBUTION command.
2. The next dialog box displays two options for collecting data (Figure 7.28, ❶):
   - OUTLOOK MAILBOX
     When this option is selected, data is collected from an email message attachment.
   - LOCAL FOLDER
     When this option is selected, data is collected from a file located on your local computer. When you select the folder option, you can specify the folder from which you want to collect the data (Figure 7.28, ❷). Click on the NEXT button.
3. Select the files to collect, and click on the NEXT button. You can select more than one file in this step (Figure 7.28, ❸ and ❹). Click on the NEXT button.
4. After the files are selected, a dialog box indicates that the data in the files will be processed. Click on the NEXT button.

   During the update process, all of the validations defined for the model are checked before the data is processed and updated in SAP BPC.

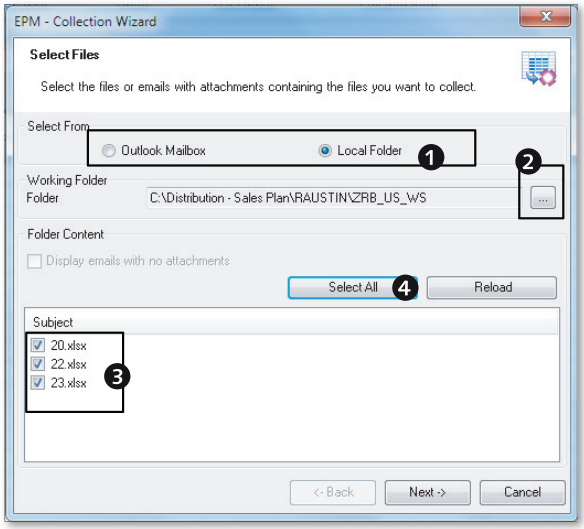


Figure 7.28 Collecting Data

You've now seen how the distribution and collection functionality works in SAP BPC. It is a powerful feature for generating and distributing reports and can be tailored to run multiple reports for different selections from a single distribution template. The ability to distribute input forms in offline mode, update those forms offline, and push them back into SAP BPC is a key feature of the interface. Although data is entered offline, all of the validations for the model are enforced when the data is pushed back to SAP BPC.

In the next section, we'll explain how to use the new feature introduced in SAP BPC 10.0 for NetWeaver to create a template for book publishing and use the features available for publishing reports.

7.4 Book Publishing

Book publishing is a new feature introduced in the SAP BPC 10.0 release for NetWeaver. The ability to publish books had been previously available in the Microsoft release in prior versions but is new to the SAP NetWeaver release. The new feature supports an intuitive graphical interface to support the creation of a book template for publishing reports.

There is often a common need for organizations to report data in SAP BPC for different member selections or to view these reports as one unit. This is especially true after a periodic close when users want to generate income statements and balance sheets for various entities within an organization and publish a book of reports. The book publishing functionality makes all of this possible. Using the book publishing feature, you can create a template and configure multiple reports for different variations of data to support this requirement. Developing a template for this requirement is similar to the steps to create a template for distributing data that we saw in the previous section. You access the interface to create templates for creating and publishing books from the EPM add-in tool for Excel via the Book Publication menu in the EPM tab. You must have the necessary authorizations in your task profile to use this functionality.

The process of defining a book for publishing is managed using a book template. A template consists of one or more sections. Each section can include one or more reports. Every section has distinct dimension member selections for which the reports included in that section are executed. You can specify more than one report when defining a book template.

After a template for book publishing is created, it can be published immediately or scheduled in the background. When a template is scheduled for publishing, this job is executed on the machine of the user who schedules the job.

When data is published using the book publishing interface, the data access available for the user who schedules the book publishing governs the content of the report. Ensure that the user executing the publishing template has sufficient member access to the data requested in the distribution list. If the user does not have access to the data, the book publishing fails to execute and returns a message saying the user does not have sufficient authorization to publish the template.

There are multiple options when publishing the template. In addition to the options available to publish a book immediately or scheduling it as a job, there are options to deliver the reports. The reports can be published in the user's desktop or in the SAP BPC server folder, and the data can be printed or published in PDF format. When you opt to publish the data in PDF format, you can store all of the data in a single PDF or store it as multiple PDFs. The new book viewer is available to view the data when the book is published as a multiple PDF file. The book will always be stored in the public library if you publish the book in multiple PDF format.

Two primary interfaces are used to distribute and collect data:

- Template: This interface is used to create and maintain book publishing templates. The list identifies the reports and selections criteria for publishing the template.
- Publishing: This interface is used for publishing a book template. The job can be scheduled to run immediately, or it can be scheduled as a future job to be run later.

7.4.1 Creating a Book Publishing Template

We'll now explain how to create a publishing template and the steps in configuring the template. We will use a report for creating the template as displayed in Figure 7.29. Recall that we created this report in Chapter 5.

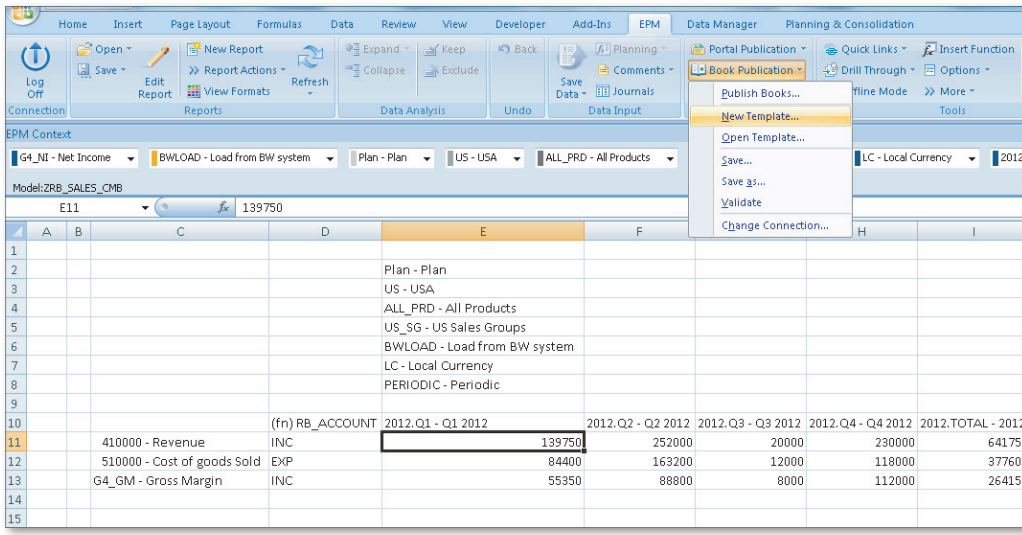


Figure 7.29 Creating a Book Publishing Template—Part A

To create a book publishing template from the EPM add-in for Excel, connect to the ZRB_SALES_CMB model in the ZRB_PR_PLAN environment. You need access to the Publish Books and Delete Published Books task to maintain and publish books.

1. In the EPM tab, select the New Template menu option under the Book Publication menu. If you have already created a template, select the Open Template menu option to edit the template.

2. This opens up the book publishing template (Figure 7.30), which contains a placeholder to enter the name of a template. This applies to the entire publishing template. A publishing template can include multiple sections, and each section in turn can include multiple reports. Enter the following parameters for the book template:

   - Template Name
     Every book publishing template has a name assigned to it. Specify the name of the template.
   - Variable Time
     The time member selected here overrides the value in the Context pane and applies the time to all of the sections in the template.
   - Add/Replace
     The Add/Replace field is set at template level to determine if content gets added or replaced when a book is published multiple times. The possible values are Add, Replace, or Specify by Section.

   The following parameters are set for each section in the template.

   - Section Name
     A book publishing template can contain one or more sections, and a section in turn can include one or more reports. Enter the name of the section here. You can think of a section as a chapter of a book.
   - Add/Replace
     This field is available only when the Specify by Section is selected in the Add/Replace book field. When the option to add or replace is set at the section level in the template, the section gets added or replaced when a book is published multiple times.
   - Report Workbook
     In this parameter, specify the report. You can select a report from the SAP BPC server folder or the local desktop. You can add more than one report for a section.
   - Workbook Description
     Enter a description for the name of the workbook selected. For every report included in the section, it is mandatory to specify a description. Click on the Select Dimensions link to specify the fixed members and variable members for the reports included in the section.

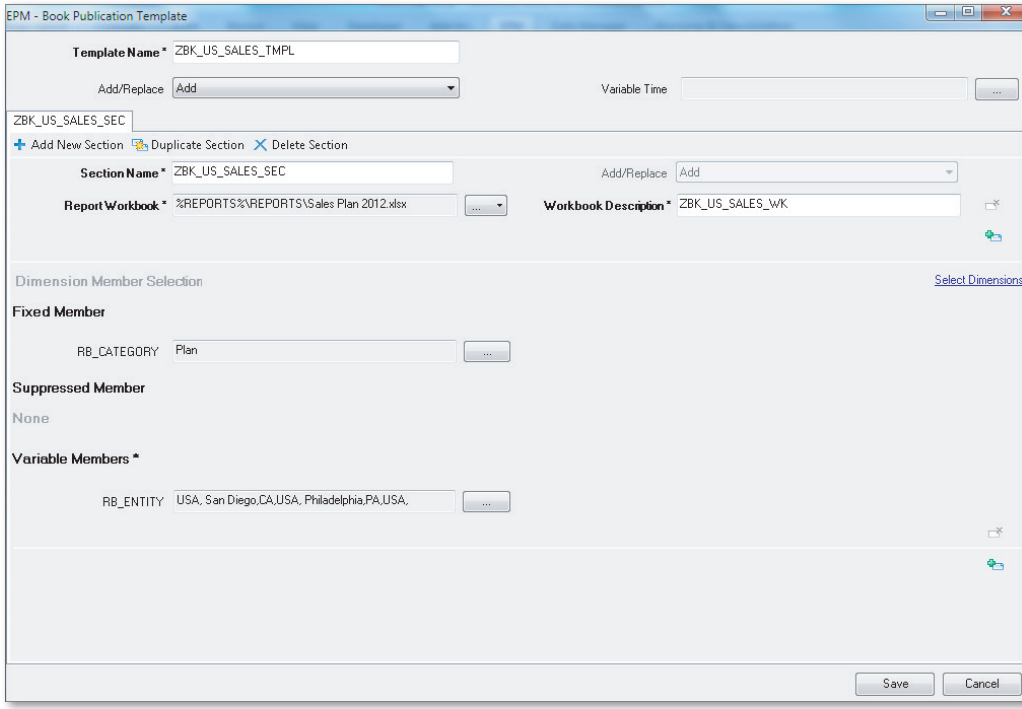


Figure 7.30 Creating a Book Publishing Template—Part B

- Multiple Reports
  Multiple reports can be added to a section by clicking on this icon. Clicking on this icon creates a new line to specify a new report and workbook description.

- Fixed Member
  This field specifies fixed filters for a template section; specify the static selections. You can include Fixed Member values for the different dimensions in your report; for example, you may need to always run a report for the PLAN category. However, you cannot specify more than one member value as a Fixed Member for a single dimension. The field is mapped to the Fixkey field in the template.

- Suppressed Member
  This is used when you want to suppress generating a report for a dimension member when the value is zero. The field is mapped to the Suppresskey field in the template.

- Variable Members

  Specify the dimension values to use to execute the report. This is used to get data from multiple members of one or more dimensions. You can specify more than one dimension in this field. If multiple values are selected for a dimension, multiple reports are created when executing the report. You can specify a relationship of the dimension members to include in the report. In the example shown in Figure 7.31, a report will be generated for each entity that is under the USA hierarchy because Member and Children are selected under Relationship. The field is mapped to the Varkeys field in the template.

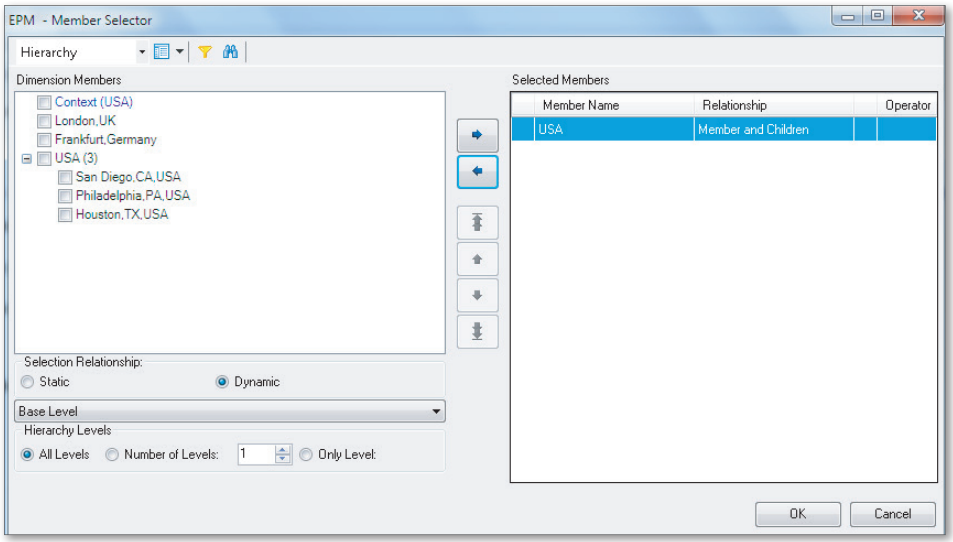


Figure 7.31 Creating a Book Publishing Template—Part C

Select the Save menu option under the Book Publication command to save the template. You can edit the template for further changes by selecting the Open Template menu option and selecting the template under the Book Publication menu. A user who wants to create a distribution template and save it to the company folder needs access to the Edit Content of Public Folder task that falls under Folder Access.

Note

If you do not specify a dimension member in the Fixkey or Varkey fields, the values for the dimension are taken from the context view of the user publishing the report.

3. When a book template needs to be modified, select the Open Template menu option under the Book Publication menu. You can select the desired template and make the necessary changes.
4. You can now validate the book publishing template you just created. Open the template using the Open Server Root Folder menu option under the Open menu. Select the Books folder, and open the template you created. To validate the template, click on the Validate menu option under the Book Publication menu. The system checks to see whether all mandatory fields are entered.

Now that we've created a publishing template, we can execute and publish the data as a single PDF or as multiple PDFs.

7.4.2 Book Publication

Before using the Book Publication Wizard, ensure that you have the necessary access. You'll need access to the Use Offline Distribution task that falls under Collaboration. The process of publication includes the following tasks:

- Selecting the publication type (publish now or schedule)
- Selecting the output format (printer or PDF)
- Selecting the templates and sections to be included in the book

Let's walk through the book publication steps now:

1. In the EPM tab of the Excel add-in interface of SAP BPC, select the Publish Books menu option under the Book Publication menu. You're prompted to select when you want to publication to be performed. Under Publication Type, select when you want to publish the template (Figure 7.32).
2. There are two options to publish the template: Publish Now and Schedule the Publication. When the Publish Now option is selected, the publication is scheduled immediately. When the Schedule the Publication option is selected, you are prompted to enter additional parameters to schedule the publication. Scheduling a publication results in a Windows task being scheduled on the user desktop. The Windows password is required for scheduling the publication. The desktop from where the job is scheduled should be on and be connected to the network when the scheduled job is executed. For our example, we have selected the Publish Now option. Click on Next to proceed with the distribution.

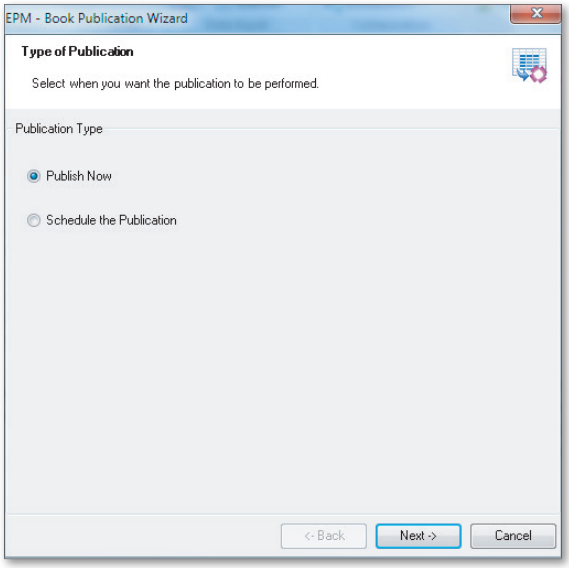


Figure 7.32 Publishing a Book Template—Part A

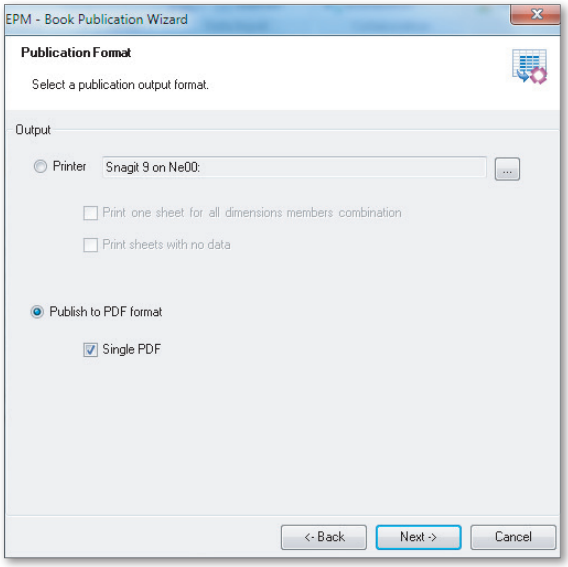


Figure 7.33 Publishing a Book Template—Part B

3. Choose an output mode for publishing when the PUBLISH NOW option is selected (Figure 7.33):

- Printer
  When this option is selected, you specify the printer to print the published reports.
- Publish to PDF format
  When this option is selected, you can specify whether all of the reports in the template should be published in a single PDF. For our example, we have selected the option to publish the data into a single PDF.

Click on Next to proceed with the publication.

4. When the option to publish the reports as a single PDF is selected, you specify the name of the file under which the book is saved (Figure 7.34). You have the option to select whether the report should be published in the local folder in the user desktop or in the public or private folder in the SAP BPC server folder. If the option to publish the reports is not specified as a single PDF, the system publishes in multiple PDF format in the public folder under the SAP BPC server folder. The new book viewer that is available in SAP BPC 10.0 can be used only if you publish to multiple PDFs. Publishing to multiple PDFs is secured if at least one of the variable key members is a secured dimension. There will be one item in the public folder for each section defined in the structure of a book.

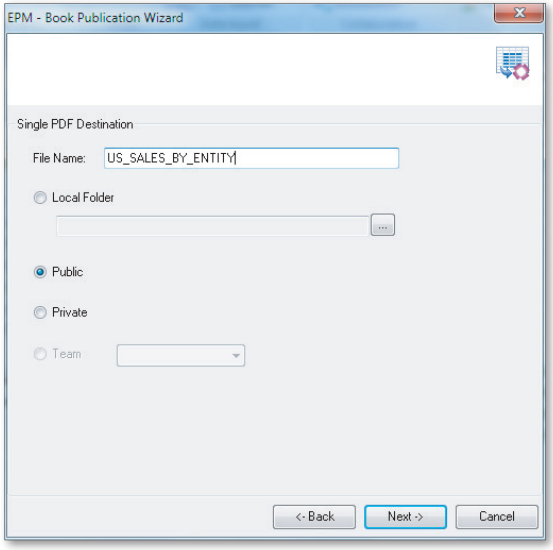


Figure 7.34 Publishing a Book Template—Part C

5. Click on Next to proceed with the publication.

6. In the new dialog box, select the templates and sections to publish (Figure 7.35). After making the selections, click on NEXT.

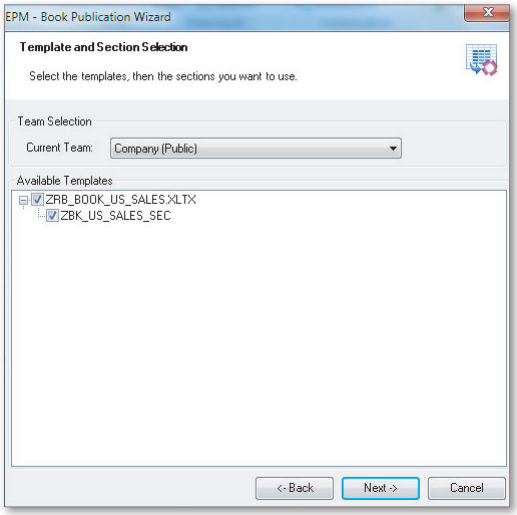


Figure 7.35 Publishing a Book Template—Part D

7. The new dialog box confirms the reports that will be created and the selections that will be used for publishing the reports (Figure 7.36). Click on NEXT to publish the selections.

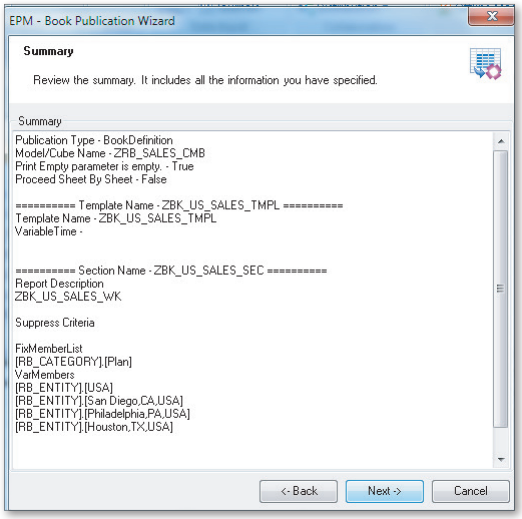


Figure 7.36 Publishing a Book Template—Part E

8. The publication template is executed immediately on the client machine. In this case, the client is the user who is executing the publication template. A message provides details of the publication job executed. Click on FINISH to complete the publication process (Figure 7.37).

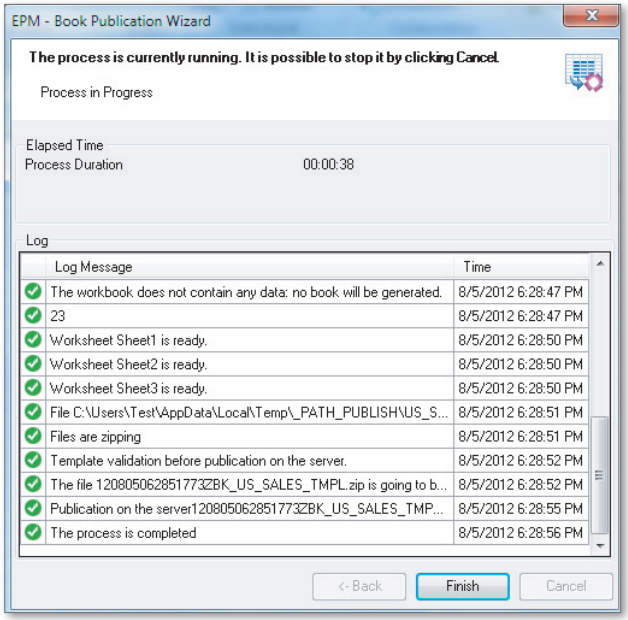


Figure 7.37 Publishing a Book Template—Part F

So the files are generated in the location specified in the publication wizard (Figure 7.38). If the file generated was a multiple PDF format file, it is saved in the public folder in the SAP BPC server folder. Click on the file to view the document.

9. Figure 7.39 displays the single PDF file published in our example.

In the next section, we'll explain how the Word and PowerPoint interfaces are used to integrate SAP BPC data into those types of documents.

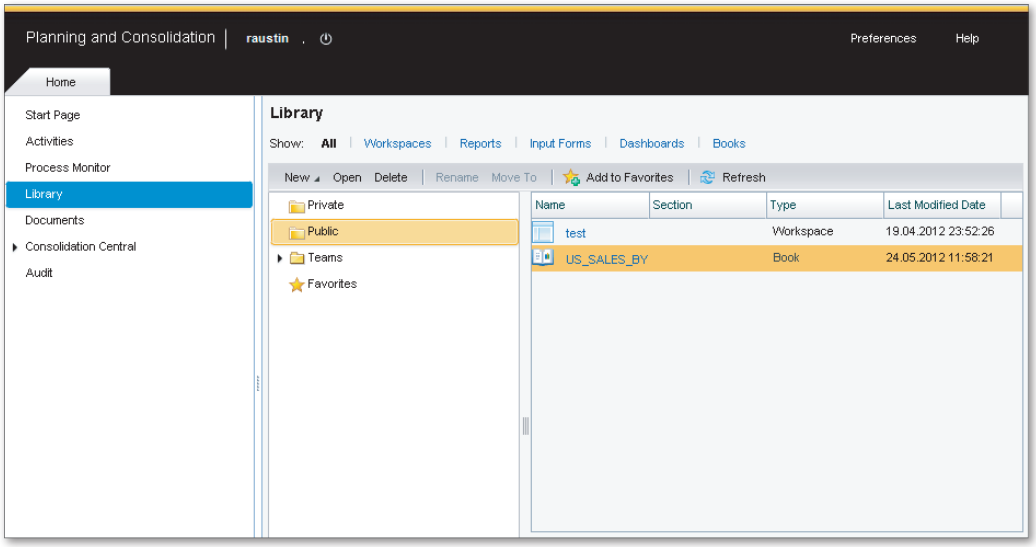


Figure 7.38 Publishing a Book Template—Part G

Figure 7.39 Publishing a Book Template—Part H

7.5 Integrating SAP BPC Data into Word and PowerPoint

The SAP BPC EPM add-in allows you to not only integrate data into Microsoft Word and PowerPoint but also to embed objects such as input schedules into these

documents. You can also add and view comments from this interface. The following features are available when you use the Word and PowerPoint interfaces:

- Log On
  When this command is selected, you can connect to an SAP BPC system from Word or PowerPoint. Additional choices are available to manage connections using the Manage Connections menu option from the Report Actions command.
- Insert Data
  You use the Insert Data command in the EPM tab for inserting SAP BPC data into a Word or PowerPoint document. After you add the SAP BPC data object into a document, you are prompted with a dialog box to select the member value for the dimensions. The values selected are used to retrieve the data. You can select only one value for a dimension. If a dimension member is not selected here, the member value in the EPM Context pane is used for retrieving the data. You can view the dimension member selections for an object you inserted by placing the cursor on the object. The selections are displayed in the EPM pane only for those dimensions whose member selections do not correspond to the value in the EPM Context pane.
- New Report
  When this command is selected, you can create a new report similar to the interface available in the EPM Excel interface.
- Report Action
  When this command is selected, there are additional menu options to delete, copy, or paste a report. A report copied from the EPM Excel interface can be pasted into Word and PowerPoint using the Paste option.
- Open
  When this command is selected, additional options are available to select a report from the local desktop or SAP BPC server folder. The menu options are available when the Open command is selected in the EPM tab.
- Comments
  The Add Comment and Find Comment menu options under the Comments menu are available to add and search comments.

- Refresh
  When you have inserted a link for the data in the document using the Insert Data button, you can refresh the value for this object using the Refresh command.

When an object is inserted into a document, you can only retrieve the data from SAP BPC; you cannot use it to update the data in the database. In addition, you will not be able to use Excel formulas or local member formulas. SAP BPC formatting features that are available when using the Excel interface are also not available here.

We'll now use a simple example to show you how you can use the EPM add-in for Word to display SAP BPC data in a document. You can access the SAP BPC for Word and SAP BPC for PowerPoint interfaces from the EPM tab.

1. Access the EPM add-in tab from the Word interface. Connect to an SAP BPC model using the same steps followed to connect to SAP BPC from the Excel interface. After you are connected, the commands to interface with the data in SAP BPC are available to use (Figure 7.40).

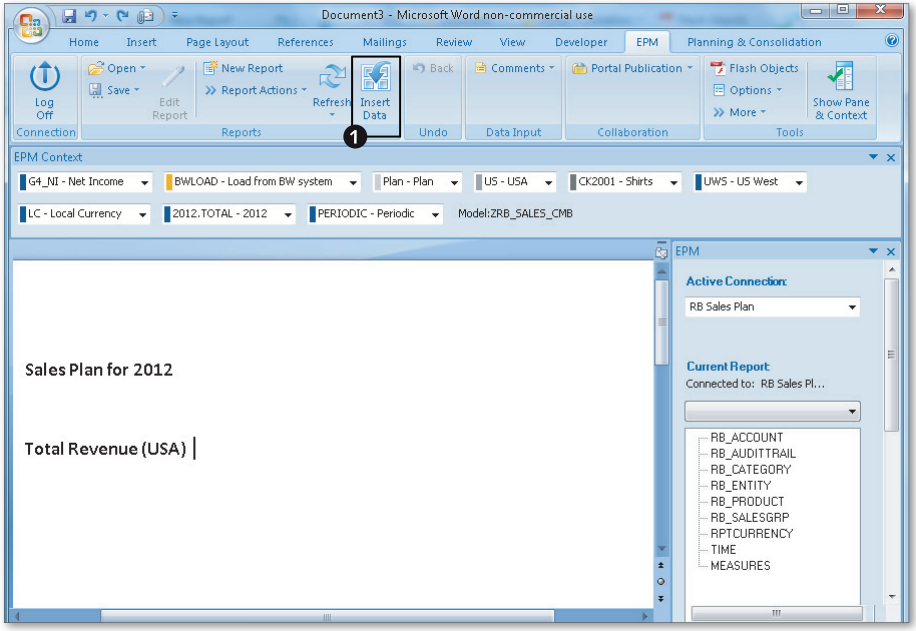


Figure 7.40 Integrating SAP BPC Data into Word—Part A

2. Open a new Word document. Highlight the cursor anywhere in the document, and then click INSERT DATA (Figure 7.40, ❶). A dialog box is presented in which you can specify the member selections for each dimension (Figure 7.41). When a value is not selected for a dimension, the member value from the EPM CONTEXT pane is used. The value is retrieved for the selections and displayed in the document. Place the cursor in the place where you added the object to view the selections. The selections are displayed in the EPM pane (Figure 7.42, ❷). The custom selections are displayed for any selection that does not match the context member value for the dimension.

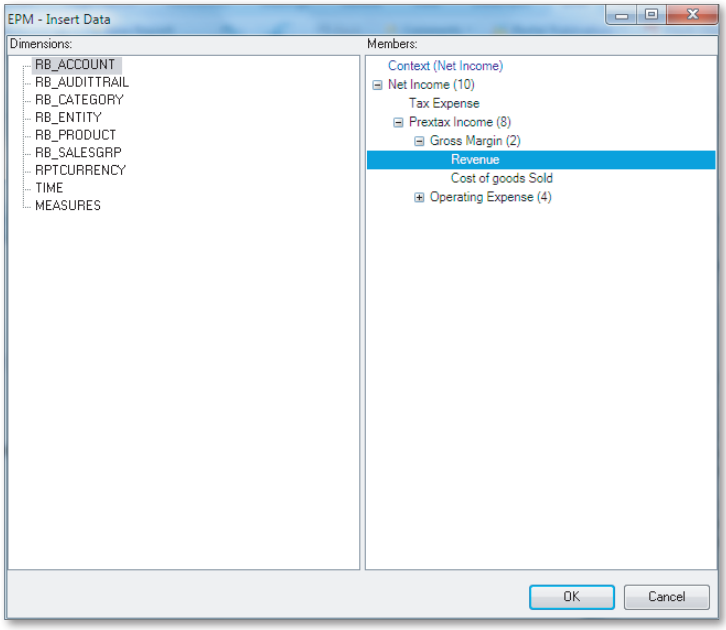


Figure 7.41 Integrating SAP BPC Data into Word—Part B

When you click on the REFRESH command in the EPM tab, the data from SAP BPC is refreshed for all of the objects inserted in the Word document.

In this section, you saw how to interface SAP BPC data into Word and PowerPoint documents using the EPM add-in.

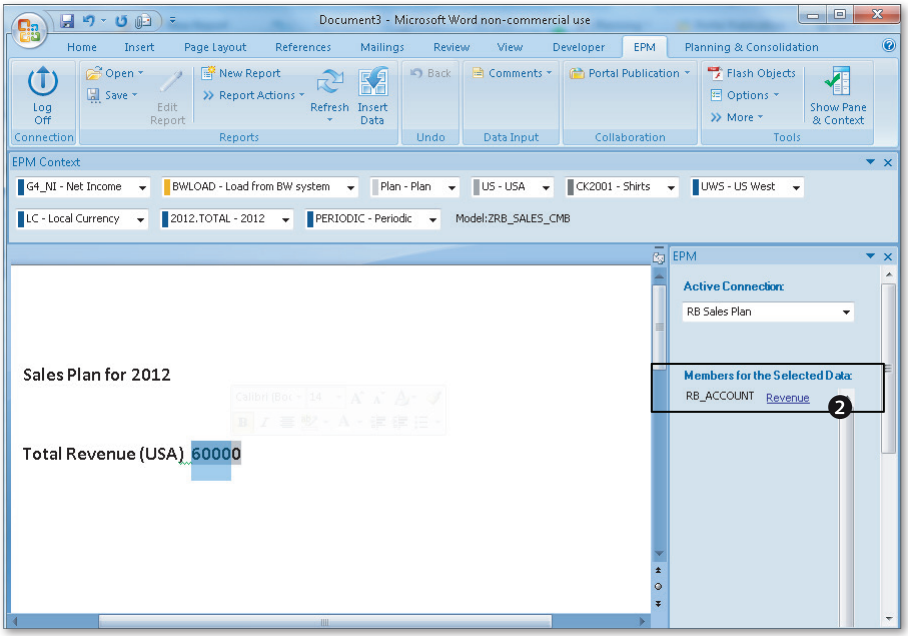


Figure 7.42 Integrating SAP BPC Data into Word—Part C

7.6 Summary

In this chapter, you learned about key collaborative features available to monitor, lock, share, distribute, and collect data in SAP BPC. You learned how to use work status to lock data from changes and to enable workflow in applications. Then you learned how to enter comments and view them in SAP BPC. We discussed the distribution and collection features for distributing data, which allow users to update data and send it back into the planning model using the collection interface. Finally, we explained the new feature introduced in SAP BPC 10.0 for NetWeaver for creating a template for publishing books and the SAP BPC interfaces for Word and PowerPoint.

In the next chapter, we'll discuss the essential tools to help with the smooth functioning of your SAP BPC system. We'll explain the process of transporting objects from the development system to the quality assurance and production systems and discuss key topics such as locking, environment parameters, statistics, audits, and document management.

SAP BPC offers a robust tool to manage changes to objects made in the development environment and provides an easy-to-use interface for moving them across the system landscape. The system also provides a set of utilities to manage locks, maintain environment and model parameters, record statistics of usage, audit changes to objects and data, and configure and manage security.

8 Essential Tools for Building Models

This chapter discusses the essential tools for maintaining SAP Business Planning and Consolidation (SAP BPC) models; in particular, this chapter discusses the SAP BPC system as a set of utilities to manage the transport of objects across the system landscape, manage locks, maintain environment and model parameters, record statistics of usage, audit changes to objects and data, and configure and manage security.

In Section 8.1, we'll explain how to transport SAP BPC objects from the development system to the quality assurance and production systems. We'll explain the changes in the process of how objects are transported in the SAP BPC 10.0 version. This will help you gain a sound understanding of how to manage transports in your organization. We'll also discuss the backup and restore tools in SAP BPC that are available to address disaster recovery scenarios.

In Section 8.2, we'll explain the locking concept used in SAP BPC for NetWeaver and cover the technical details concerning how the system handles locks and maintains the integrity of data. We'll also look into the configuration tables that are available to customize the locking parameters for a model.

In Section 8.3, we'll discuss the parameters you can set for an environment and the models that belong to an environment. You'll see how these parameters influence the functionality of a model.

In Section 8.4, we'll introduce how to enable statistics for a model and the benefits it provides in managing the performance of your model.

In Section 8.5, we'll cover how to enable auditing for a model and the benefits it provides in monitoring the changes made in your model.

In Section 8.6, we'll discuss how to use the content management tool in SAP BPC to store, manage, and enforce the security of unstructured data.

In Section 8.7, we'll discuss security, which is an important component in any model. SAP BPC offers two types of security: one to manage the tasks a user can perform, and another to determine the data-level access a user has in a secured model. In this section, you'll learn how to set up security for your model in SAP BPC.

We'll start by looking at the interface used for transporting objects in SAP BPC and how it aligns with the standard process of transporting objects in the SAP NetWeaver Business Warehouse (BW) system.

8.1 Transporting SAP BPC Objects

SAP supports a landscape that includes development, quality assurance, and production systems to develop, test, and implement a model in SAP BPC. The process of development begins, appropriately, in the development system. When development is complete, unit testing is performed, and the development objects are transported to the quality assurance system, where they are tested extensively to confirm that the functionality works as expected. This is normally the system where integration/user acceptance tests are performed. When the testing is completed successfully in the quality assurance system, the changes are moved to the production system. You should follow this process assiduously to avoid any issues of unintended changes going into production, leading to serious consequences. SAP BPC for NetWeaver supports the process of moving objects from one system to another via the transport mechanism.

In SAP BPC version 5.1, the transport mechanism was not available because the product was not integrated with the SAP NetWeaver system. The process used in this version moved or did a backup of objects in the development system and then restored them in the target system. With the integration of SAP BPC with NetWeaver, in the SAP BPC 7.0 for NetWeaver version, transport functionality was extended to the entire application set (now called environment). This feature was an improvement to the prior version because it provided a process to transport objects from a source system to a target system. However, it still did not provide the ability for the user to select the individual objects inside an application set to transport. The whole application set (environment) had to be selected for the transport. Subsequently, a few enhancements in the SAP BPC 7.5 for NetWeaver version provided a process

to individually select input schedules or reports and send them to a target system via executing a program in the SAP NetWeaver BW system.

With the introduction of SAP BPC 10.0, the standard transport process available in the SAP NetWeaver BW system is now available to transport SAP BPC objects. You select the TRANSPORT menu option in the administrator's workbench and select the options to transport the objects. The transport process in SAP BPC for NetWeaver now truly leverages the existing framework in SAP systems to transport objects. Using this tool, you can identify objects that need to be moved across the landscape and send them in a streamlined and controlled fashion, reducing any problems in this process.

SAP Best Practices does not recommend doing any development type activity in the production system because objects should be fully tested before they are moved, but there are cases where the customer needs to modify objects directly in the production system. This is especially true in reporting, where you may need to create ad hoc reports directly in the production system. In these scenarios, you can set the access level for users to make the changes directly in production.

The transport mechanism involves the following steps:

1. Create or modify objects in the development system.
2. Perform unit testing in the development system.
3. Create a transport request in the development system for the objects created or modified.
4. Transport the request to the quality assurance system.
5. Perform integration/user acceptance testing in the quality assurance system.
6. Move the objects to the production system.

In SAP BPC 10.0, the same framework used in SAP NetWeaver BW is available for transports, and the mechanism now supports the ability to manage objects at a granular level.

Follow these steps to transport objects:

1. Log in to the SAP NetWeaver BW system, and execute Transaction RSA1. Click on the TRANSPORT CONNECTION option (Figure 8.1, ❶). Under the SAP TRANSPORT folder, click on OBJECT TYPES (Figure 8.1, ❷ and ❸).

2. The option to select objects and create a transport for SAP BPC is available under the More Types folder in the center pane. The following features apply when transporting SAP BPC objects (Figure 8.1, ❹):
   - The objects that are collected to be displayed in the transport interface are based on the setting in Grouping. The grouping is based on the flow of information. You can choose from Only Necessary Objects, In Data Flow Before, In Data Flow Afterwards, In Data Flow Before and Aftrwds, and Save from System Copy. We recommend that you select the Only Necessary Objects option before starting the process to transport SAP BPC objects (Figure 8.1, ❺).
   - Under Collection Mode, set the option to Collect Automatically (Figure 8.1, ❻). When using this option, the system will automatically identify and check the objects that are to be transported. The objects displayed can then be overridden manually by selecting or deselecting objects as necessary.
   - To create a transport, click on Select Objects under Environment in the center pane. This will prompt a dialog box to select the environment. Select the environment that contains the objects that you want to transport, and click on Transfer Selections. The environment is listed in the right pane. When you expand the environment, you will see the underlying objects (Models, Business Process Flows, Business Rules, Dimension, etc.) for the selected environment.

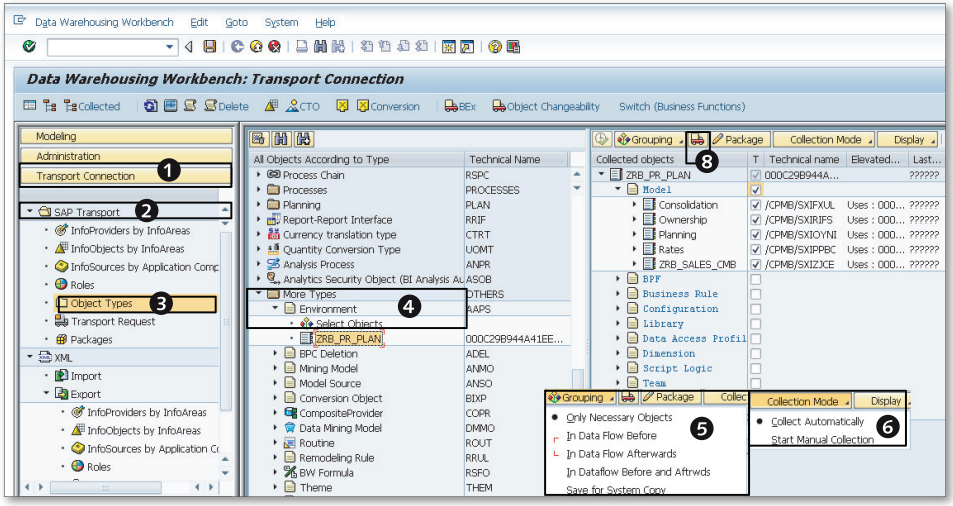


Figure 8.1 Transporting Objects in SAP BPC for NetWeaver—Part A

You can expand on an object category and select individual objects under it. Using this process, you adopt the granular concept and limit the objects that are transported.

If you select the object category, all of the objects below it are selected. If you unselect it, all of the objects below it are unselected. You can also select and right-click on a node and select the menu option TRANSPORT ALL BELOW (Figure 8.2, ❼).

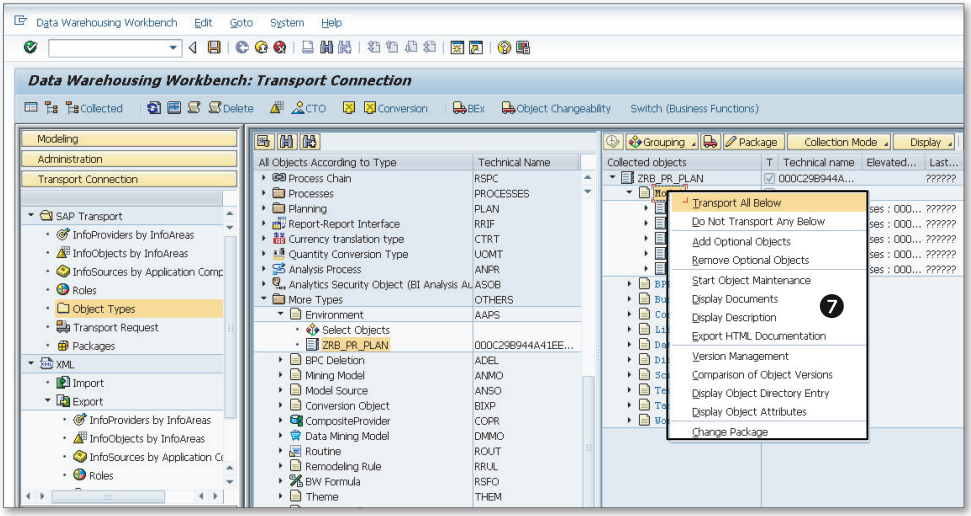


Figure 8.2 Transporting Objects in SAP BPC for NetWeaver—Part B

3. When you are ready to create a transport, click on the transport icon (refer to Figure 8.1, ❽). All of the objects with a checkmark in the TRANSPORT column are added to the transport. You're prompted to create the transport request in the following screen.

4. You may sometimes want to delete SAP BPC objects in target systems. Certain types of objects—MODEL, DIMENSION, DATA MANAGER PACKAGE, DATA MANAGER PACKAGE GROUP, and DATA MANAGER PACKAGE LINK—support deletion mode. To create a transport for deletion, click on MORE TYPES in the center pane, and double-click on SELECT OBJECTS under BPC DELETION (Figure 8.3, ❾). Then select the objects you want to delete in the right pane.

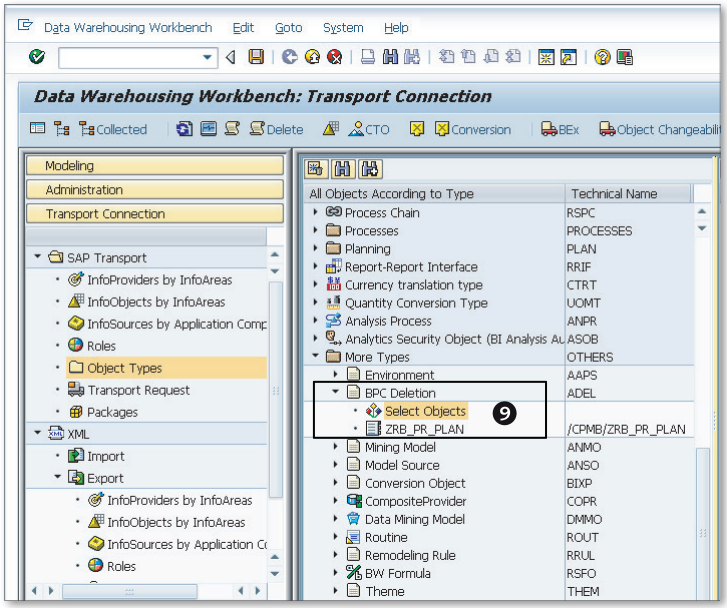


Figure 8.3 Transporting Objects in SAP BPC for NetWeaver—Part C

5. To ensure a transport is successful, confirm that the dependent objects are included in the transport. For example, if you are transporting a model, make sure all of the dimensions required for the transport are in the current transport or are already available in the target system prior to sending this transport. When a dependent object is not included in the transport and is not available in the target system, the transport fails.
6. After a transport is created, it has to be released before it can be transported to the target system. Use Transactions SE09 and SE10 to display, maintain, and release transports (Figure 8.4). Specify the user or the request number to display transports.
7. Set the environment to offline in the source and target system before a transport request is applied.
8. The Change and Transport System (CTS) has to be set up before the transport process can be initiated in the target system. The Basis team in your organization is responsible for this task. Essentially, this requires configuration of SAP NetWeaver Transport Framework (by creating transport domains and routes)

and configuration of the system to import client dependent objects (creating the BW background user ALEREMOTE and creating an RFC destination to allow the import method to be executed in the correct client).

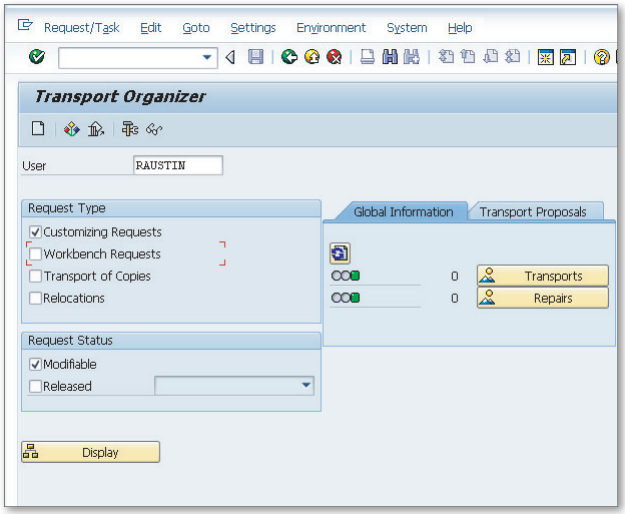


Figure 8.4 Transporting Objects in SAP BPC for NetWeaver—Part D

9. When an object assigned to a transport is transported and it already exists in the target system, the changes to the object are transported, and the configuration of the object is sychronized with the one in the source system. If an object transported does not exist in the target system, it is created as a new object in the target system.
10. When a transport fails, you can get additional information about the details of the transport. To display the logs, you can view the log associated to the transport either by using either Transaction SE09 or Transaction SLG1. When using Transaction SLG1, enter "UJ" in the OBJECT field, enter "UJT" in the SUBOBJECT field, and click on EXECUTE (Figure 8.5, ❿, ⓫, and ⓬).

The transport administrators in the SAP Basis team perform the job of importing the request into the target system. After the transport is imported, you can view and display the logs, which provide detailed information on how the transport was processed in the target system, as well as any errors that occurred.

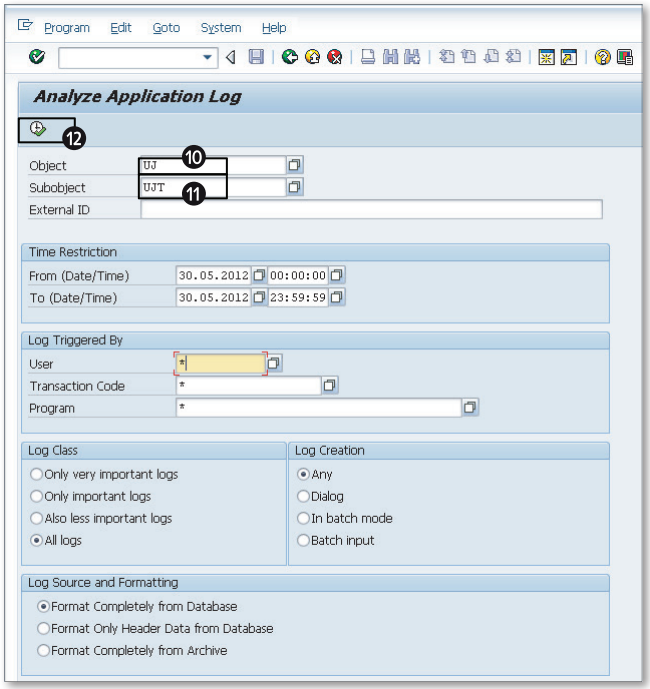


Figure 8.5 Transporting Objects in SAP BPC for NetWeaver—Part E

You should now understand how to use the transport process to move objects from the development system to the quality assurance and production systems.

Transaction UJBR is available to back up and restore an environment. This feature is especially useful for addressing disaster recovery scenarios. It also provides an option to copy an environment from one system to another system. This transaction allows users to back up and restore objects, metadata tables, master data, and transaction data in an environment. You should use the transport mechanism outlined previously to transport objects from the development system to quality assurance and production systems. You can use the backup and restore tool in disaster recovery scenarios and when you want to create a copy of an environment in another system.

Next, we'll review the locking process used in SAP BPC when data is updated for a model.

8.2 Concurrency Locking

Locking is an important concept to understand and consider during model development. The purpose of locking is to prevent two users from updating the same data at the same time. The model should prevent this from happening to avoid data inconsistency.

We'll first discuss different approaches to locking. The approach to locking impacts the internal process used inside SAP BPC before data is updated.

8.2.1 Approaches to Locking

Locking can be considered from two viewpoints. From the granular view, locking can be set at the record level on the data that is updated. The granular approach has the advantage of locking only the records that are updated, but this is likely to take more time because every record that needs to be updated is locked during the process. This option also consumes more memory because you'll be acquiring locks for every record that is updated.

The second approach is to lock based on a range of data. With this approach, locking is based on the range of data the user is updating. This makes it faster for the system to acquire locks but has the disadvantage of making it easy for you to lock more records than necessary. Although the end objective is to prevent two users from updating the same region of data, it is necessary to achieve a balance when using these options.

We'll now discuss the locking features in SAP BPC for NetWeaver, such as *concurrency locking,* which is based on a mixture of these options.

8.2.2 Locking Features in SAP BPC for NetWeaver

In SAP BPC, data is uploaded to the database when data is loaded using the Data Manager package and when data is updated via manual update using the interfaces for Excel, Word, and PowerPoint. (Data is updated even when journals are posted, which we'll discuss in more detail in Chapter 9.) During the update process, data integrity must be maintained so that two users do not update the same region of data at the same time. It is also important to ensure that one user does not overwrite data entered by another user.

SAP BPC for NetWeaver uses the concept of *concurrency locking* to lock data during updates. Concurrency locking comes into play only when a user sends data to update the database; no locks are obtained in SAP BPC prior to the update when the user enters data.

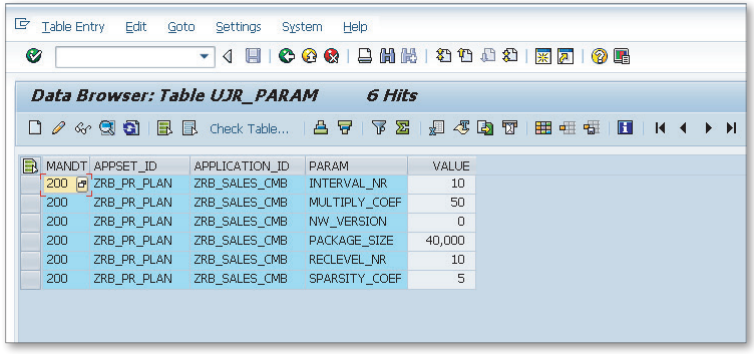
Note

Locking in SAP BPC is mandatory. There are no settings to bypass locking.

The concepts behind locking differ between SAP BPC and the other tools for planning used in SAP NetWeaver BW—namely SAP NetWeaver Business Planning and Simulation and SAP NetWeaver BW Integrated Planning. In these tools, data is locked when a user opens the planning template to enter plan data for a particular selection. In SAP BPC, data is locked only when the plan/consolidation data is saved. However, the same framework for locking is used by all three tools.

8.2.3 Locking Process in SAP BPC for NetWeaver

SAP BPC for NetWeaver obtains locks only when a user initiates the process to update data. This update is also referred to as the *write back* process. The request for update sent by the user is split into packets, and the default size of a packet is 40,000 records. This setting is governed by the `PACKAGE_SIZE` parameter in Table UJR_PARAM (Figure 8.6) and can be changed by modifying the default settings using Transaction UJR0 for a model in the environment.



MANDT	APPSET_ID	APPLICATION_ID	PARAM	VALUE
200	ZRB_PR_PLAN	ZRB_SALES_CMB	INTERVAL_NR	10
200	ZRB_PR_PLAN	ZRB_SALES_CMB	MULTIPLY_COEF	50
200	ZRB_PR_PLAN	ZRB_SALES_CMB	NW_VERSION	0
200	ZRB_PR_PLAN	ZRB_SALES_CMB	PACKAGE_SIZE	40,000
200	ZRB_PR_PLAN	ZRB_SALES_CMB	RECLEVEL_NR	10
200	ZRB_PR_PLAN	ZRB_SALES_CMB	SPARSITY_COEF	5

Figure 8.6 Parameter Table for Concurrency Locking

The packet of data for the update is analyzed, and the region of data that should be locked is determined and set. The SAP BPC system determines whether it needs to perform record-level locking or range locking of data.

Record-Level Check

When the number of records to be updated is less than 10, record-level locking is used. The value of 10 is set in the parameter `RECLEVEL_NBR` in Table UJR_PARAM and can be configured to a different value for a model in the environment, based on the requirements of that application. To modify this parameter, execute Transaction UJR0, and click on the green checkmark (Figure 8.7). When the `RECLEVEL_NBR` parameter is set to a high number, updates may take longer to complete and result in higher memory usage due to the time it takes to obtain locks.

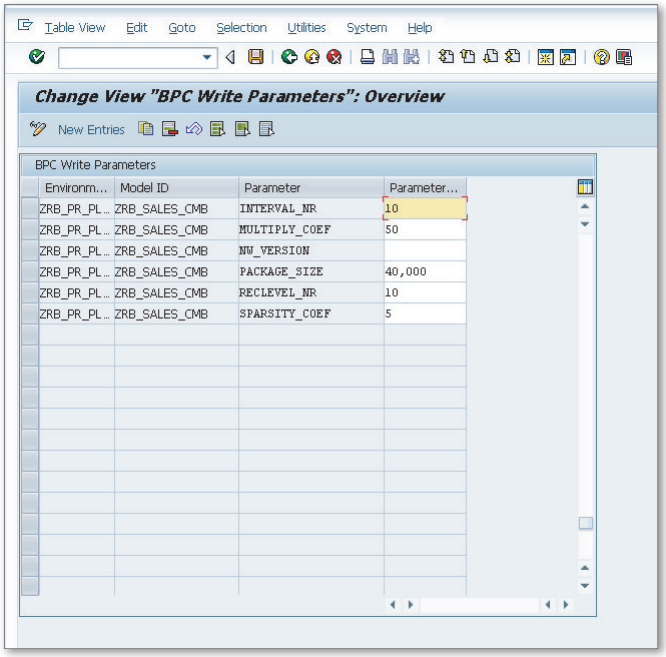


Figure 8.7 Customizing Concurrency Locking Parameters

Sparsity Check

When the number of records is equal to or higher than the value in the parameter `RECLEVEL_NBR`, a sparsity check is performed. The sparsity check is carried out as follows.

First, the system determines the number of unique members in each dimension in the record set and then multiplies the number of unique member counts of each dimension to arrive at a value. For example, assume that there are four dimensions, and the numbers of unique dimension members in the dimensions are as follows:

- Account: 15
- Time: 3
- Category: 2
- Entity: 1

When you multiply these numbers, the value is 90. We'll call this value *X*. Next, the system multiplies the total number of records to be updated by the sparsity coefficient parameter `SPARSITY_COEF` in Table UJR_PARAM. We'll call this value *Y*. The default value for the `SPARSITY_COEF` parameter is 5. You can change the value of this parameter using Transaction UJR0.

If $X > Y$, the record set is considered to be sparse. If the number of records to be updated in this example is 14, the value of *Y* would be $14 \times 5 = 70$. Because $X > Y$, the record set would then be considered to be sparse.

If records are sparse, the number of unique dimension member values for each dimension is determined and sorted in ascending order. The unique member records of each dimension are multiplied until the value of the `MULTIPLY_COEF` parameter in Table UJR_PARAM is reached.

For example, let's say there are four dimensions, and the numbers of unique dimension values in the dimensions sorted in ascending order are as follows:

- Entity: 1
- Category: 2
- Time: 3
- Account: 15

Let's also say that, for this application, the value of `MULTIPLY_COEF` is 50. Now, multiply the unique dimension member counts in the Entity and Category dimensions; the value is $1 \times 2 = 2$, which is less than 50. Next, multiply the value obtained in the previous step by the unique Time dimension's member count, which results in a value of $2 \times 3 = 6$. This value is still less than 50. Continue multiplying the value obtained in the previous step by the unique Account dimension's member

count, and you'll obtain a value of 6 × 15 = 90. At this point, the value exceeds the MULTIPLY_COEF value of 50. In this case, individual values in the Entity, Category, and Time dimensions are locked. The range of values in the Account dimension is also locked.

If records are not sparse, all of the members of dimensions that have 10 or fewer distinct values are locked. The value of 10 is dependent on the INTERVAL_NR parameter in Table UJ_PARAMETER and can be modified using Transaction UJR0. For those that have 10 or more distinct values, the system creates a lock between the low and high values for the member values in that dimension.

If the system is able to obtain locks, the locks are obtained, and the data is updated. After the data is updated, the locks are released. If there is more than one package to update, the system processes the next package to obtain the locks using the same process for that package and updates it.

If the system is not able to obtain locks for a package, the update for that package and the subsequent package fails with an error indicating the system was not able to obtain locks for the update. But the package(s) for which the system was able to obtain locks, and for which data was processed successfully, cannot be rolled back. For example, let's say there were three packages to update for a request sent by a user. Assume that the system was able to successfully acquire locks on the first package and update the data in the model but was not able to acquire locks on the second package. As a result, the second package and the subsequent updates fail, even though the first package was updated in the cube and contains the changes made by the user.

The failed packages can be run again because SAP BPC posts only delta values during this update. Figure 8.8 shows an overview of the concurrency process.

Note

For journals, the delta mechanism is not used when posting values. The package size for journals is set to 99,999,999 to ensure that the concept of packages is not used when updating data using journals.

In the next section, we'll discuss the use of web administration parameters and see how they influence application behavior.

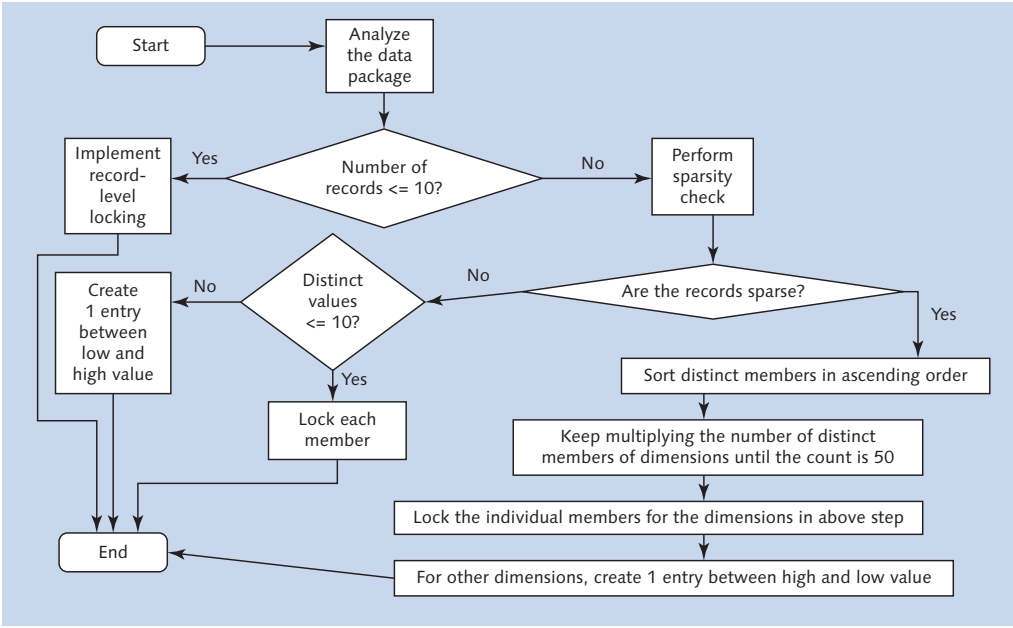


Figure 8.8 Concurrency Locking Process

8.3 Administration Parameters

Web administration parameters influence the behavior of the environment and models inside an environment. In the SAP BPC 7.x versions, the administration parameters were exclusively set using the web interface. In the SAP BPC 10.0 version for NetWeaver, the process of setting the administration parameters has changed. The administration parameters in SAP BPC 10.0 are classified into the following three types:

- Functional parameters: The functional parameters are set in the SAP BPC Web Client. These are parameters defined by the user to influence the behavior of the environment or model. An example of a functional parameter is `APPROVAL-ORG`, which is required for setting and using work status inside a model.
- Configuration parameters: This type of parameter is primarily technical and is set in the SAP NetWeaver BW system. They fall under three levels: global, environment, and model. A parameter defined under the global level applies to all of the environments in the SAP BPC system. The parameters in the environment

apply to all of the models set for a specified environment. The parameters at the model level apply to a specific model.

- System parameters: This type of parameter applies to the environment for example like the availability of the environment to the users and is set in the SAP BPC Web Client.

The functional parameters are applicable to environment and model objects, and they are set in the SAP BPC Web Client. Table 8.1 lists the functional and system parameters you define in the Web Client.

Recall that the configuration parameters are set as an IMG parameter in the SAP NetWeaver BW system, and they are either global, environment, or model. You define the parameters using Transaction SPRO and by selecting the SAP Reference IMG option. In the IMG structure, you can access the three levels for defining parameters under Configuration Parameters in the Planning and Consolidation hierarchy. Table 8.2 lists the parameters you define in the BW system.

Parameters	Description
APPROVALORG	If you want to use the work status feature, you must use this field to identify the hierarchy level (H1, H2, H3, ..., H*n*) in the Owner dimension. This hierarchy is used to identify the implied manager. The owner associated with a node in the hierarchy is the manager of the owners associated to the children. You can define only one hierarchy for each model within an environment.
AVAILABLEFLAG (Required)	Controls whether the system is offline or not. `Yes` means the system is online and available for sending data to the database. You can take the system offline by changing the value to `No`.
AVAILABLEMSG	This message displays to users who try to access an environment that is offline.
Calculation	This parameter is used for the account transformation and is a rule type under business rules. You can change this value for a model in the Business Rules view under the Rules Domain in the Web Client.

Table 8.1 Parameters Defined in the SAP BPC Web Client

Parameters	Description
IntcoBookings	This parameter is used for the intercompany booking and is a rule type under business rules. You can change this value for a model in the Business Rules view under the Rules Domain in the Web Client.
Opening Balance	This parameter is used for opening business rules. You can change this value for a model in the Business Rules view under the Rules Domain in the Web Client.
JRN_REOPEN_PROPERTY	A custom journal module assumes that the property named UB must be present in the Account dimension to further filter the journals to reopen. The default is group. If group is specified, there is no need to modify the Account dimension.
ORG_INTCO (Required)	The value set here should also be a member ID in the INTCO dimension in the ownership model if you're using dynamic hierarchies.
ORG_PARENTPROPERTY (Required)	This parameter is used with dynamic hierarchy statutory models when defining fixed hierarchies. The value must match the value in the ParentProperty property value of entities in the statutory model.

Table 8.1 Parameters Defined in the SAP BPC Web Client (Cont.)

Parameter Level	Parameter Name	Description
Environment (Required)	ALLOW_EXTENSIONS	Defines the file extensions of files the system allows users to upload to the model, including Data Manager files, content library files, web-ready files, and library files. When set to the default ALL, SAP BPC allows all extensions.
Environment (Required)	ALLOW_FILE_SIZE	The maximum file size SAP BPC allows users to upload. The default value is 100 MB.

Table 8.2 Parameters Defined in the IMG in the SAP NetWeaver BW System for SAP BPC

Parameter Level	Parameter Name	Description
Model (Required)	BPC_STATISTICS	When set to ON, various SAP BPC modules write detailed runtime statistics to Tables UJ0_STAT_HDR and UJ0_STAT_DTL. You can use this information to monitor system performance. Valid values are ON and OFF.
Environment	DEFAULT_ EXTENSIONS	The file extensions the system allows users to upload by default: XLS, XLT, DOC, DOT, PPT, POT, XML, MHT, MHTML, HTM, HTML, XLSX, XLSM, XLSB, ZIP, PDF, PPTX, PPTM, POTX, POTM, DOCX, DOCM, DOTX, DOTM, CDM, TDM, PNG, GIF, JPG, CSS, and MRC. Also see ALLOWEXTENSIONS.
Environment	LOGLEVEL	Used by an environment to control the level of the ABAP log, which you view via Transaction SLG1. LOGLEVEL has the following possible values: ▶ 0—None: Log is off. ▶ 1—Error: Log only error, abort, and exit messages. ▶ 2—Warning: Log warning, error, abort, and exit messages. ▶ 3—Info: Log info, status, error, abort, and exit messages. ▶ 4—Trace: Log info, status, error, abort, exit, and trace (highly detailed) messages.
Environment	MAXLRCOLUMNS	The maximum number of columns to display in a live report in the SAP BPC Web Client. The value includes header and data columns.
Environment	MAXLRROWS	The maximum number of rows to display in a live report in the SAP BPC Web Client. The value includes header and data rows. For example, if you specify a value of 5, one heading row and four data rows are displayed.

Table 8.2 Parameters Defined in the IMG in the SAP NetWeaver BW System for SAP BPC (Cont.)

Parameter Level	Parameter Name	Description
Model	ORG_ACCOUNTOWN	The default value is PGROUP.
Global Parameter (Required)	SMTPAUTH	The authentication method of the SMTP server. 0 = Anonymous 1 = Basic 2 = NTLM This setting does not change the method on the SMTP server but must match the type of authentication enabled on it. Failure to set this appropriately can result in errors from the email server.
Global Parameter (Required)	SMTPPORT	Port number for your SMTP email server. The default is port 25, which is the default SMTP server port number.
Global Parameter (Required)	SMTPSERVER	The name or TCP/IP address of the SMTP email server the system uses to send email.
Global Parameter (Required)	SMTPUSER	The username from which email in the system originates.
Global Parameter (Required)	TEMPLATEVERSION	Current version number of the dynamic templates in the environment. Whenever you add to or change the input schedule or report dynamic templates, you should increment this version number so that users automatically receive the new templates when they log into this environment. You can also reset the template version from the Web Client.

Table 8.2 Parameters Defined in the IMG in the SAP NetWeaver BW System for SAP BPC (Cont.)

You will now see how to set the parameter for collecting statistics for an SAP BPC model. Follow these steps to set the parameter for an SAP BPC model to turn the statistics on:

1. Log in to the SAP NetWeaver BW system. Execute Transaction SPRO, and then click on the SAP Reference IMG menu option. In the IMG structure displayed, under Configuration Parameters in the Planning And Consolidation hierarchy, click on Set Model Parameters to maintain the parameter for a model (Figure 8.9, ❶).
2. A new dialog box will prompt you to select the environment and model to set the parameter. Using the dropdown box, select the desired environment and model (Figure 8.9, ❷). Then click on Execute (Figure 8.9, ❸).

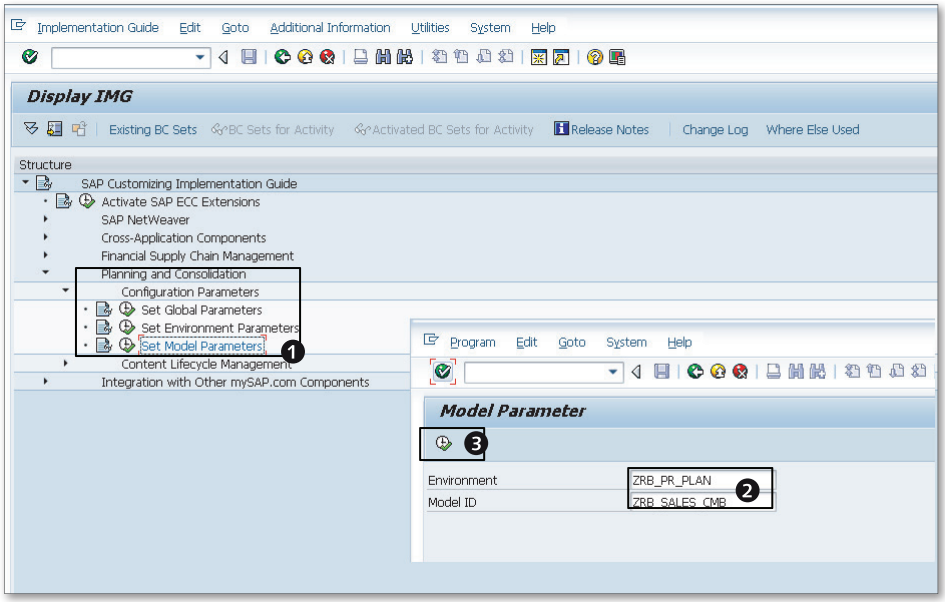


Figure 8.9 SAP BPC Parameter to Turn On Statistics—Part A

3. You will see a list of parameters defined for the model specified. To create a new parameter, click on the Create menu icon (Figure 8.10, ❹). In the new dialog box, enter "BPC_STATISTICS" as the parameter name, and then click on the checkmark icon (Figure 8.10, ❺ and ❻). The system will prompt a new dialog box to set the value for the parameter. We want to turn the parameter value to On. After entering the parameter value, click on the checkmark icon (Figure 8.10, ❼ and ❽).

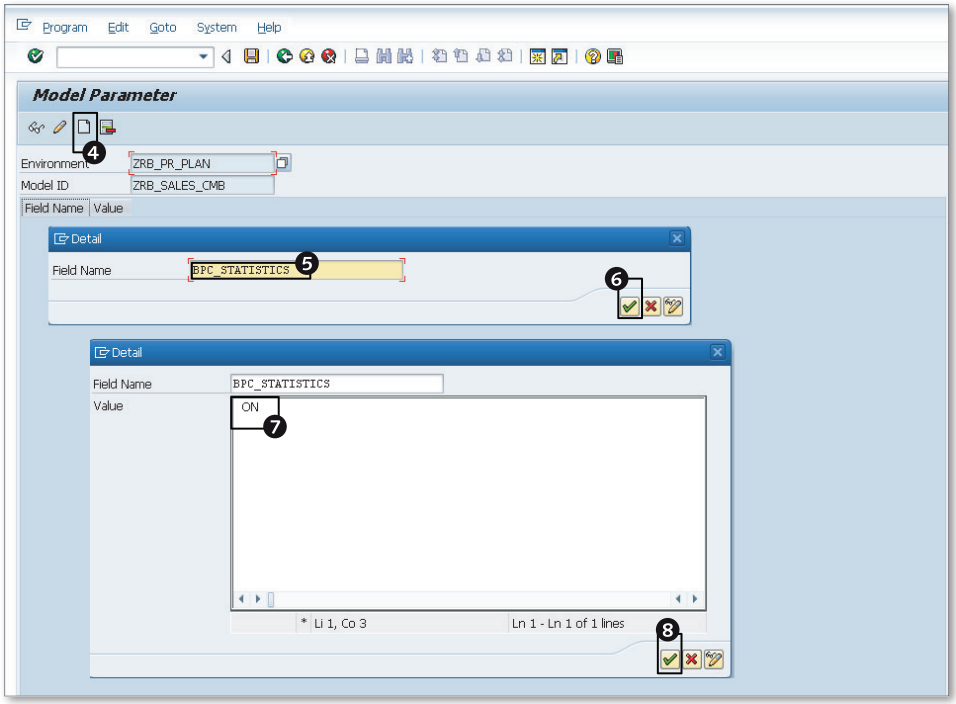


Figure 8.10 SAP BPC Parameter to Turn On Statistics—Part B

The next screen displays the parameter created for the model (Figure 8.11).

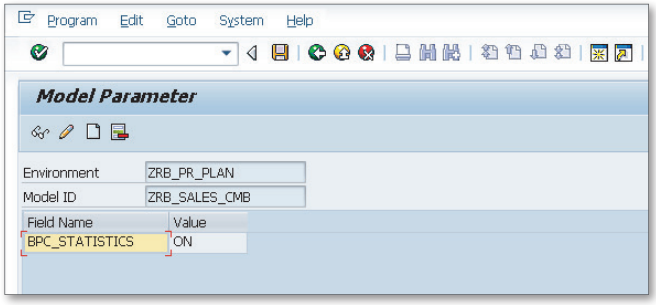


Figure 8.11 SAP BPC Parameter to Turn On Statistics—Part C

4. You have now created SAP BPC parameters to turn the statistics On for a model. The system will now create statistics data when a report, input form, or logic script is executed for that model.

In the next section, we'll discuss the steps involved in collecting statistics in SAP BPC and the benefits this offers.

8.4 Statistics

The statistics tool in SAP BPC is designed to measure the performance of SAP BPC models and to gather information about how the system is used. It is similar to the tool used in SAP NetWeaver BW for collecting statistics to analyze the time taken to load and report data on SAP NetWeaver BW objects.

The statistics data collected allows you to measure load times and retrieve data. Using these statistics, you can identify bottlenecks and take timely action before they become bigger issues. This is a valuable tool that can help with maintaining a smoothly running SAP BPC system.

The statistics tool can record the time taken to execute the following modules inside a model:

- Shared query engine: Display the time taken to retrieve data from an InfoCube for a given report.
- Write back: Measure the time taken to update data in an InfoCube.
- Logic script: Track the time taken to execute logic script.

The default value for the `BPC_STATISTICS` parameter is Off, but it must be set to On (at the model level) before the system can collect statistics for a model. We saw how to turn statistics On for a model in the previous section when we discussed administration parameters.

The statistics recorded are stored as header and detail records in SAP NetWeaver BW. The header data represents user actions, and the detail data represents the events associated with the actions. For example, the header data may represent the user action of sending data for update in SAP BPC; the detail data then represents the individual events associated with this action—such as checking whether the user is authorized to make changes and checking the work status to confirm that the data is not locked. All of these represent the detail events associated with the action initiated by the user.

Table UJ0_ACTION lists the actions for which you can collect statistics. You can collect statistics on the following types of actions:

- Read data through Shared Query Engine (SQE)
- Run logic script
- Write back data

Different types of events are associated with an action. Table UJ0_EVENTS displays the list of events that is used in SAP BPC.

The following are some of the common events during the course of writing back data:

- Check security
- Check validation
- Check work status
- Obtain concurrency locks

After the web parameter for collecting statistics for a model is set to On for a model, statistics data is collected in Tables UJ0_STAT_HDR and UJ0_STAT_DTL.

After the parameter for collecting statistics is set to On, you can view statistics on a model by using Transaction UJSTAT (Figure 8.12) to select the data based on the GUID, user, start date, start time, action, environment, and model:

- Statistics Session: A system-generated value used as the key for the action or the event under which it is stored
- User ID: The user who initiated the action or event
- Statistics Start Date: The start date of the action or event
- Statistics Start Time: The start time of the action or event
- Action ID: The action performed
- Environment: The environment to which this record pertains
- Model: The model driving the event or action

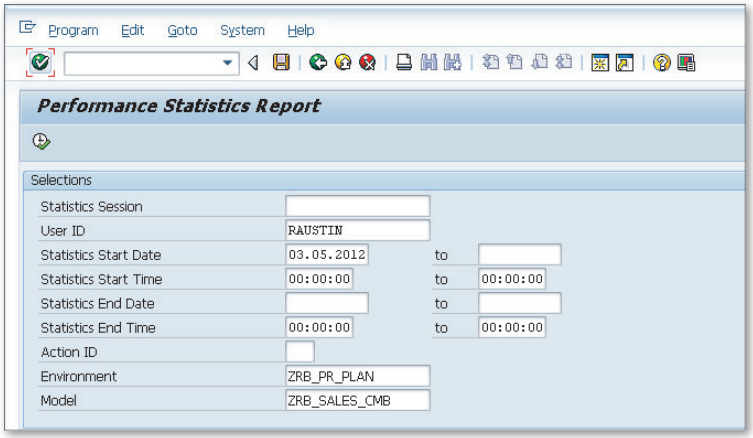


Figure 8.12 Selecting Filter Criteria for Displaying Statistics

Figure 8.13 shows how the performance statistics report will be displayed after you execute Transaction UJSTAT.

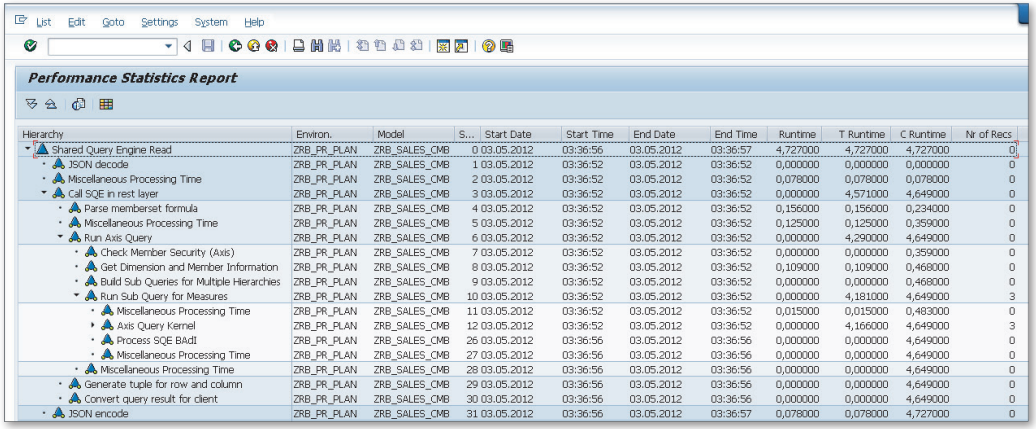


Figure 8.13 Displaying Statistics

When you have finished analyzing the statistics, you can set the `BPC_STATISTICS` parameter to Off or delete this parameter for the model. This will reduce the overhead in collecting the statistics when it is not required.

You can delete all statistics for an environment or model using the program UJO_STATISTICS_DELETE (Figure 8.14). It is not possible to selectively delete statistics records from an environment or model.

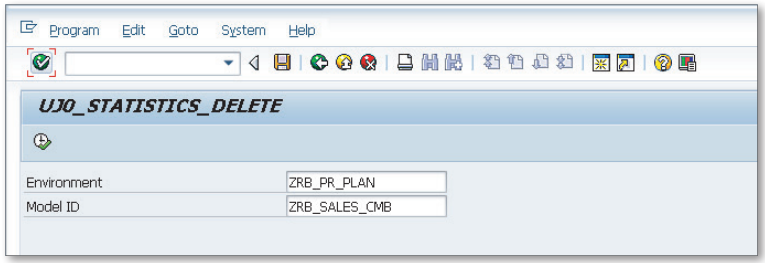


Figure 8.14 Deleting Statistics Data

In this section, we discussed how to collect and view statistics for a model. In the next section, we'll explain how to turn on the audit functionality and view audit information.

8.5 Audit

The audit functionality in SAP BPC provides the framework for monitoring changes to model configuration and data. SAP BPC for NetWeaver supports these two categories for auditing and provides the functionality to safeguard the SAP BPC system. There are two types of audits:

- Activity audit: This audit type is used to record changes to metadata. The audit captures administration activity, business rule changes, business process flow (BPF), and dimension activity inside an environment.
- Data audit: This audit type captures changes to transaction data and is applicable for a model.

The configuration for activating audits is enabled in the SAP BPC Web Client. The Manage Audit and View Audit settings tasks provide the access to manage audits. To activate an activity or data audit, select the Audit view under the Features domain in the Administration workspace (Figure 8.15, ❶). The pane to the right displays the options to configure an activity audit for the environment in the top

and a data audit for models in the bottom. Click the EDIT menu option to select the options to enable an activity audit (Figure 8.15, ❷ and ❸). Select a model in the bottom, and click on the EDIT menu to set the options for a data audit (Figure 8.15, ❹). The new dialog box prompts you to specify the option for a data audit (Figure 8.16).

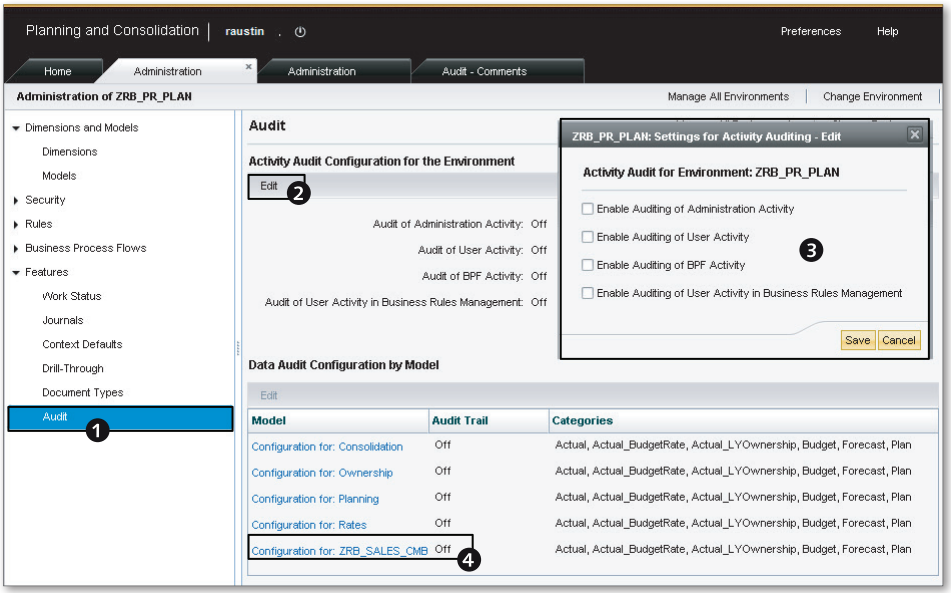


Figure 8.15 Settings for an Activity Audit

You can perform the following actions for recording an activity audit:

- Add, modify, delete, or copy operations for environments, models, and dimensions
- View user activity (work status locks)
- Record activities for BPF
- Record activities for Business Rules Management

Activity audit data is stored in Table UJU_AUDACTHDR and Table UJU_AUDACDET. Data audit data is stored in Table UJU_AUDDATAHDR, Table /1CPMB/KIABGAD, and Table /1CPMB/KIGTQAD. You can display failed audit logs using Transaction SLG1.

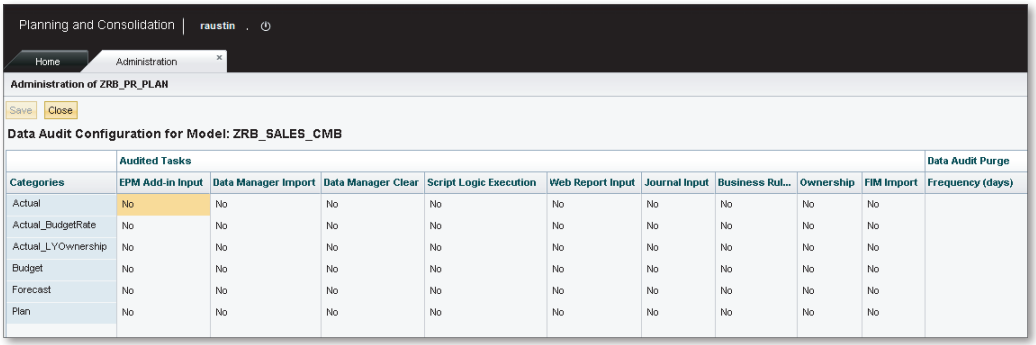


Figure 8.16 Settings for a Data Audit

You can also archive audit data. Two standard process chains are provided to do this—one for archiving activity audit data and another for archiving data audit data. The two process chains available for this purpose are `/CPMB/ARCHIVE_ACTIVITY` and `/CPMB/ARCHIVE_DATA`. Reporting is not possible on audit data that has been archived.

Standard reports are available to display audit data in SAP BPC, and you can use a filter criteria when reporting the data. To access reports, select the AUDIT view in the HOME workspace in the Web Client (Figure 8.17). A variety of reports can be run from the menus listed there.

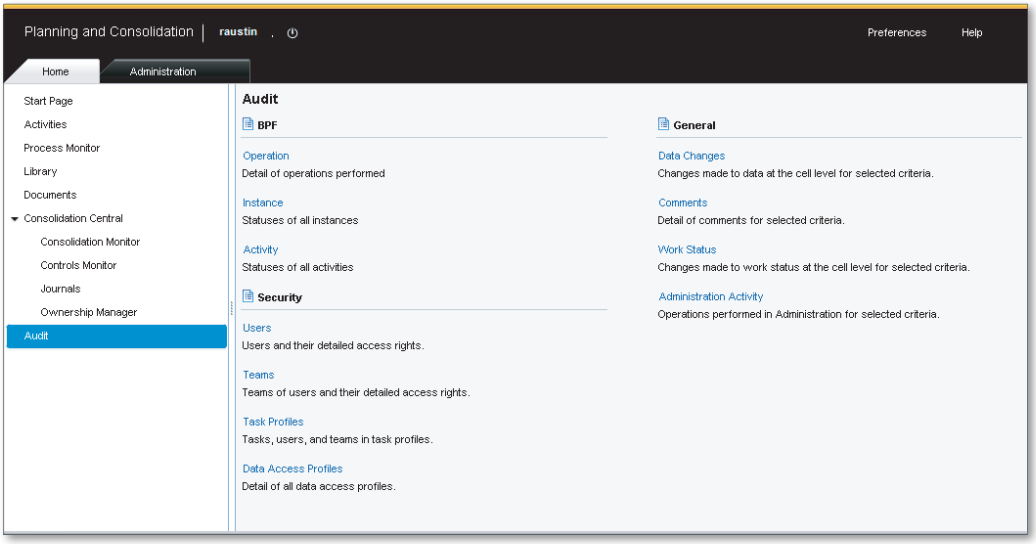


Figure 8.17 Audit Reports

You now know how to enable and view audit data. In the next section, we'll explain how to use the content library to create and maintain unstructured documents.

8.6 Documents

You access the Documents view via the Home workspace in the SAP BPC Web Client and use it for creating and sharing unstructured data such as documents and HTML. Documents are loaded via this interface, and the file types you can store in the Documents view are specified in the `DEFAULT_EXTENSIONS` SAP BPC parameter for an environment in the SAP NetWeaver BW system. You should have the appropriate access to upload and view documents; specifically, you need access to the Edit Documents and Administer Documents tasks to create and manage documents.

8.6.1 Accessing the Documents

You access the documents from the document library by clicking on the Documents view in the Home workspace. Existing documents in the document library are listed with the following information:

- Title
- Created by
- Creation date
- Type
- Subtype

8.6.2 Posting a Document

Follow these steps to post a document:

1. Click on the Documents view in the Home workspace.
2. Click on the Add menu option under the Documents view in the Home workspace (Figure 8.18, ❶). In the dialog box that opens, choose Document to add the document (Figure 8.18, ❷). Click on the Document tab to define the file you want to add to SAP BPC. In that Document tab, select File • Browse to select the file. Enter a title and description for the document. Select a Type and

Sub-type for the document from the dropdown box. You can post documents to designated locations using this subtype option.

3. In the Context tab, select the model to which the document should be associated. (Figure 8.18, ❸). If you set the model context, you can control the document via work status settings and lock down documents that have the model context specified.

4. In the Sharing tab (Figure 8.18, ❹), you can opt to select the users and teams who will have access to the document. Send an email to users who have access to this document after the document has posted.

5. Click on the OK button to add the document (Figure 8.18, ❺).

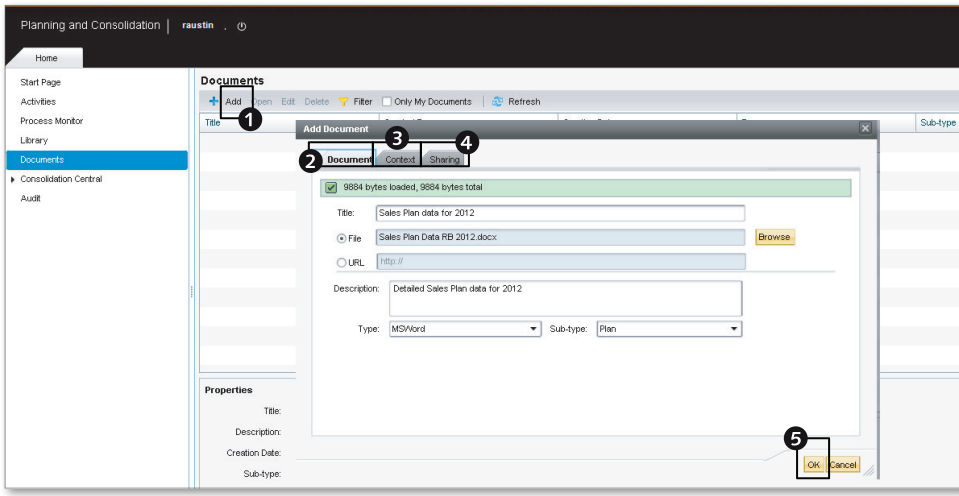


Figure 8.18 Posting a Document

You now know how to post a document to the content library. Next, we'll explain how to edit the properties of a document in the content library.

8.6.3 Editing Document Properties

To edit document properties in the document library, select the document, and click on the Edit menu option (Figure 8.19). For documents, you can edit the title, type, subtype, model context, and team access. After you make changes, confirm by clicking on the OK button in the Edit Content dialog box.

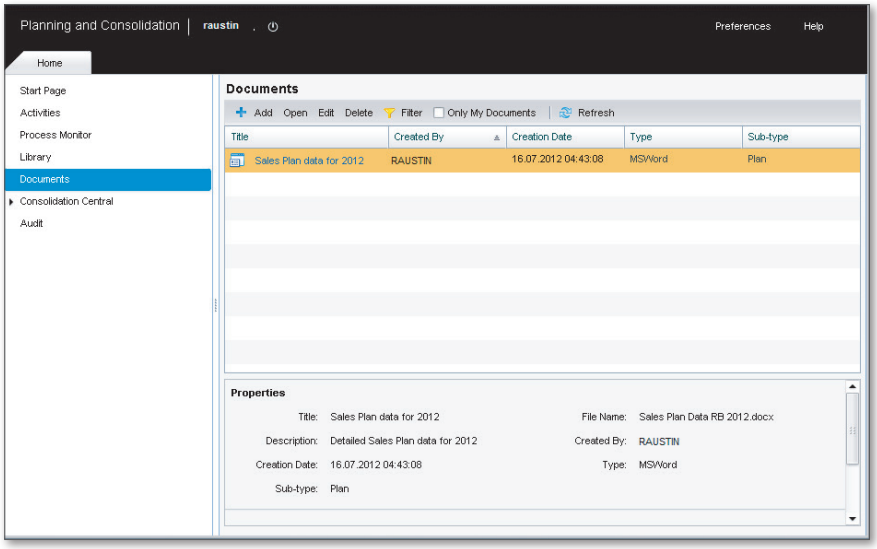


Figure 8.19 Editing a Document

8.6.4 Add a Web Page to Documents

Follow these steps to add a web page to the content library:

1. Click on the ADD menu option in the DOCUMENTS view in the HOME workspace. This opens the ADD DOCUMENT dialog box in which you can add the document (Figure 8.20).

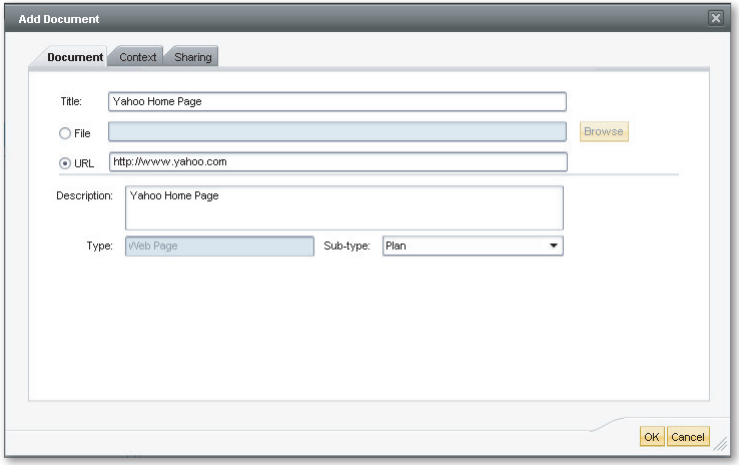


Figure 8.20 Creating a Web Page

2. The process to create a web page is similar to adding a document. Select URL as the option to create the web page.

   In this example, we'll include a link to Yahoo!'s home page as a web page. Select the URL option, and enter the URL for the Yahoo! home page. Specify a TITLE and DESCRIPTION for the web page. Optionally, select the SUB-TYPE where the web page should be located.

3. Specify the model context under the CONTEXT tab. The users or teams who will have access to the web page are set in the SHARING tab.

4. After you click OK, the web page is saved.

Under the menu option in the documents view, you have additional options for managing the documents:

- OPEN
  Opens the selected document or web page.
- DELETE
  Deletes one or more selected documents.
- FILTER
  Specifies a criterion to limit the documents displayed.
- ONLY MY DOCUMENTS
  Displays only the documents created by the user.

In this section, we explained how to use the DOCUMENT view to post, maintain, and view documents. In the next section, we'll review concepts related to managing security within SAP BPC.

8.7 Security

In this section, we'll discuss how to configure users and how to define their security levels. SAP BPC provides a simple interface to define task and data access profiles. Task profiles authorize the actions a user can perform, and member profiles define access on dimensions that are marked as secure in a model. SAP BPC supports the role concept to extend security based on a user's role in an organization. In this section, we'll discuss the terms used in the context of security.

8.7.1 Users

A user should be created in the SAP NetWeaver BW system using Transaction SU01 before being added as a user in an environment in SAP BPC. The user can then be added as a user into an environment in SAP BPC using the Web Client. The user should be granted access to perform tasks inside an environment.

8.7.2 Team

A team is a group of users. Multiple users will use a model and each user can play a different role, based on his responsibilities in the organization. A team is helpful when you need to provide the same access to a group of users. When a team is created, a folder to store reports and input forms for the team is also created. One user in the team can be assigned as a team leader. A team leader has additional access to save reports and input forms to the team folder.

8.7.3 Task Profile

The types of activities a user can perform in SAP BPC can be grouped into a task profile. Examples of activities are loading data, entering and maintaining data, executing reports, maintaining comments, creating journals, and so on. Recall that we have been discussing the task-level security access required for performing different activities. For instance, we said that a user needs access to MANAGE AUDIT to be able to configure audit settings.

The following tasks groups are available for setting security in SAP BPC:

- Administration
- Analysis and Collection
- Audit
- Business Process Flows
- Collaboration
- Comments
- Consolidations
- Data Manager

- Folder Access
- Journal
- Publish
- System Reports
- System Security
- Web Reporting
- Web Service
- Work Status

The following administration tasks are available inside the administration group:

- Manage Environments
- Manage Models
- Manage Business Rules
- Manage Data Locks and Work Status
- Manage Dimensions
- Manage Environment Status
- Manage Drill Throughs
- Manage Document Types
- View Environments
- View Models
- View Business Rules
- View Data Locks and Work Status
- View Dimensions
- View Environment Status
- View Drill Throughs
- View Document Types

Within any environment, three types of task profiles are provided by default:

- System admin: A system admin user can administer security to other users. The user can create, modify, and delete environments.
- Primary admin: A primary admin can perform all administration tasks except creating or maintaining environments.
- Secondary admin: A secondary admin can manage dimension members.

8.7.4 Data Access Profile

When a model is created, the dimensions in the model can be marked as secured. When a dimension is marked as secured, the user needs access to read and write data to the model. The member access profile indicates the dimension data that can be read or written in a model. Users should be granted access to the data access profile to read and write data for the model.

8.7.5 Security Model

The option to manage security is available in the SECURITY domain in the ADMINISTRATION workspace (Figure 8.21). The following are the steps to configure security for a model:

1. Create a user in the SAP NetWeaver BW system (using Transaction SU01).
2. Add the user to an environment in SAP BPC.
3. Create the task profiles based on roles.
4. Create member access profiles based on roles.
5. Create a team.
6. Assign task profiles to teams.
7. Assign member profiles to teams.
8. Assign users to teams.

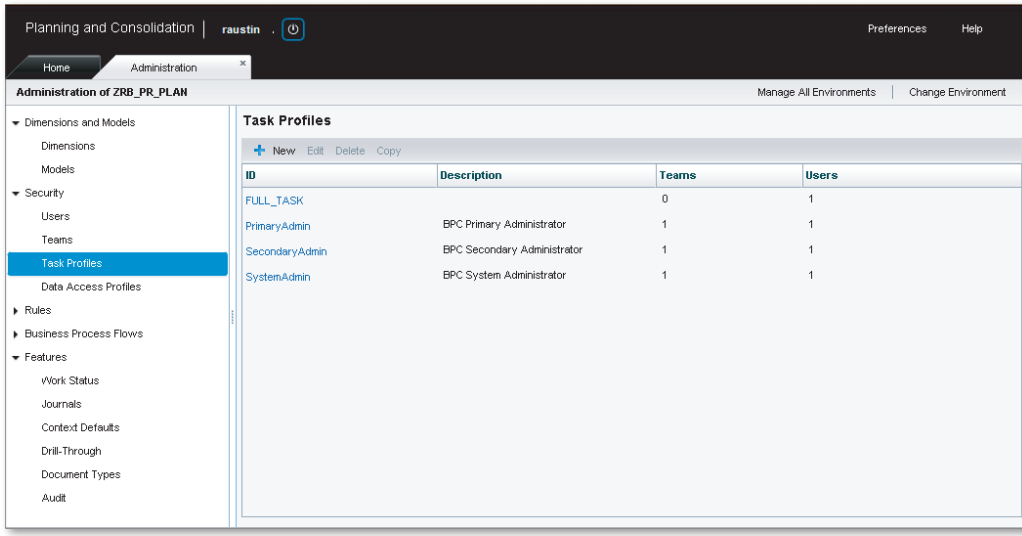


Figure 8.21 Security View

8.8 Summary

In this chapter, you learned about several key tools used in managing and maintaining an SAP BPC model. We started by looking at the steps for moving objects from the development system to the quality assurance and production systems using the improved transport process introduced in SAP BPC 10.0. We reviewed how the SAP BPC system uses concurrency locking to prevent two users from updating the same region of data simultaneously. We also reviewed the parameters that are set at the environment and model levels and how they influence the functionality of SAP BPC models. We then studied the process of collecting statistics to assess system performance and activating auditing of data to monitor changes to objects and data in the system. Furthermore, we looked at how to use the document interface to post, view, and maintain documents in SAP BPC. Finally, we discussed the concepts involved in enforcing task-level and data-level security.

In the next chapter, you'll be introduced to business rules and how to use them to support common tasks related to business planning and consolidation. You'll also learn how to create and use journal templates to post adjustments and top-side entries. In addition, we'll review general tasks that are carried out when consolidating financial data.

Consolidated financial reporting is a key requirement for meeting statutory requirements and catering to internal management reporting needs. SAP BPC helps minimize compliance risk with accurate reporting that meets statutory requirements, reduces cycle time to close books, and increases stakeholders' confidence through quick and accurate financial close.

9 Consolidation with SAP BPC

Let's turn our attention to the use of business rules such as account transformation, carry forward balance, intercompany booking, intercompany elimination (using US elimination and Eliminations and Adjustments business rules), and controls. Upcoming coverage of consolidation with SAP BPC will discuss the intercompany reconciliation process and stress the importance of the process before performing intercompany eliminations. We will also illustrate the consolidation of investments for different consolidation methods. In this chapter, we'll review topics related to performing financial consolidation using SAP BPC.

In Section 9.1, we'll introduce steps involved in the consolidation process.

In Section 9.2, we'll discuss topics related to preparing, collecting, and consolidating steps, and we'll detail the use of different types of business rules that you can configure in SAP BPC. Business rules allow users to set up standard business processes such as carry forward balances, account transformation, intercompany booking, intercompany elimination, and controls without having to develop code. In this section, we'll also discuss how to perform the necessary task of matching intercompany transactions.

In Section 9.3, we'll provide an overview of investment consolidation and discuss how the concepts apply to SAP BPC.

In Section 9.4, we'll discuss the intercompany elimination method that is used in legal consolidation, which uses the eliminations and adjustments business rule.

In Section 9.5, we'll introduce the use of journals in planning and consolidation models and describe how to configure and use journals.

To begin, let's discuss the steps involved in the consolidation process.

9.1 Steps in Consolidation

The consolidation process involves the following main steps:

- **Prepare**
  The prepare step involves setting up the basic configuration to support the process of consolidation. This step entails finalizing the dimensions and data model required for consolidation, setting up master data, and configuring business rules and security to support the process.
- **Collect**
  The collect step involves collecting data through manual entry or data upload. This step also consists of entering journals to make adjustments, validating the data, and translating data to one or more reporting currencies.
- **Consolidate**
  The consolidate step involves running consolidation functions such as carry forward balance, account transformation, intercompany matching, intercompany booking, intercompany elimination, consolidation of investments, and so on.
- **Report**
  The report step involves reporting on the consolidated data. You can leverage reporting tools such as SAP BPC for Excel and SAP BusinessObjects Dashboard and use drill-through capabilities that allow drill-through from SAP BPC to SAP NetWeaver Business Warehouse (BW) and SAP ECC systems.

In the previous chapters, we discussed some topics relating to the prepare and collect steps such as setting up the data model and loading data to SAP BPC models by either loading data from a flat file or an SAP NetWeaver BW system or by directly entering data using input forms. In this chapter, we'll discuss some of the consolidation functions and see how you can leverage business rules to perform these functions.

In the next section, we'll initially discuss business rules at a high level and later discuss their usage in detail when discussing specific consolidation functions.

9.2 Business Rules

Business rules are a key feature of SAP BPC because they enable an organization to set up rules to execute standard business processes without developing custom code. Some of the standard business processes used in planning and consolidation are as follows:

- **Currency translation**
  This is required when an organization does business in more than one country and uses multiple currencies for transacting business. When an organization uses multiple currencies, it needs to convert transactions in different currencies to one or more reporting currencies. Currency translation is used in both planning and financial consolidation models. Business rules can be set up to perform currency translation. We discussed the business rules table for currency translation in Chapter 6.
- **Carry forward**
  The carry forward of balances is an essential step in the creation of a balance sheet for an organization and refers to when the closing balance of accounts for a fiscal period is transferred as the opening balance of a subsequent fiscal period. The carry forward balance rules can also be used to close current period net income and post to the balance sheet to create a fully balanced balance sheet. The setup for this process is handled in SAP BPC with the use of a business rules table.
- **Account transformation**
  In some scenarios, accounts are consolidated or transformed into another account—an activity that is used extensively in cash flow applications. Account transformation business rules are used to aggregate values posted to specific combinations of source account, flow, category, and data source and post them to aggregated destination account, flow, category, and data source.
- **Intercompany matching/booking**
  Organizations may have entities that do business with each other. These business transactions are called intercompany transactions. It is important that both parties involved in an intercompany transaction post the transaction and amount accurately. For instance, if an entity within an organization reports an intercompany sales transaction, the trading partner associated with this transaction should post an intercompany expense transaction. The amount posted in a matching currency by each entity must be the same. This is required to produce accurate

consolidated results. In SAP BPC, we can set up an intercompany transactions matching/reconciliation process. If there are differences in amounts reported by entities, we can use the intercompany booking business rule to post/book differences in amounts.

Later in this chapter, we'll discuss the intercompany transaction matching process and discuss the use of business rules to book differences.

- **Intercompany elimination**
  This is applicable for an organization that has entities that do business with each other. A consolidated balance sheet or income statement of the organization shows transactions that relate to external companies. The transactions such as sales, cost of goods sold (COGS), or receivables and payables that result from exchange of goods or services within the organization should not be counted. In these cases, it's essential to eliminate the intercompany transactions. The intercompany eliminations process creates entries that eliminate intercompany transactions.

- **Controls**
  In Chapter 6, we discussed using Transaction UJ_VALIDATION to validate combinations of dimension members used when entering or loading data. In addition to UJ_VALIDATION, SAP BPC 10.0 version for NetWeaver introduced a new rule type called controls. Controls replaces the validation business rule that was used in the prior versions. The controls rules functionality in SAP BPC acts as a check mechanism that enables an organization to ensure consistency and accuracy of data. For example, using the controls rules feature, an organization can set up business rules that would compare balances in assets and liabilities/owner's equity accounts for a given period and report variances, if any, for that period.

The advantage of using the business rules table is that it provides out-of-the-box functionality to perform the preceding processes. This helps in reducing the total cost of ownership (TCO) of developing and maintaining applications in SAP BPC.

We'll study the use of business rules to accomplish these functions in detail.

9.2.1 Carry Forward

One of the steps in preparing a balance sheet is to take the closing balance of a fiscal period and use it as the opening balance of the subsequent fiscal period. In accounting terminology, this is called the *initialization of balances*.

To facilitate the process of carrying forward balances, SAP BPC provides the ability to automatically generate opening balances for a new fiscal period using the business rules tables. You can use this procedure to set up the carry forward process for both planning and consolidation models.

The *Flow dimension*, or the *Sub-Table dimension*, is normally used in the carry forward business rule. Balance sheet accounts, such as inventory accounts, have an opening balance at the beginning of a period, additions and transfers for the period, and a closing balance for the end of the period. The Flow dimension helps break the account into various flow types such as opening, additions, transfers, and closing.

The carry forward process requires the following dimension properties to be included in the AUDIT and CATEGORY dimensions:

- `DATASRC_TYPE` **of the AUDIT dimension**
  When you use this property, the data loaded into SAP BPC can be classified as input (`I`), manual (`M`), or automatic (`A`). The logic for carry forward in SAP BPC is set to work only for data that has the data source type set to input or manual. Automatic data sources are generated and addressed during the consolidation process.
- `COPYOPENING` **of the AUDIT dimension**
  Set this property to `Y` to perform carry forward balance for data associated with a data source. When this property is set to `N`, the associated data source is excluded from the carry forward process.
- `OPENING_DATASRC` **of the AUDIT dimension**
  Use this property to specify the data source to which the opening balances are written. If this value is set to blank, the opening balances are set to the same value as the DataSource on the source records.
- `CATEGORY_FOR_OPE` **of the CATEGORY dimension**
  You use this property when you want to use a specific category as the basis for creating opening balances. When the setting is active, the carry forward balance uses the category specified in this property for performing carry forward balances. If this property is blank, the balances are copied from the same category. For example, you can use this property when you want to copy the closing balance of actual data to the opening balance of plan data.
- `OPENING_YEAR` **of the CATEGORY dimension**
  You use the value specified in this property as the year to read values for creating the carry forward balance. If the value of this property is set to blank, the

balances are copied over from the previous year. In this property, you can either use an absolute value, such as 2011, or an offset value, such as -1.

- **`OPENING_PERIOD` of the CATEGORY dimension**
  You use the value specified in this property as the period to read values for creating the carry forward balance. If the value of this property is set to blank, the balances are copied over from the last period of the year (e.g., DEC). In this property, you can use either the absolute value or the offset value.

Table 9.1 explains the details for setting up the carry forward business rules table.

Field Name	Description
Source Account	Specify the source account to use for the carry forward balance. The value specified here can be one of the following: ▸ Base member ▸ Parent member A list of members defined and filtered using a value of the `DIMLIST` property in the Account dimension.
Source Flow	Specify the source flow to use for the carry forward balance. The value specified here can be one of the following: ▸ Base member ▸ Parent member ▸ A list of members defined and filtered using a value of the `DIMLIST` property in the Flow dimension.
Source Audit	Specify the source audit member to use for the carry forward balance. The value specified here can be one of the following: ▸ Base member ▸ Parent member
Destination Account	Specify a base member as the destination account.
Destination Flow	Specify a base member for the destination flow.
Destination Audit	Specify a base member for the destination audit dimension.

Table 9.1 Carry Forward Balance Business Rule

Field Name	Description
Reverse Sign	Select this box if you want to reverse the sign of the amount when creating the carry forward balance from the source account. This is especially used in scenarios where you do not want to carry forward balances of specific accounts. To do this, you may initially carry forward all balance sheet accounts and reverse out specific balance sheet accounts that you do not want to carry forward. This results in a net effect of those specific accounts that you do not want to carry over (that were reversed) to zero.
Audit Type	Specify all of the audit members you want to use for the conversion. This corresponds to the values set in the `Audit Type` property of the Audit dimension. You can select from the following choices: Input, Manual, or All. Note that audit members that are type `A` (Automatic) are not considered for carry forward balance conversions.
Same Period	You can select this box when the source time period is to be used as the destination time period.
Apply to YTD	You can select this box to calculate YTD values for an application that is PERIODIC.
Remark	This is a brief description of the business rule.

Table 9.1 Carry Forward Balance Business Rule (Cont.)

After the setup is complete, the `COPYOPENING` script is used to execute the logic to create the opening balances for a given period. Add the following logic in your model (if one does not already exist) and call it "COPYOPENING.LGF":

```
*RUN_PROGRAM COPYOPENING
CATEGORY = %C_CATEGORY_SET%
CURRENCY = %CURRENCY_SET%
TID_RA = %TIME_SET%
OTHER = [ENTITY=%ENTITY_SET%]
*ENDRUN_PROGRAM
```

SAP BPC provides a data package to execute the carry forward process. When the package is executed, the system prompts the user to enter category, entity, currency, and time selections for performing the carry forward process.

Let's review an example to understand how to set up carry forward balances in an SAP BPC model. We will carry forward closing balance in the Land & Buildings account as of December 2011 (2011.12) to opening balance of January 2012 (2012.01).

1. Figure 9.1 shows the dimensions for model ZRB_CONS_MODEL. They include the Flow dimension, which is used for executing carry forward balances. The data entry mode for a consolidation model is year to date (YTD).

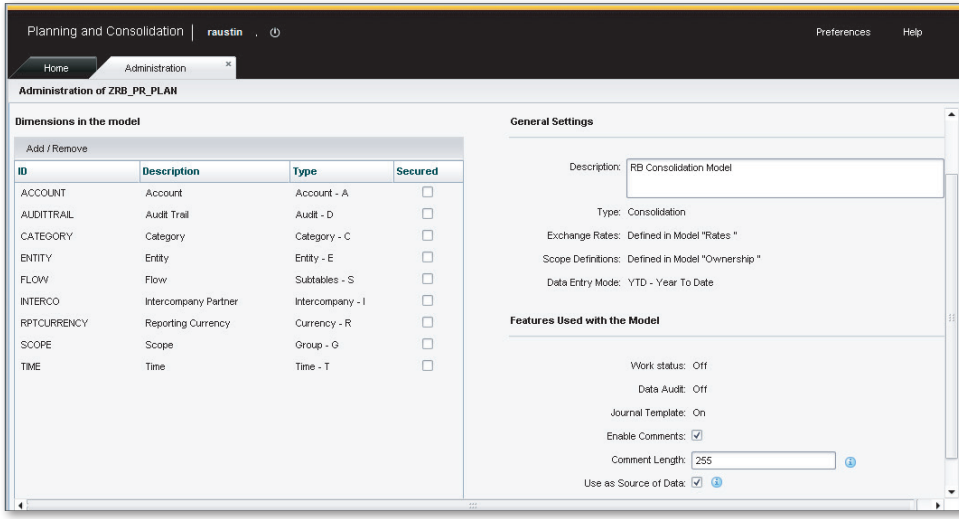


Figure 9.1 Configuring Carry Forward Balance Business Rule—Part A

2. In the Business Rules view of the Rules domain, click on the Add/Remove Rule Types menu option to check the business rules to use inside a model (Figure 9.2, ❶ and ❷). Confirm that the Carry-Forward business rule is checked in the Rule Types for model before proceeding to the next step.

3. Figure 9.3 shows the members of the Flow dimension, which are used to reflect flows such as open, movements, and close.

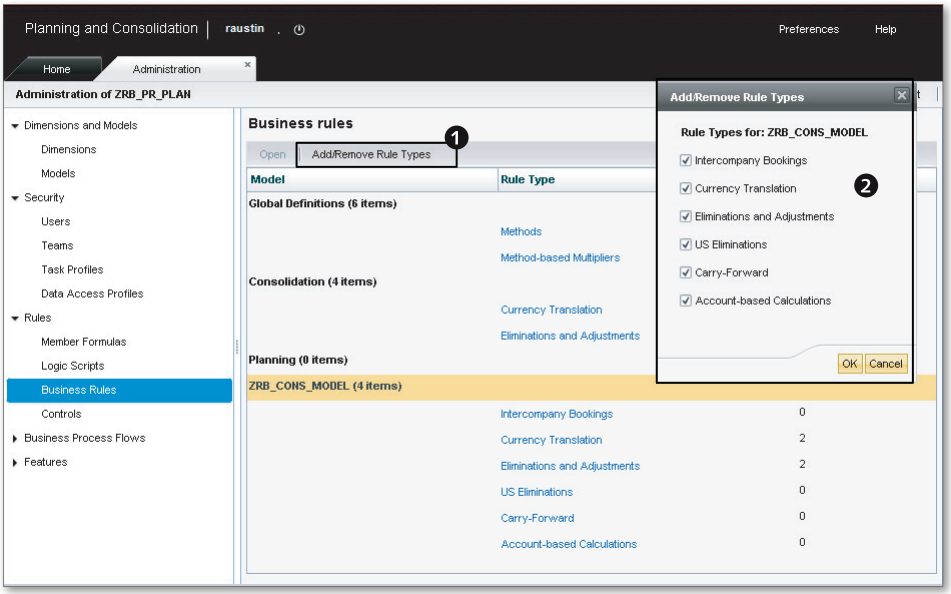


Figure 9.2 Configuring Carry Forward Balance Business Rule—Part B

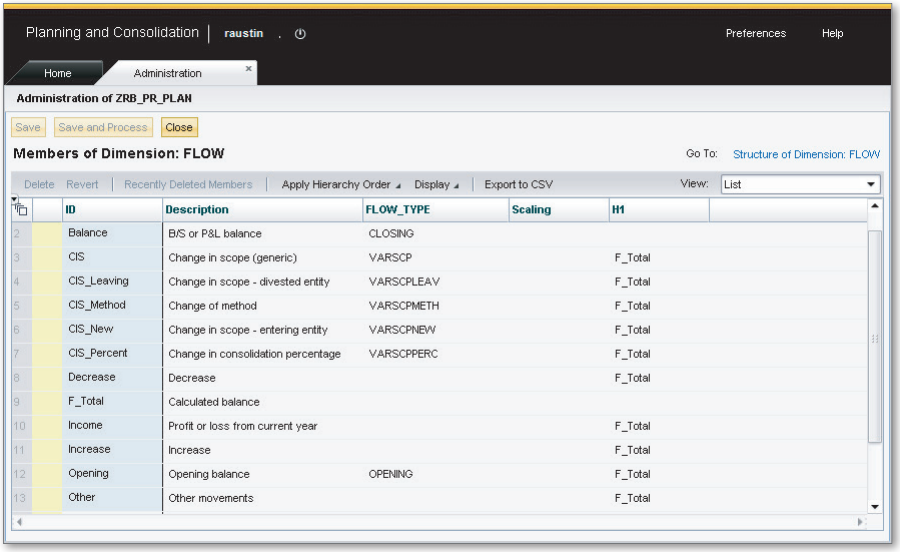


Figure 9.3 Configuring Carry Forward Balance Business Rule—Part C

4. Figure 9.4 displays the closing value for LAND & BUILDINGS for the US entity for the December 2011 period. In this example, the flow BALANCE represents the closing balance.

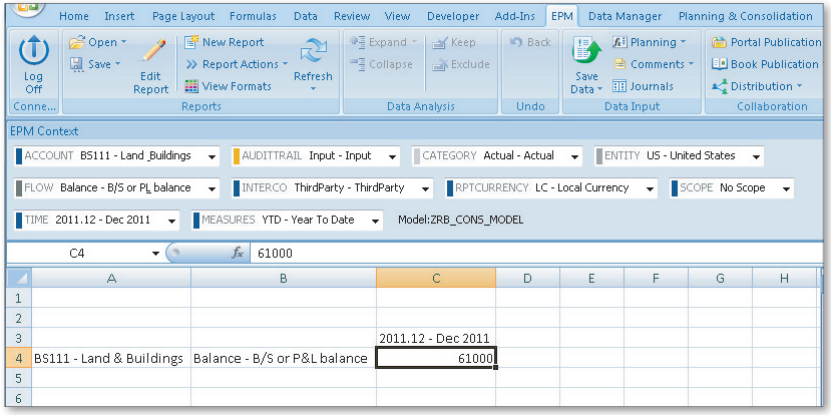


Figure 9.4 Configuring Carry Forward Balance Business Rule—Part D

5. Define the business rule to configure the carry forward balance of the LAND & BUILDINGS account, as shown in Figure 9.5. In our example, we want to copy the closing balance for this account as of December 2011 to the opening balance as of January 2012 when the carry forward balance script is executed for January 2012.

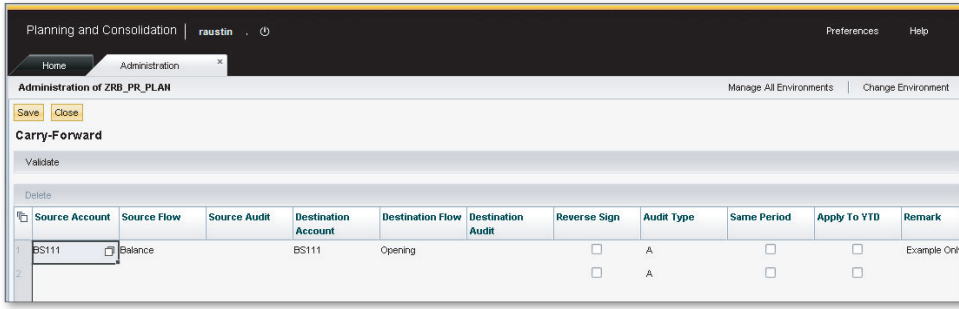


Figure 9.5 Configuring Carry Forward Balance Business Rule—Part E

6. Create the standard logic script COPY_OPENING.LGF (if one does not already exist) to execute the copy opening balance business rule, as shown in Figure 9.6.

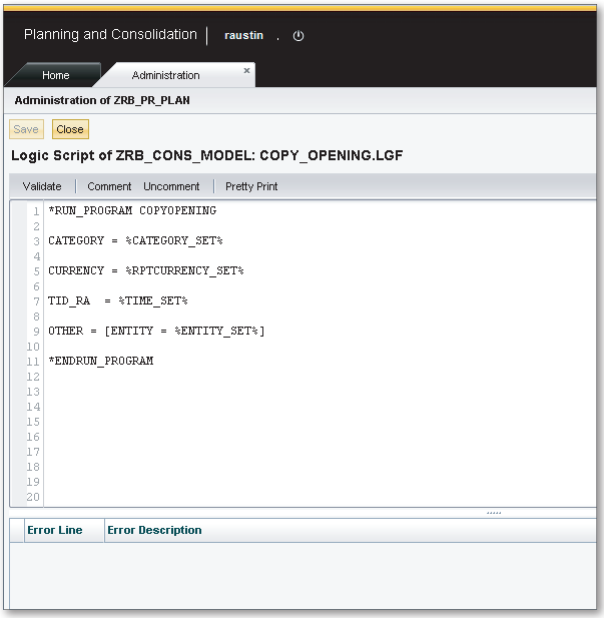


Figure 9.6 Configuring Carry Forward Balance Business Rule—Part F

7. Create the Data Manager package for executing a carry forward balance (if one does not already exist). The process chain associated with the carry forward balance is `/CPMB/OPENING_BALANCES`. Run the data package to execute the carry forward balance script, as shown in Figure 9.7.

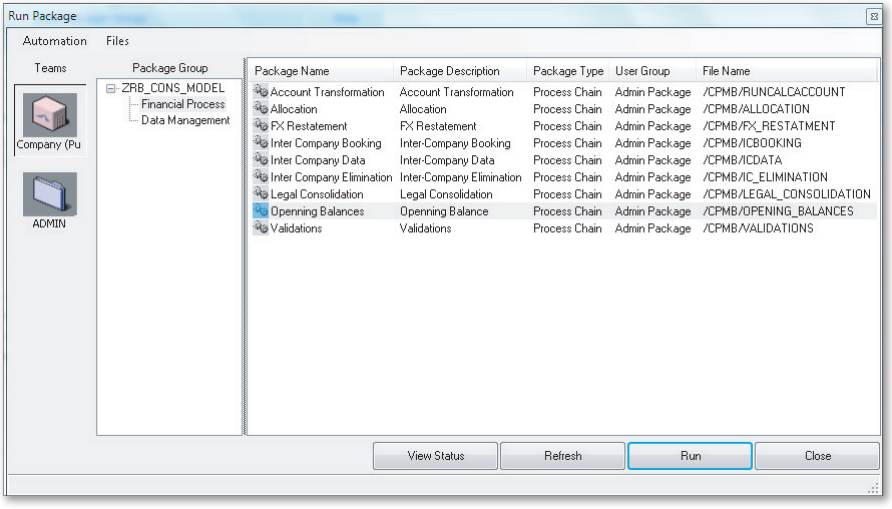


Figure 9.7 Configuring Carry Forward Balance Business Rule—Part G

8. Make the selections shown in Figure 9.8. We chose the time period January 2012 because we want to create the opening balance for that period.

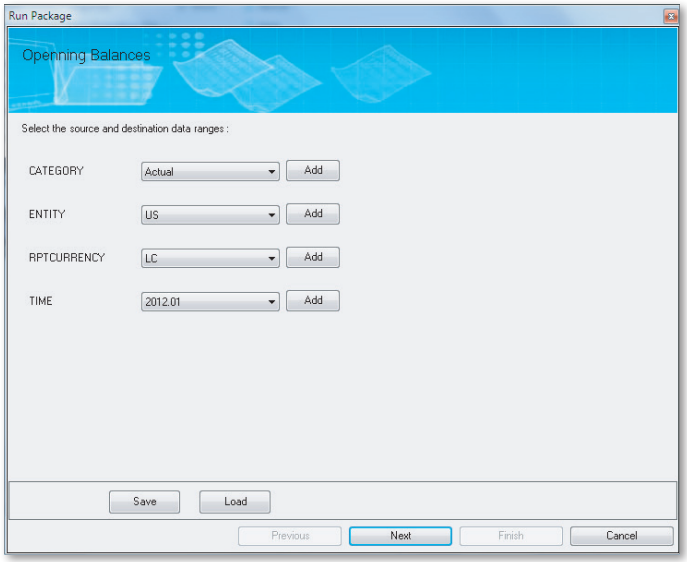


Figure 9.8 Configuring Carry Forward Balance Business Rule—Part H

9. After executing the package, report the opening balance (OPENING flow) for the LAND & BUILDINGS account, entity United States, and January 2012. The carry forward balance script has generated the opening balance entries for January 2012. The closing balance for December 2011 has been copied to the opening balance for January 2012 (Figure 9.9).

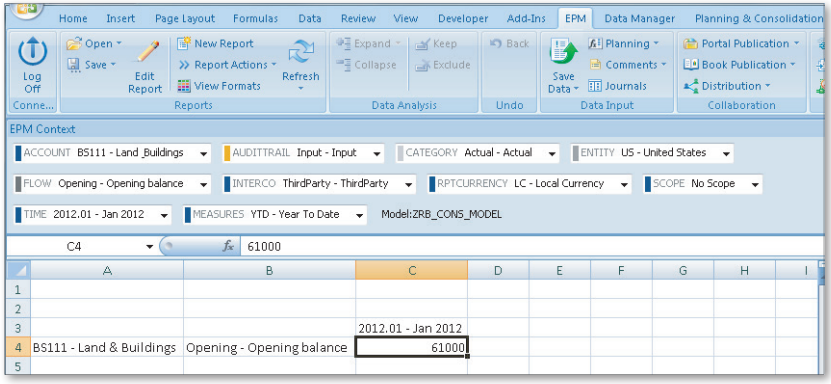


Figure 9.9 Configuring Carry Forward Balance Business Rule—Part I

You should now understand how to use business rules to perform carry forward balances. Next, we'll study the use of the account transformation business rule.

9.2.2 Account Transformation

Account transformation business rules are used to aggregate values posted to specific combinations of source account, flow, category, and audit dimension members and to post them to aggregated destination account, flow, category, and audit. The business rules allow us to perform calculations by adding or subtracting values. The rules also allow us to use reference data in other years/periods to determine amounts to post.

Table 9.2 explains the details for setting up the account transformation business rules table.

Field Name	Description
ID	Several account transformation business rules can be grouped under a specific transformation group ID. This is the ID that is used in the logic script for performing account transformation.
Description	This describes the transformation group.
Source Account	Specify the source account to use for account transformation. The value specified here can be one of the following: ▸ Base member ▸ Parent member A list of members defined and filtered using a value of the `DIMLIST` property in the Account dimension
Category	Specify the source category to use for the account transformation. The value specified here can be one of the following: ▸ Base member ▸ Parent member A list of members defined and filtered using a value of the `DIMLIST` property in the Category dimension

Table 9.2 Account Transformation Business Rule

Field Name	Description
Source Flow	Specify the source flow to use for the account transformation. The value specified here can be one of the following: ▶ Base member ▶ Parent member A list of members defined and filtered using a value of the DIMLIST property in the Flow dimension
Source Audit	Specify the audit source to use for the account transformation. The value specified here can be one of the following: ▶ Base member ▶ Parent member A list of members defined and filtered using a value of the DIMLIST property in the data source dimension
Destination Account	Specify a base member as the destination account.
Destination Category	Specify a base member as the destination category.
Destination Flow	Specify a base member for the destination flow.
Destination Audit	Specify a base member as the destination for the Audit dimension.
Reverse Sign	Select this box if you want to reverse the sign of the amount when calculating accounts. This is especially used in scenarios where you want to subtract values when performing calculations. Sometimes, you may transform values from a source account of type LEQ (stored as –ve in the database) to a destination account of type AST (stored as +ve in the database). If you want to maintain the same sign of source account for the transformation, you reverse the sign.
Source Year	This is the reference year to read values for performing account transformation. You can use an absolute value (for example, 2010) or an offset value (for example, –1). If blank, the reference data will be taken from the same year.

Table 9.2 Account Transformation Business Rule (Cont.)

Field Name	Description
Source Period	This is the reference period to read values for performing account transformation. You can use an absolute value (for example, 2010) or an offset value (for example, –1). If blank, the reference data will be taken from the same period.
Forced INTCO Member	This can be used to force a destination dimension INTCO member to contain a specific value.
Apply to YTD	You can select this box to calculate YTD values for an application that uses PERIODIC measures.
Level	This is the integer that specifies the sequence of execution of the rule.
Remark	This is a brief description of the business rule.

Table 9.2 Account Transformation Business Rule (Cont.)

Let's discuss an example to illustrate the use of account transformation business rules. Consider a YTD model that includes the account BS111 representing the Land & Buildings account. We'll use the transformation business rule to calculate YTD movements of the Land & Buildings account. The YTD movements of the Land & Buildings account may later be used in cash flow calculations. The following are the steps to calculate the movements of the account:

1. Using the carry forward business rule, carry forward the closing balance of the Land & Buildings account from the ending balance of the previous year to the opening balance of the current period. After you've performed the carry forward balance, the opening balance of Land & Buildings for the current period will be in flow OPENING.
2. We assume that the closing balance of the Land & Buildings account for the current period is available in flow BALANCE of the model. This information could have been either entered by users using input templates or loaded into the model.
3. Using account transformation, we can calculate the YTD movement to Land & Buildings by calculating the difference between the closing flow (BALANCE) and opening flow (OPENING) for the account. After the account transformation, the YTD movements of the Land & Buildings account will reside in movement flow

(MOV). Figure 9.10 contains two rules to perform the calculation. In rule 2, we've set the REVERSE SIGN field to X to subtract value.

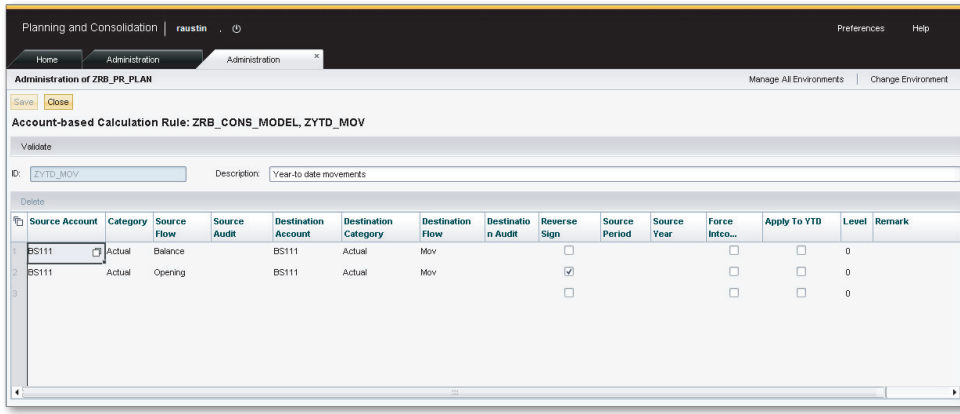


Figure 9.10 Account Transformation Business Rule

4. After we configure the business rules, the `CALCACCOUNT` script is used to execute the logic to perform account transformation for a given period. Add the following logic in your model (if one does not already exist), and call it "CALCACCOUNT.LGF":

```
*RUN_PROGRAM CALC_ACCOUNT
   CATEGORY = %C_CATEGORY_SET%
    CURRENCY = %CURRENCY_SET%
    TID_RA = %TIME_SET%
    CALC=CF
    OTHER = [ENTITY=%ENTITY_SET%]
*ENDRUN_PROGRAM
```

5. Run the Account Transformations Data Manager package to perform the account transformation. The Data Manager package will execute the `CALCACCOUNT.LGF` script. The process chain associated with the Account Transformation Data Manager package is `/CPMB/RUNCALCACCOUNT`.

6. Account transformation business rules are used extensively to calculate cash-flow accounts, such as cash flow from operations, cash flow from financing, and cash flow from investing. For example, cash flow from operations may involve starting with the net profit and adding increases to liability accounts related to working capital and subtracting increases to asset accounts related to working capital. As we saw in the preceding example, we may need to use account transformation

business rules in conjunction with the carry forward balance business rule to calculate specific values.

Note

When using account transformation business rules, if there are dependencies in your calculations, separate the independent and dependent rules using the ID or the transformation group. Sequence the execution of the rules for each group based on dependencies—first, the independent rules and then the dependent rules.

Next, we'll explain how to set up the configuration to perform intercompany matching and booking.

9.2.3 Intercompany Matching/Booking

You use the intercompany elimination process when an organization is composed of entities that do business with each other. During consolidation, the intercompany transactions should be eliminated.

For the intercompany elimination process to work correctly, it is important that you match intercompany transactions. To illustrate this, let's first understand how eliminations work; specifically, we'll discuss U.S. eliminations. U.S. elimination is an easy way to do one-sided eliminations. You could use it when ownership eliminations (consolidation of investments) are not required.

Consider an intercompany transaction between two entities, E1 and E2, within the organization. We'll assume that the transactions are in the same currency. Let's say E1 sells goods to E2 and thus posts an intercompany revenue transaction. At the same time, E2 would post an intercompany expense/COGS transaction for the same amount. During elimination, the revenue transaction of E1 and the expense transaction of E2 are eliminated. The elimination entry is posted to the elimination entity associated with the first common parent (in the entity hierarchy) of E1 and E2. When the revenue transaction of E1 is eliminated, the process debits revenue and credits the plug account (also called elimination account) associated with the revenue account. When the expense transaction of E2 is eliminated, the process debits the plug account associated with the expense account and credits the expense account. Assuming that the same elimination account is associated with the revenue and expense accounts, if amounts entered by E1 and E2 are the same, the elimination account would net to zero as a result of the eliminations. A consolidated report

that displays data for the group consisting of entities E1, E2, and the elimination entity would report transactions that have eliminated intercompany transactions. Any mismatch in transaction amount entered by the two entities involved in the intercompany transaction may lead to incorrect consolidated results.

This issue underscores the need for both parties of an intercompany transaction to enter the transaction amount accurately. Complexity may arise if the two entities deal with two different currencies. It is important that the exchange rates used to convert the transaction values to a matching/group currency are such that, after currency conversion, the transaction values posted by the two entities match.

SAP BPC provides functionality to match intercompany transactions and automatically book differences using the IC (intercompany) booking business rules. This matching process makes available transactions entered by trading partners that relate to a user's entity. These matching transactions entered by trading partners are posted into the user's entity under a specific data source. This allows the user belonging to a specific entity to view transactions entered by trading partners even though the user may not have security access to the trading partner's entity/entities.

The intercompany matching process can either be set up in a separate model or can be done within an existing model that contains intercompany transactions. The matching process should be done using a matching currency.

The output of the intercompany matching process is a report that displays the intercompany transaction amount in a matching currency that shows what the entity has entered and what the trading partner(s) has entered.

The following are the steps to perform intercompany matching and booking:

1. Import or enter data in the local currency.
2. Translate data in local currency to the matching/group currency.
3. Perform the initial setup needed to perform intercompany matching.
4. Execute the Data Manager package ICDATA to generate entries needed to perform intercompany matching.
5. Run the intercompany matching report.
6. Correct mismatches manually or by using the automatic booking (`ICBOOKING`) function.

As stated previously, the first two steps involve loading transaction data in local currency and converting the transaction to matching/group currency. Next, we'll detail the initial setup required to perform intercompany matching.

Setup for Intercompany Matching

Consider these guidelines for performing U.S. intercompany elimination in SAP BPC:

- To support intercompany elimination, the SAP BPC model should include the Intercompany dimension. The dimension type is I.
- The Intercompany dimension should include a property called `Entity`, and the value of this property should match the member ID of the Entity dimension.
- The Account dimension should include the `ELIMACC` property, and the value of this property should match the member ID of the Account dimension.
- The Entity dimension should include an `ELIM` property, the value of which can be `Y` or `N` (or blank). The elimination process reads all entries for entities that have the value of this property set to `N` (or blank) and then posts eliminations, if applicable, to entities that have the `ELIM` property set to `Y`.

For performing intercompany matching, perform the following setup:

- Include the property `PINTCO` in the Account dimension. Set the values of all intercompany matching accounts (base members) to `Y`. These accounts will be considered for intercompany matching.
- Create a new hierarchy of intercompany accounts used in matching for performing intercompany matching and booking. Set the nodes in a manner that matching can be accomplished.

  For instance, assign accounts associated with Notes Payable and Notes Receivable under a matching node. When an intercompany matching transaction takes place, one entity would be posting Notes Payable and the other entity would be posting Notes Receivable. By assigning these two sets of accounts under a matching node, you can perform intercompany matching and booking at that level. A sample hierarchy in Figure 9.11 displays the hierarchy nodes used for intercompany matching. Intercompany accounts are assigned to these nodes.

ID	EVDESCRIPTION	PARENTH	GROUP	ACCTYPE	PINTCO
CTLCIS0000	Intercompany Check TOTAL			AST	
CTLICBFR00	Intercompany Short Term	CTLCIS0000		AST	S
CTLICBFR01	Intercompany Short Term - Receivables	CTLICBFR00		AST	
CTLICBFR02	Intercompany Short Term - Payables	CTLICBFR00		LEQ	
CTLICEND00	Intercompany Long Term	CTLCIS0000		AST	S
CTLICEND01	Intercompany Long Term - Receivables	CTLICEND00		AST	
CTLICEND02	Intercompany Long Term - Payables	CTLICEND00		LEQ	
CTLICEXP00	Intercompany Operating	CTLCIS0000		INC	S
CTLICEXP01	Intercompany Operating - Revenues	CTLICEXP00		INC	
CTLICEXP02	Intercompany Operating:- Expenses	CTLICEXP00		EXP	

Figure 9.11 Intercompany Account Hierarchy

- Include the property `IC_ORIGINE` in the Audit dimension. Set the value of this property of all members that will be used as sources for matching to `I`. For example, set this property for the input member to `I`.
- In addition, create new audit members as specified in Figure 9.12. Note that members DEBIT1, CREDIT1, TOTAL1, DEBIT2, CREDIT2, and TOTAL2 house the intercompany matching data. These members are used in the intercompany matching report and do not fall under the hierarchy used for consolidated reporting.

ID	EVDESCRIPTION	PARENTH1	IS_CONVERTED	IC_ORGINE
TOTAL1				
DEBIT1	My Asset/Income	TOTAL1	N	D
CREDIT1	Their Liability/Expense	TOTAL1	N	C
TOTAL2				
DEBIT2	Their Asset/Income	TOTAL2	N	D2
CREDIT2	My Liability/Expense	TOTAL2	N	C2
ICDIFF	Intercompany Differences		G	

Figure 9.12 Audit Members That House Intercompany Matching Entries

The preceding setup is required before you execute the `ICDATA` process to generate intercompany matching entries. Let's move on to the details of the `ICDATA` matching process.

Intercompany Matching—ICDATA

The `ICDATA` process generates the intercompany entries recorded by a user's entity and the entries pertaining to the user's entity that were recorded by the trading partners. These entries are all posted under the user's entity that will be doing the matching. So the `ICDATA` process makes available transactions entered by trading partners that relate to a user's entity even though the user may not have access to the trading partner's data. The data is stored in separate DataSources listed earlier in Figure 9.12.

After performing the initial setup discussed in the previous section, run ICDATA.LGF via a Data Manager package called ICDATA. The process chain associated with the ICDATA Data Manager package is /CPMB/ICDATA. This process generates the entries needed for performing intercompany matching. The code in ICDATA.LGF is as follows:

```
*RUN_PROGRAM ICDATA
        CATEGORY = %C_CATEGORY_SET%
           CURRENCY = %CURRENCY_SET%
           DATASRC = %DATASRC_SET%
           TID_RA = %TIME_SET%
           ENTITY = %ENTITY_SET%
           TYPE = 'I'.
*ENDRUN_PROGRAM
```

After you run the ICDATA Data Manager package, run the intercompany matching report to perform reconciliation.

Intercompany Matching Report

The intercompany matching report is used to reconcile intercompany transactions entered by an entity with the transactions entered by its trading partners. Create a report that displays for an entity, the trading partner (INTCO) and intercompany account in the rows and the members of the DATASRC dimension that houses the matching entries (namely, DEBIT1, CREDIT1, TOTAL1, DEBIT2, CREDIT2, and TOTAL2) in the columns. The DataSources displayed in the columns represent the entity's postings and the corresponding trading partner's postings. Figure 9.13 displays the format of the report.

	Choose Entity:	E1							
	Choose Period:	2008.AUG							
			DEBIT1	CREDIT1	TOTAL1	DEBIT2	CREDIT2	TOTAL2	
	Intercompany Entity (ICP)	Account	My Asset or Income	Their Liability or Expense	Total 1	Their Asset or Income	My Liability or Expense	Total 2	
	I_E2 - Entity 2	CTLICBFR00 - Intercompany Short Term	84	(84)					
	I_E2 - Entity 2	**CTLICBFR01 - Intercompany Short Term: Receivables**	**84**		**84**				
	I_E2 - Entity 2	IC1210 - Inter Company A/R	84		84				
	I_E2 - Entity 2	**CTLICBFR02 - Intercompany Short Term: Payables**		**84**	**84**				
	I_E2 - Entity 2	IC2210 - Inter Company A/P		84	84				

Figure 9.13 Intercompany Matching Report

In this report, the users chose Entity and Time to perform matching. You're trying to match intercompany short-term receivables and payables transactions for entity

E1. Entity E1 has posted an intercompany A/R transaction for $84 specifying entity E2 as the trading partner. Entity E2 has posted an intercompany A/P transaction for $84 specifying entity E1 as the trading partner. Because the intercompany transactions entered by the two entities match, the TOTAL1 value at the CTLICBFR00 level (matching node level) is 0.

If intercompany transactions match at the matching node level, the value of TOTAL1 and TOTAL2 will be 0. If there are mismatches, the value of TOTAL1 and TOTAL2 at the matching node level will not be 0, and the mismatch can either be corrected manually or booked automatically by using the `ICBOOKING` function. In the next section, we'll see how to set the business rules for automatic booking of mismatches.

Automatic Booking of Intercompany Transactions—Business Rule

Mismatches may be identified during intercompany transaction reconciliation. The `ICBOOKING` function is available to automatically book mismatches of intercompany transactions. Table 9.3 details the fields in the business rules table.

Field Name	Description
Parent Matching Account	Specifies the hierarchy node at which, if a mismatch occurs, automatic posting needs to be done.
Type	Specifies how the automatic booking needs to be done. The possible values are Seller Rule, Buyer Rule, and Greatest Amount. If Seller Rule is specified, the seller will receive the variances. If Buyer Rule is specified, the buyer will receive the variances.
Booking Destination DataSource	Specifies the audit member to which differences will be booked.
Maximum Booking Amount	Threshold amount above which the automatic booking is not posted automatically.
Debit Account	Base account to be debited.
Debit Flow	Base flow to be debited.
Debit INTCO	Base INTCO dimension that will be debited.
Credit account	Base account to be credited.

Table 9.3 ICBooking Business Rule

Field Name	Description
Credit Flow	Base flow to be credited.
Credit INTCO	Base INTCO dimension that will be credited.
Destination Flow	Specifies a base member for the destination flow.
Destination Data Source	Specifies a base member as the destination data source.
Remark	A brief description of the business rule.

Table 9.3 ICBooking Business Rule (Cont.)

Let's discuss an example to illustrate the usage of this business rule table. Account hierarchy node CTLICBFR00 represents Intercompany Short Term Receivables/Payables. If a mismatch occurs at this level, we would want to debit account ICDIFF01 and credit account IC2210 (IC Accounts Payable) for the mismatched amount. We would also want to post the entry to data source ICDIFF. We'll create the business rule entry in Table 9.4 to generate the automatic posting.

Rule Fields	Rule Values
Parent Matching Account	CTLICBFR00
Type	Seller Rule
Booking destination data source	ICDIFF
Maximum booking amount	99,999,999
Debit account	ICDIFF01
Debit flow	Balance
Debit INTCO	
Credit account	IC2210
Credit flow	Balance
Credit INTCO	
Remark	Receivable/payable

Table 9.4 ICBooking Business Rule (Example)

Automatic Booking of Intercompany Transactions—Logic Script

After setting up the business rule, run ICBOOKING.LGF via a Data Manager package called ICBOOKING. The process chain associated with the ICBOOKING Data Manager package is /CPMB/ICBOOKING. This process automatically creates the postings to correct mismatches in intercompany accounts. The code in ICBOOKING.LGF is as follows:

```
*RUN_PROGRAM ICBOOKING
        CATEGORY = %C_CATEGORY_SET%
           CURRENCY = %CURRENCY_SET%
           TID_RA = %TIME_SET%
           ENTITY = %ENTITY_SET%
*ENDRUN_PROGRAM
```

In this section, we've explored the need to perform intercompany matching and have seen how to use the ICDATA and ICBOOKING functions to reconcile intercompany transactions and correct mismatches. The intercompany transaction matching process is a prerequisite to performing intercompany elimination, which we'll discuss next.

9.2.4 Intercompany Eliminations (U.S. Elimination)

The intercompany elimination process is used when an organization is composed of multiple entities that do business with each other. During consolidation, the transactions between the individual entities should be eliminated. In this section, we'll discuss U.S. intercompany eliminations. Later in the chapter, we'll also discuss intercompany elimination using the eliminations and adjustments business rules.

Consider these guidelines for performing U.S. intercompany elimination in SAP BPC:

- To support intercompany elimination, the SAP BPC model should include the Intercompany dimension. The dimension type is I.
- The Intercompany dimension should include a property called Entity, and the value of this property should match the member ID of the Entity dimension.
- The Account dimension should include the ELIMACC property, and the value in this property should match the member ID of the Account dimension.
- The Entity property should include an ELIM property, the value of which can be Y or N (or blank). The elimination process reads all entries for entities that have the value of this property set to N (or blank) and then posts eliminations, if applicable, to entities that have the ELIM property set to Y. These entities are also referred to as elimination entities.

The U.S. elimination process scans all base-level, non-elimination entities, which are entities with the property `ELIM <>Y`. If the application has a Currency dimension, the elimination process restricts its action to all reporting currencies, which are currencies that have the property `REPORTING = Y`.

The Account dimension has a property called `ELIMACC`, which is used in elimination. If the value of this property is blank, the entry associated with this account is not eliminated. If the account has a valid value, and if the system decides the entry needs to be eliminated, the elimination is posted to the plug account (the elimination account which is the account specified in the `ELIMACC` property).

The elimination is posted to the elimination entity below the first common parent. The common parent is derived as follows:

1. The system identifies the two entities for which a common parent must be found. The first entity is the current entity member. The second entity is the entity corresponding to the current intercompany member. This entity of the intercompany member is obtained by reading the content of the `ENTITY` property of the current intercompany member. The system searches in the entity hierarchy for the first member that has both entities as descendants. This is the common parent.
2. Next, the system searches in the immediate descendants of the common parent for a valid elimination entity (an entity that has the property `ELIM = Y`). This is the entity where the system stores the results of the elimination.
3. The default elimination logic does its searches in the organization hierarchy of the Entity dimension. If no common parent is found, no elimination occurs. If no elimination entity is found below the first common parent, the next common parent is searched.

In summary, the elimination entry is posted to the elimination entity associated with the first common parent (in the ENTITY hierarchy) of the intercompany transaction. The account transaction is eliminated. The other side of the transaction will be posted to the plug account (the elimination account) that is defined as the property of the Account dimension.

Let's review the transactions resulting in elimination of intercompany transactions one more time. Consider an intercompany transaction between two entities, E1 and E2, within the organization. Let's say E1 posts an intercompany revenue transaction. At the same time, E2 posts an intercompany COGS/expense transaction

for the same amount. During elimination, the revenue transaction of E1 and the expense transaction of E2 are eliminated. The elimination is posted to the elimination entity associated with the common parent of E1 and E2. When the revenue transaction of E1 is eliminated, the process debits the revenue and credits the plug account associated with the revenue account. When the expense transaction of E2 is eliminated, the process debits the plug account associated with the expense account and credits the expense account. Assuming that the same plug account is used for sales and expense accounts, if the transaction amounts posted by the two entities are the same, the plug account would net to zero as a result of the elimination of sales and expense transactions of E1 and E2, respectively.

You create entries in the business rules table to maintain the source and destination data source to use when calling the logic to eliminate intercompany transactions using U.S. eliminations. This allows you to keep elimination entries posted to a separate data source. Table 9.5 explains the details for setting up the business table for U.S. eliminations.

Field Name	Description
Audit Source	This is the audit dimension member of the data that is read to which elimination needs to be applied.
Audit Destination	This is the destination audit dimension member to which the eliminations entries are posted.
Remark	This is a brief description of the business rule.

Table 9.5 U.S. Eliminations Business Rules

Now let's review an example of how to eliminate intercompany transactions in an SAP BPC model using U.S. eliminations.

1. We are using the model ZRB_CONS_MODEL that was used for demonstrating the use of the carry forward and account transformation rule for this as well. The IntCo dimension is a required dimension in the model to post intercompany transactions.
2. Figure 9.14 shows the members of the Account dimension. The elimination account is a property of the Account dimension; as you can see, the PL020 (Intercompany Sales) and PL130 (Intercompany Cost) members have the elimination plug account ICPLUG associated with them. When eliminations are carried out, the other side of posting of eliminations is posted to the ICPLUG account.

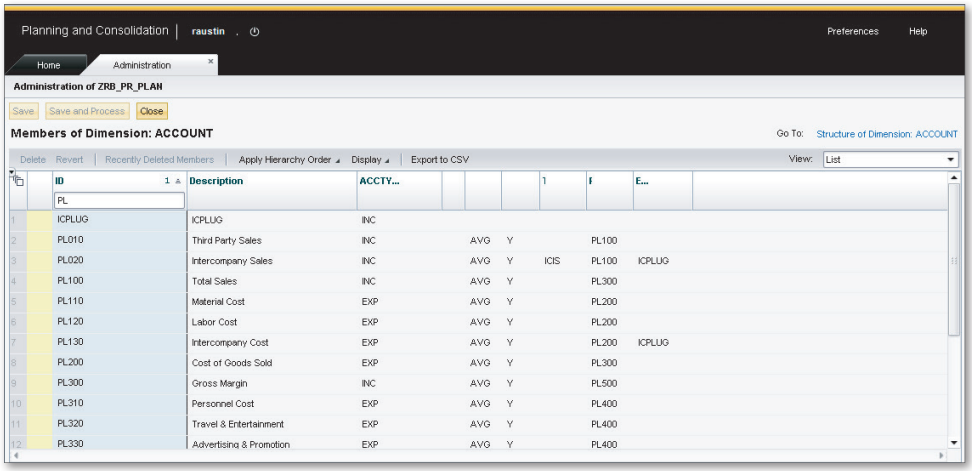


Figure 9.14 Configuring Intercompany Eliminations—Part A

3. Figure 9.15 shows the members of the Entity dimension, of which ELIM is a property. You can see that the entity E_NA (elimination entity for North America) has the `ELIM` property set to `Y`; this entity is an elimination entity. Consider the three entities under node NAMERICA (North America): CA (Canada), US (United States), and E_NA (elimination entity). The entity E_NA is identified as the elimination entity; for example, eliminations for intercompany transactions between entities CA and US will be posted to entity E_NA.

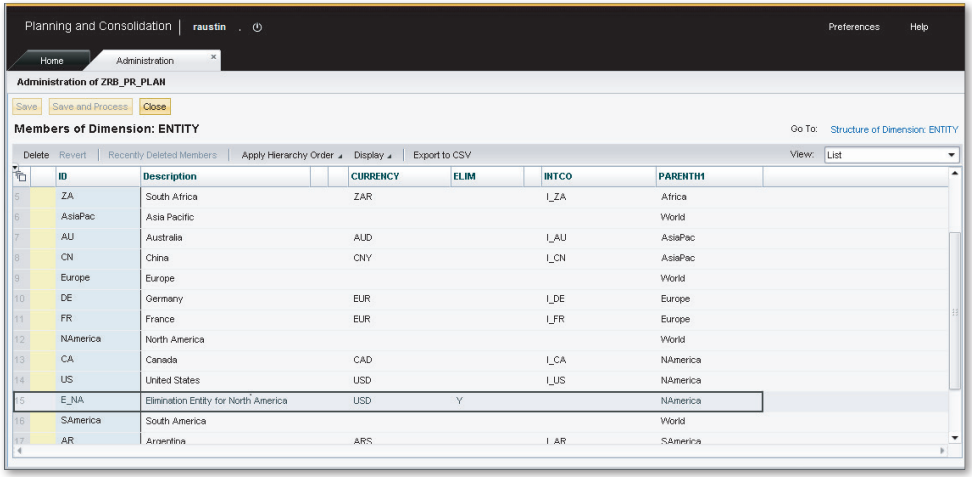


Figure 9.15 Configuring Intercompany Eliminations—Part B

4. Figure 9.16 displays the intercompany transactions for May 2012. Entity CA (Canada) has recorded a sales transaction in account PL020 (Intercompany Sales) with trading partner I_US (USA). The transaction was entered under the INPUT DataSource. At the same time, entity USA recorded a cost transaction in account PL130 (intercompany cost transaction with trading partner I_CA (Canada).

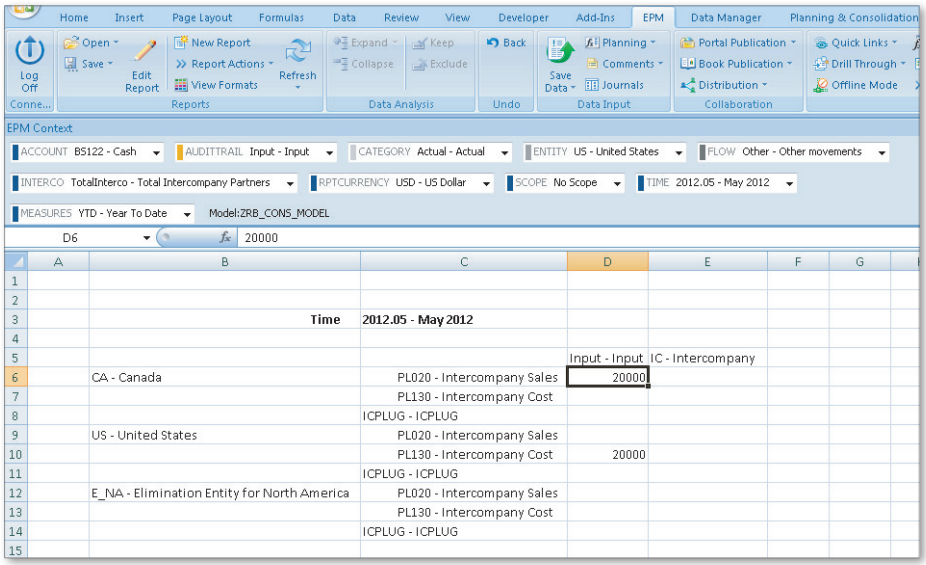


Figure 9.16 Configuring Intercompany Eliminations—Part C

5. Set business rules for U.S. eliminations so that the elimination entries for records from the audit source Input are created under the member ICELIM (InterCompany Eliminations), as shown in Figure 9.17.

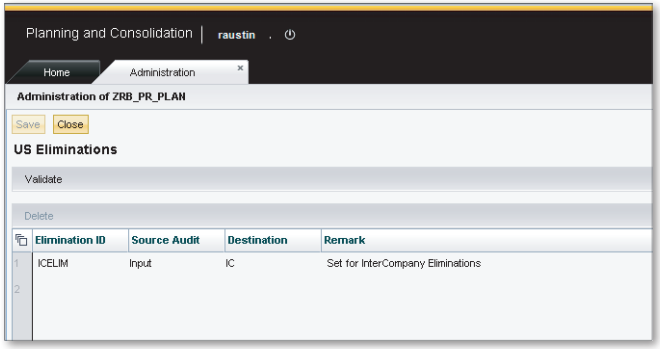


Figure 9.17 Configuring Intercompany Eliminations—Part D

6. Create the standard logic script ICELIM.LGF (if one does not already exist) to execute the U.S. intercompany eliminations, as shown in Figure 9.18.

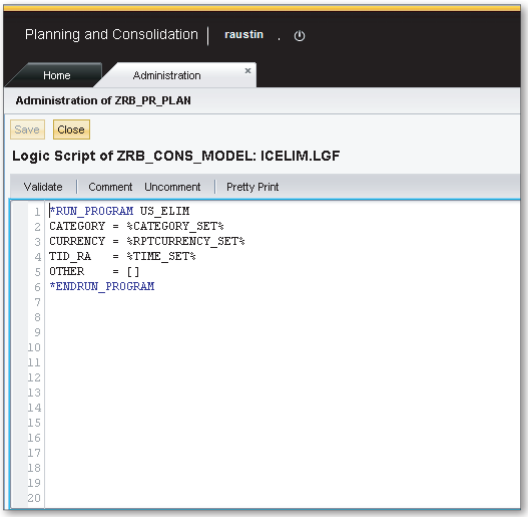


Figure 9.18 Configuring Intercompany Eliminations—Part E

7. Run the data package to execute the INTER COMPANY ELIMINATION script, as shown in Figure 9.19. The process chain associated with the elimination process is `/CPMB/IC_ELIMINATION`.

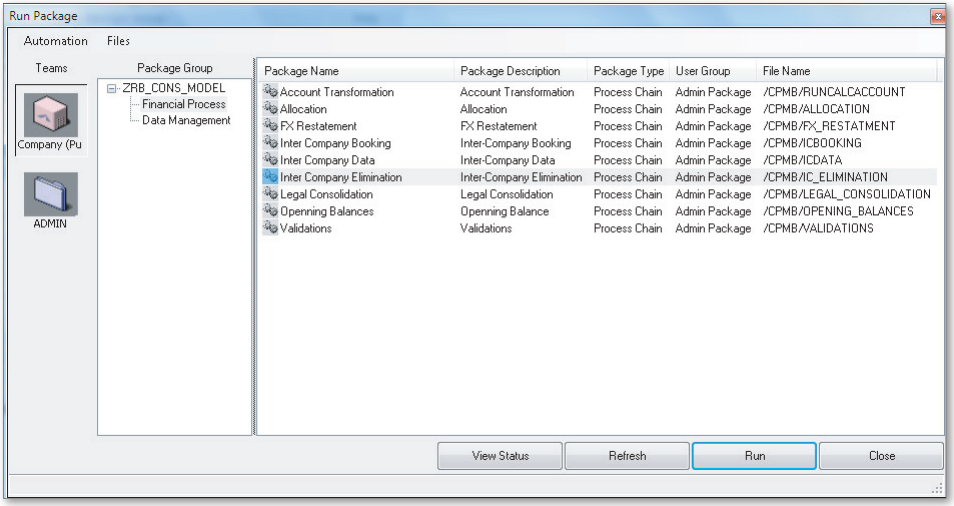


Figure 9.19 Configuring Intercompany Eliminations—Part F

8. Run U.S. eliminations for the selections shown in Figure 9.20. In our example, we chose May 2012.

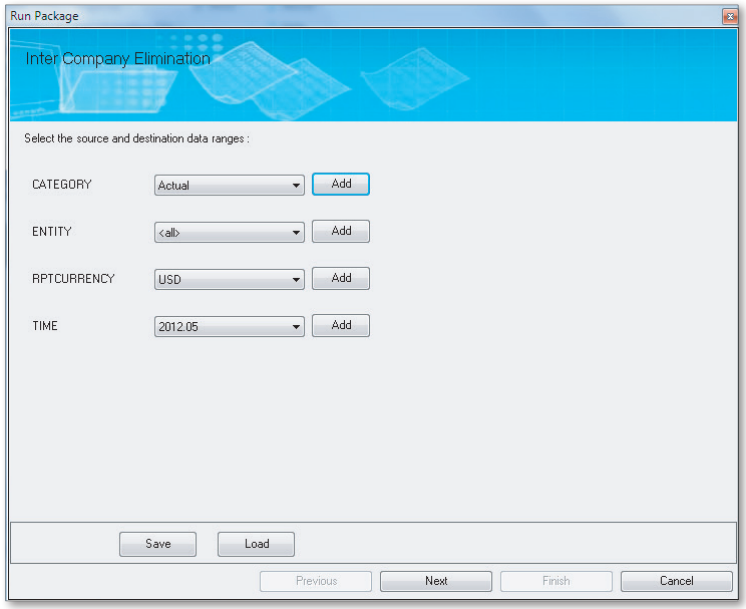


Figure 9.20 Configuring Intercompany Eliminations—Part G

9. After executing the package, you'll notice that the intercompany sales transaction for CA (Canada) and the intercompany cost transaction for US (the United States) are eliminated (Figure 9.21). The elimination entries are posted to the elimination entity E_NA. The net effect on the plug account is equal to zero.

We've discussed the use of business rules to perform intercompany U.S. elimination. In the next section, we'll explore the use of the controls rule introduced in SAP BPC 10.0 for NetWeaver and how it supports the data consistency maintenance in SAP BPC.

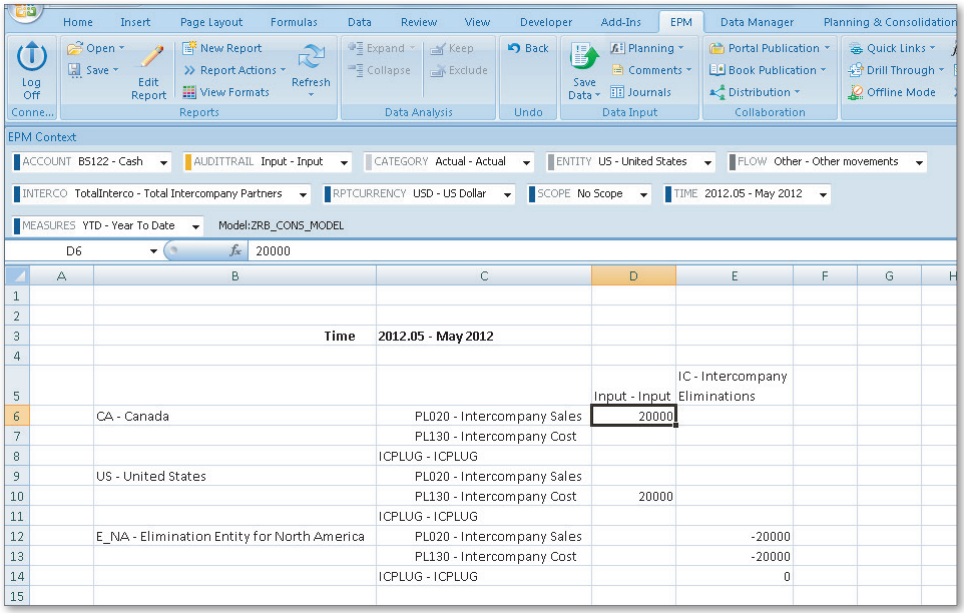


Figure 9.21 Configuring Intercompany Eliminations—Part H

9.2.5 Controls

Control is a new rule type introduced in SAP BPC 10.0 for NetWeaver. It replaces the validations functionality that was available under business rules in the SAP BPC 7.x versions. Controls provide enhanced features to ensure the consistency and accuracy of data in a model and are superior to the validations functionality that existed in the prior versions of SAP BPC.

Controls are enabled at the model level to ensure that the data inside a model is reliable. Controls are used to verify that the data in a model is accurate and suitable for reporting. For example, a control can be defined to ensure that the total of the assets matches the total of liabilities/owner's equity data for a reporting period. Controls can be used in both planning and consolidation models.

The following features are available when defining controls:

- The setting to use controls is enabled at the model level. Controls can be enabled only when work status is set for a model. Work status is set inside the Work Status view under the Features domain in the Administration workspace. We discussed work status in detail in Chapter 7.

- There are two security tasks that determine the access available when configuring controls.
  - View control definition: This task provides the access rights to a user to view controls.
  - Edit control definition: This task provides the access rights to a user to edit controls.
- Controls are defined in the ADMINISTRATION workspace of the Web Client. Control is enabled for a model and is defined from the CONTROLS view of the RULES domain in the Web Client. Every control defined has a level associated with it. The following four levels are available when defining the control:
  - 1: Basic
  - 2: Standard
  - 3: Advanced
  - 4: Comprehensive
- Two properties, `CONTROL_LEVEL` and `CTRL_CURRENCY_NOT_LC` in the Entity dimension, determine the applicability of a control for an entity and the currency used in the execution of a control for that entity.
  - The property `CONTROL_LEVEL` set for an entity in the Entity dimension defines the controls that are applicable to it. A value between `1` and `4` can be set as the value for an entity. If no value is specified for an entity, all of the controls of type comprehensive (`4`) apply to that entity.
  - The value in the `CTRL_CURRENCY_NOT_LC` property for an entity determines if the data in the local currency or reporting currency is used when executing the controls for an entity. If a value is set to `N`, controls are executed for the data in the local currency of the entity. If a value is set to `Y`, controls are executed for the data in the reporting currency of the entity. If no value is specified, the controls are executed for the data in the local currency.
- Controls can be defined for a model only after it is enabled. This is done in the CONTROLS view under the RULES domain in the ADMINISTRATION workspace of the Web Client.
- A control definition includes a header and a detailed section. In the header section, the technical name of the control, level, type, equation, threshold, and breakdown dimensions are specified.

- Control level: Recall that the level for a control can be set as Basic (1), Standard (2), Advanced (3), or Comprehensive (4). In the Entity dimension, you can specify a value in the CONTROL_LEVEL property. The value specified for this property determines the controls that are applicable for an entity.
- Type: A control can be defined as a blocking or a warning control. If a control is defined as a blocking control, work status cannot be set if the control fails.
- Equation type: The equation type indicates the comparison to use when executing a control. Valid values are equal, not equal to, less than, and greater than operators.
- Threshold: A threshold can be set when defining a control. For example, a threshold can be set to accept a difference of 1% when executing controls. In this example, if the control fails and when the difference is less than 1%, the system considers that the control has passed.

- A maximum of two dimensions can be specified as breakdown dimensions. It is not mandatory to specify a breakdown dimension. When a breakdown dimension is specified, the system automatically selects the children of the specified member associated with the breakdown dimension and executes the control for each of those dimension members. For example, let's say a dimension is selected as a breakdown dimension in a control. When the dimension member that is associated to the breakdown dimension is a parent, the control is executed for all of the children below that dimension member.
- The detailed section of a control is divided into two parts:
  - The top part of the detailed section consists of the dimension member selections that you want to use for the control—the source selections. The category and the time period to use are specified when assigning the control set.
  - The bottom part of the detailed section consists of the dimension member selections you want to use for the comparison—the target selections. In addition, a category can be optionally specified to use in the target selection for comparing the data. If a category is not specified, the category specified in the control assignment is used. You can also specify the year and period offsets to use in the target selections. The offsets to the year and periods are with respect to the period selected in the control assignment. More than one row of data can be created in the top and bottom parts of the detail section. The results of the top and the bottom part of the selections are used when

executing the control and are compared to determine whether a control has passed or failed.

- A document can be created and linked to a control to provide the user with information on why the control was created and how it is used inside a model.

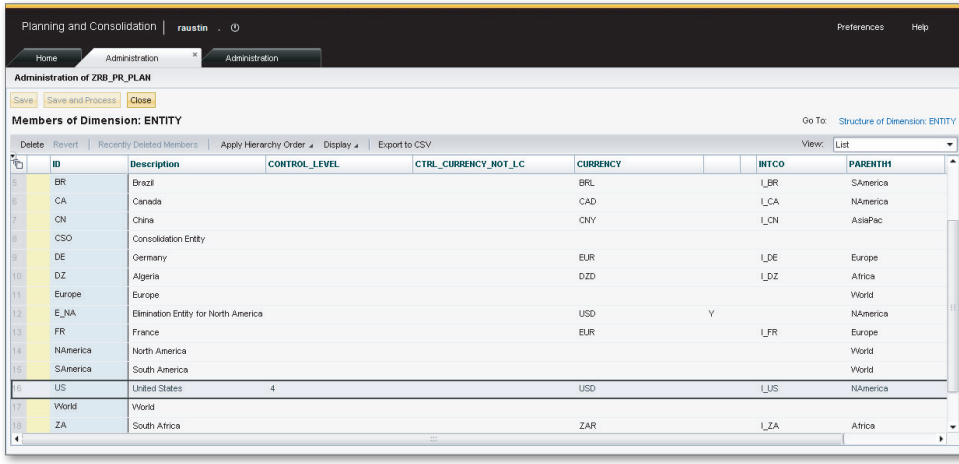
A control set can be defined that allows more than one control to be selected and grouped in a control set. Using this feature, all of the related controls are grouped into a single control set and executed as one.

Finally, a control set is assigned to a category and a time dimension member. A control can be executed only for the category and time period assigned to the control set.

You can view and execute controls from the HOME workspace in the Web Client. The CONTROLS MONITOR view under the CONSOLIDATION CENTRAL component is available to run. Select the model in the upper-right corner of the CONTROLS MONITOR to display the controls assigned to a model. You can use this view to execute the control and display the results.

So let's review an example of how to define and run controls in an SAP BPC model.

1. Set the `CONTROL_LEVEL` and `CTRL_CURRENCY_NOT_LC` property values for members in the ENTITY dimension (Figure 9.22). This is optional and is required only if you want to run specific controls for an entity or if you want to use the reporting currency as the currency to use when executing a control for an entity.



	ID	Description	CONTROL_LEVEL	CTRL_CURRENCY_NOT_LC	CURRENCY		INTCO	PARENTH1
5	BR	Brazil			BRL		I_BR	SAmerica
6	CA	Canada			CAD		I_CA	NAmerica
7	CN	China			CNY		I_CN	AsiaPac
8	CSO	Consolidation Entity						
9	DE	Germany			EUR		I_DE	Europe
10	DZ	Algeria			DZD		I_DZ	Africa
11	Europe	Europe						World
12	E_NA	Elimination Entity for North America			USD	Y		NAmerica
13	FR	France			EUR		I_FR	Europe
14	NAmerica	North America						World
15	SAmerica	South America						World
16	US	United States	4		USD		I_US	NAmerica
17	World	World						
18	ZA	South Africa			ZAR		I_ZA	Africa

Figure 9.22 Configuring and Running Controls—Part A

2. Enable the work status settings for a model (Figure 9.23). The work status setting is enabled for a model from the WORK STATUS view under the features domain in the ADMINISTRATION workspace. You must set this before controls can be created and maintained for a model.

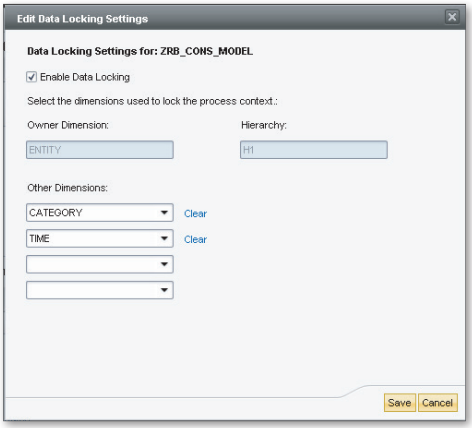


Figure 9.23 Configuring and Running Controls—Part B

3. Select the CONTROLS VIEW under the RULES domain. Select the model, and click on the menu button ENABLE CONTROLS to enable controls, and check the ENABLE CONTROLS FOR ... checkbox (Figure 9.24, ❶ and ❷). After the controls are enabled for a model, click on the menu option OPEN to create a control.

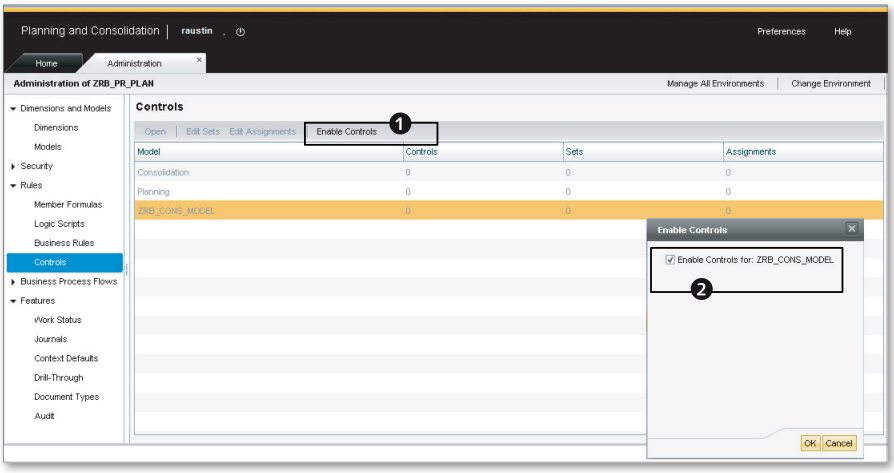


Figure 9.24 Configuring and Running Controls—Part C

4. You will see the control definition interface with the list of controls defined for the model. To create a new control, click on the NEW menu option (Figure 9.25).

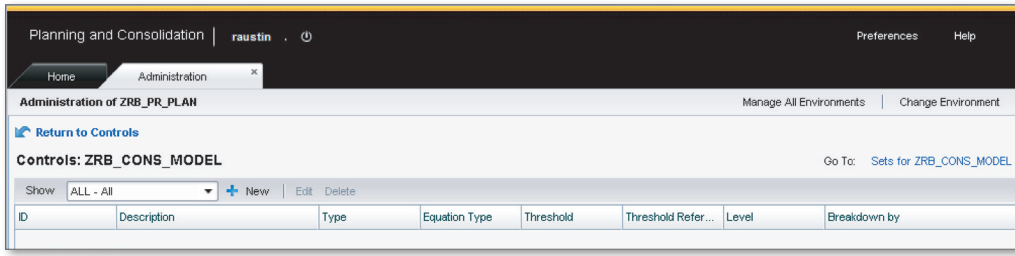


Figure 9.25 Configuring and Running Controls —Part D

5. Define a control as shown in Figure 9.26. Every control has a unique ID associated with it. Notice that the AUDITTRAIL dimension is selected as a BREAKDOWN DIMENSION. The dimension member selected for the breakdown in the source is AUDITTOTAL. In the case of the breakdown dimension, the dimension member that is selected in the source has to be selected in the destination as well. The control is designed to ensure that the total of all of the assets (BS100) equals the liabilities and equity (BS200). The tolerance threshold is set to 5%, which means that the control will tolerate a difference of up to 5% when comparing the source and destination selections. Please note that in real-life scenarios, balance in assets should exactly match the sum of liabilities and equity and that the threshold of this check should be 0%.

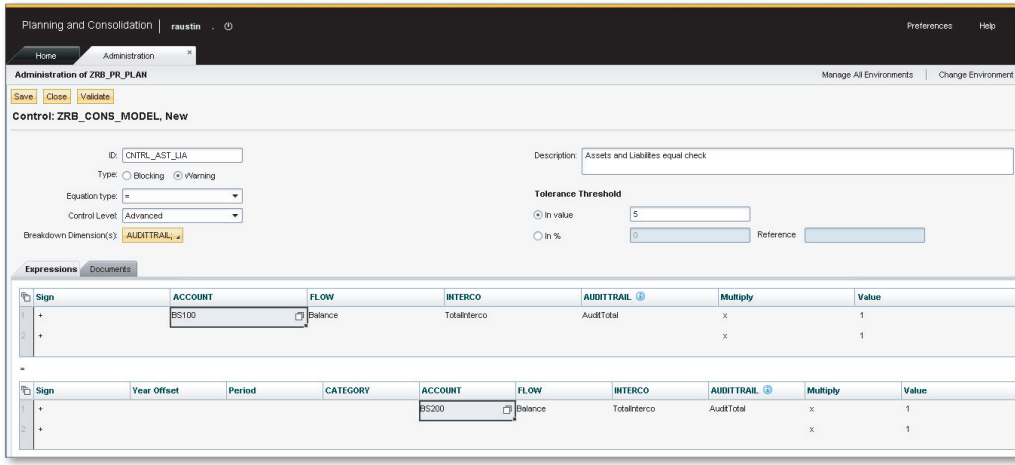


Figure 9.26 Configuring and Running Controls—Part E

6. After setting the definition for the control, click on the SAVE menu button to save the control. Click on the CLOSE menu button to go back to the controls definition interface.
7. Next, we'll create a control set and assign to it the control that we created. Remember that the control set allows a group of controls to be executed together in a single step. Select the control you created in the earlier steps, and click on the link GO TO: SETS FOR ZRB_CONS_MODEL (Figure 9.27). In the interface for maintaining control sets, click the menu option NEW to create a new control set.

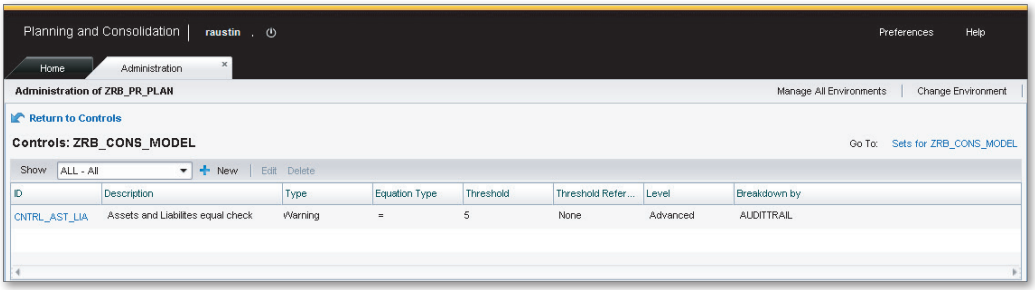


Figure 9.27 Configuring and Running Controls—Part F

8. In the interface for creating a control set, enter an ID for the control set (Figure 9.28). Click on the ADD/REMOVE menu button to select the controls for the new control set. Select the controls you want to associate to the control set (Figure 9.29), and then save the control set and return to the control set interface.

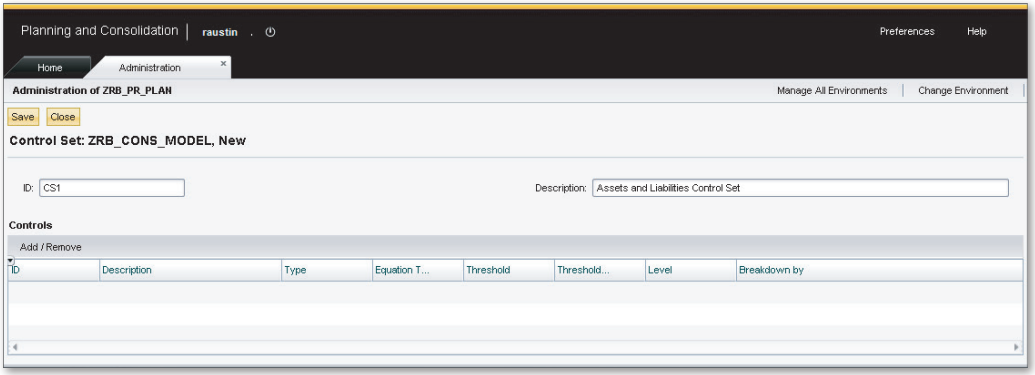


Figure 9.28 Configuring and Running Controls—Part G

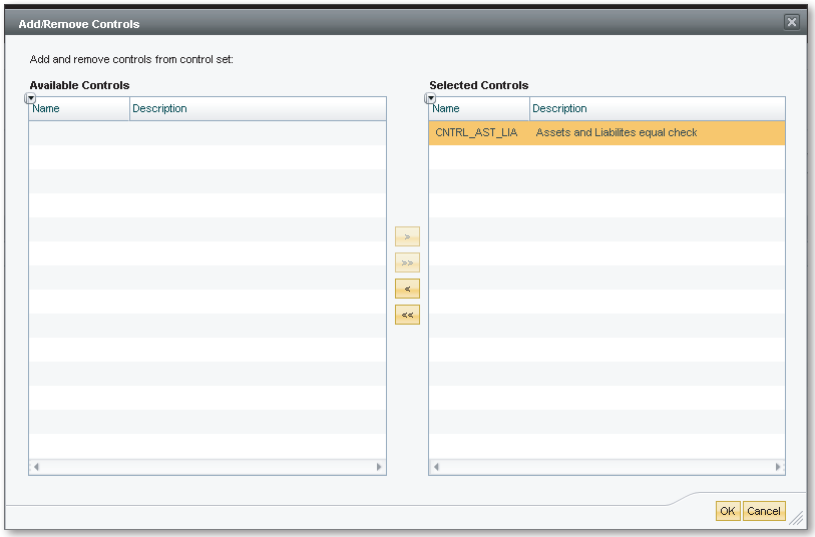


Figure 9.29 Configuring and Running Controls—Part H

9. In the control set interface, click on a control set, and click on the Go To: Assignments for ZRB_CONS_MODEL link (Figure 9.30).

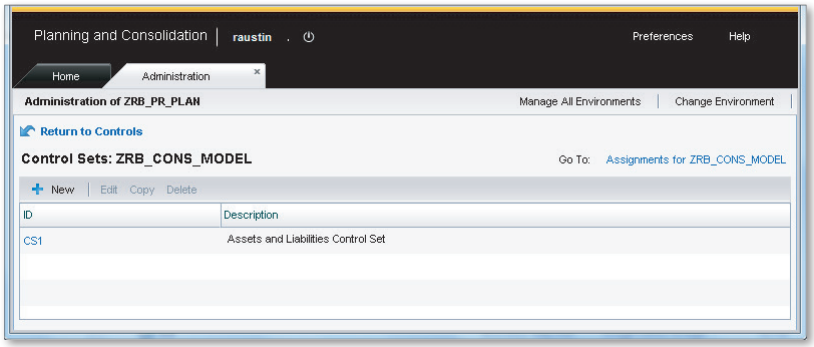


Figure 9.30 Configuring and Running Controls—Part I

10. In the assignments interface, click on the Edit menu button, and select the TIME and CATEGORY dimension members. For a time and category intersection, select a category set (Figure 9.31). You have assigned a control set to a time and category selection, so you're now ready to execute the control.

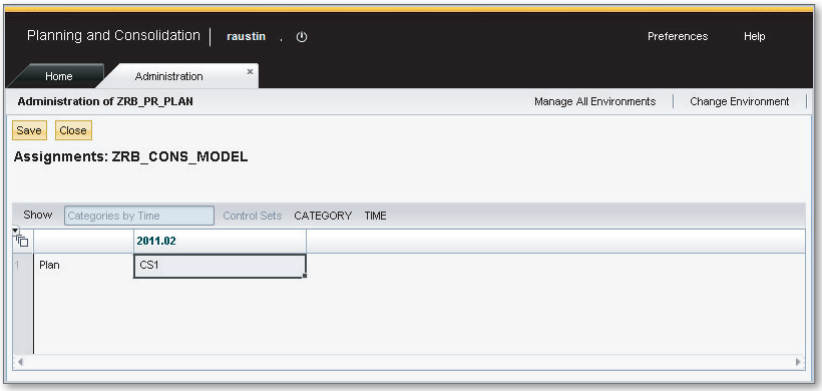


Figure 9.31 Configuring and Running Controls—Part J

11. In the HOME workspace and in the CONSOLIDATION CENTRAL domain, select the CONTROLS MONITOR view. Select the model for which you assigned the controls. In this view, select the same CATEGORY and TIME dimension member you used when assigning a control set. Select the Scope (or Group) dimension member (Figure 9.32). You will see the control set displayed with the hierarchy selection for the time and category as defined in the ownership model. The hierarchy selections should be defined in the ownership model.

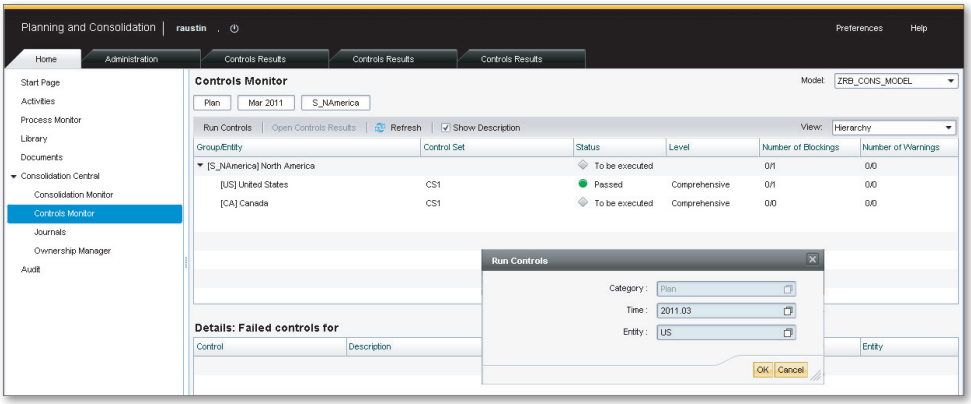


Figure 9.32 Configuring and Running Controls—Part K

In the report displayed in Figure 9.33, you see the asset and liability/equity data for the category and time data for which we will be executing the control we created earlier.

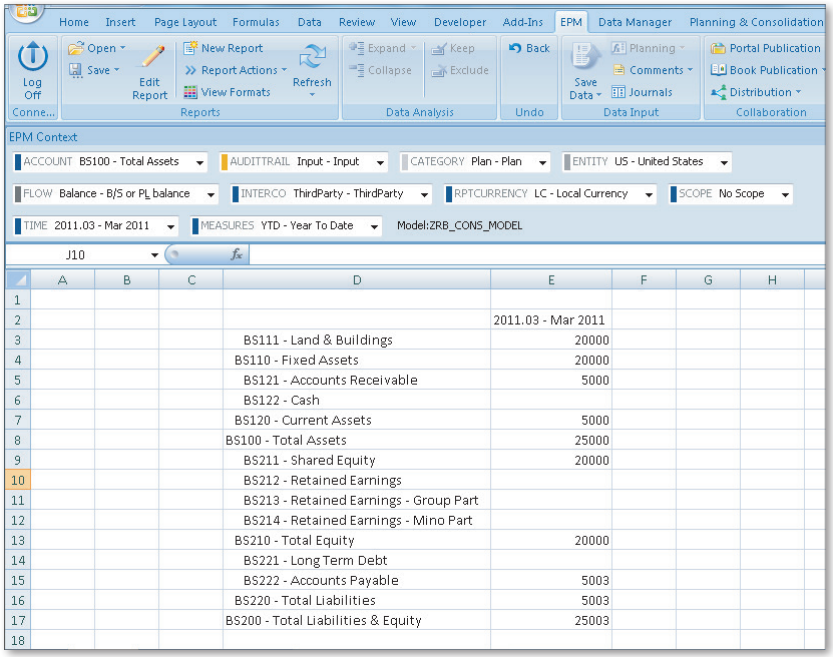


Figure 9.33 Configuring and Running Controls—Part L

12. Click at the node level or a base member, and click on the menu option Run Controls to run the control (refer back to Figure 9.32).
13. You can view the status of the control in the Status column (Figure 9.34). You can click on the entity member and on Open Controls Results to view the detailed log of the control executed (refer back to Figure 9.32).

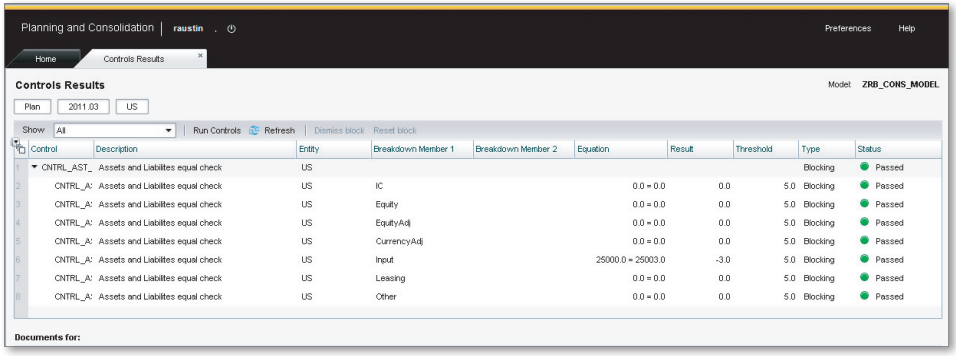


Figure 9.34 Configuring and Running Controls—Part M

In the next section, we'll discuss consolidation concepts and give an overview of investment consolidation.

9.3 Consolidation of Investments

A corporation may have a number of legal subsidiaries that may or may not be fully owned. Consolidating investments deals with eliminating ownership and booking minority interest of these investments. Three methods are commonly used when consolidating investments:

- Purchase method
- Equity method
- Proportional method

The purchase method is generally used when the percentage of ownership is greater than 50%. The equity method is generally used when the percentage of ownership is less than 50%. The proportional method is similar to the purchase method (generally used when the percentage of ownership is greater than 50%) and is generally used in Europe. Let's explore these three methods more closely, starting with the purchase method.

9.3.1 Purchase Method

At the time of purchase, the consolidation of investment process reflects the elimination of ownership that is common between the parent and subsidiary, goodwill for the premium paid for the purchase, and minority stakeholders' claim to the asset. After this first consolidation, assuming that the ownership percentage does not change, subsequent consolidations post the parent and minority share of the subsidiary's earnings.

Let's say that Company A acquires 75% of subsidiary Company B for $50,000. Figure 9.35 displays the books of Company A, Company B, and the total, elimination, and consolidated group balances.

At the time of acquisition, the following transactions are performed:

- Consolidate balances of parent and subsidiary.

- Eliminate ownership that is common between the parent and subsidiary—eliminate the investment account of the parent Company A ($50,000) that pertains to the subsidiary and common stock of subsidiary Company B ($40,000).
- Post any excess value paid that is above the book value of the acquired subsidiary to goodwill—the book value of the common stock of Company B is $40,000. Company A acquired 75% of Company B. So Company A acquired $30,000 of the book value of the stock. The excess amount of $20,000 ($50,000 – $30,000) paid by Company A is booked to goodwill.
- Post minority interest. The amount of minority interest appearing in the balance sheet is calculated by multiplying common shareholder equity of the subsidiary by the percentage of the minority interest—the minority interest of Company B is 25% (100% – 75%) of $40,000 which amounts to $10,000.

Purchase Method

	Company A	Company B	Total	Elimination	Group
Goodwill				$20,000.00	$20,000.00
Cash	$100,000.00	$40,000.00	$140,000.00		$140,000.00
Investments	$50,000.00		$50,000.00	-$50,000.00	$0.00
Total	$150,000.00	$40,000.00	$190,000.00	-$30,000.00	$160,000.00
					$0.00
Common Stock	-$130,000.00	-$40,000.00	-$170,000.00	$40,000.00	-$130,000.00
Retained Earnings	-$20,000.00		-$20,000.00		-$20,000.00
Minority Interest				-$10,000.00	-$10,000.00
Total	-$150,000.00	-$40,000.00	-$190,000.00	$30,000.00	-$160,000.00

Company A owns 75% of Company B for $50,000

Figure 9.35 Purchase Method (Example)

9.3.2 Equity Method

In the equity method, reported financial data of the equity unit is not taken into consideration in the consolidated financial report. The financial data for the equity unit is not entered into the consolidation system. Only the changes to the owner's equity are taken into consideration. This affects the investment value and goodwill stated in the consolidated balance sheet.

Let's say that Company C buys 25% of Company D for $15,000. Figure 9.36 displays the books of Company C, Company D, and the elimination and consolidated group balances. The book value of Company D is $40,000, of which Company C owns $10,000 (25% of $40,000). The investment value of the group is adjusted so that the net value is $10,000. This is done by eliminating $5,000 from the investments

value of $15,000. The excess amount paid of $5,000 ($15,000 – $10,000) is posted to goodwill.

Equity Method				
	Company C	Company D	Elimination	Group
Goodwill			$5,000.00	$5,000.00
Cash	$100,000.00	$40,000.00	-$40,000.00	$100,000.00
Investments	$15,000.00		-$5,000.00	$10,000.00
Total	$115,000.00	$40,000.00	-$40,000.00	$115,000.00
Common Stock	-$30,000.00	-$40,000.00	$40,000.00	-$30,000.00
Retained Earnings	-$85,000.00			-$85,000.00
Minority Interest				$0.00
Total	-$115,000.00	-$40,000.00	$40,000.00	-$115,000.00
Company C owns 25% of Company D for $15,000				

Figure 9.36 Equity Method (Example)

9.3.3 Proportional Method

The proportional method of consolidation collects the units of the balance sheet and income statement into the consolidated statements based on the investor unit's proportion of ownership in the investee's unit. In other words, the percentage that is not owned is eliminated.

Let's say that Company E acquires 75% of subsidiary Company F for $50,000. Figure 9.37 displays the books of Company E, Company F, and the total, elimination, and consolidated group balances.

At the time of acquisition, the following transactions are performed:

- Consolidate balances of parent and subsidiary.
- Eliminate ownership that is common between the parent and subsidiary—eliminate the investment account of the parent Company E ($50,000) that pertains to the subsidiary and common stock of subsidiary Company F ($40,000).
- Post any excess value paid that is above the book value of the acquired subsidiary to goodwill—the book value of the common stock of Company F is $40,000. Company E acquired 75% of Company F. So Company E acquired $30,000 of the book value of the stock. The excess amount of $20,000 ($50,000 – $30,000) paid by Company E is booked to goodwill.
- Eliminate the proportion of balances that are not owned—Company F has $40,000 in a cash account. Company E owns 75% of Company F. The remaining 25%

(100% – 75%) of cash should be eliminated. The amount eliminated from the cash account is $10,000.

The main difference between the purchase method and the proportional method is that the purchase method posts the non-owned portion to a minority share account, whereas the proportional method eliminates the proportion of balances that are not owned.

Proportional Method					
	Company E	Company F	Total	Elimination	Group
Goodwill				$20,000.00	$20,000.00
Cash	$100,000.00	$40,000.00	$140,000.00	-$10,000.00	$130,000.00
Investments	$50,000.00		$50,000.00	-$50,000.00	$0.00
Total	$150,000.00	$40,000.00	$190,000.00	-$40,000.00	$150,000.00
					$0.00
Common Stock	-$130,000.00	-$40,000.00	-$170,000.00	$40,000.00	-$130,000.00
Retained Earnings	-$20,000.00		-$20,000.00		-$20,000.00
Minority Interest					$0.00
Total	-$150,000.00	-$40,000.00	-$190,000.00	$40,000.00	-$150,000.00
Company E owns 75% of Company F for $50,000					

Figure 9.37 Proportional Method (Example)

In the next section, we'll detail the steps involved in consolidating investments.

9.3.4 Consolidation of Investments

The following high-level steps accomplish consolidation of investments:

1. Set up dimension members and properties.
2. Create rate, legal, and ownership models.
3. Set relevant model parameters.
4. Configure the consolidation business rules at the environment level and consolidation models.
5. Input methods, percentage control, consolidation, and ownership in the ownership model.
6. Configure the eliminations and adjustments business rule in the legal model.
7. Load exchange rates and data for the legal consolidation model, and then run consolidation from the consolidation monitor and analyze the results.

Let's discuss each of these steps in more detail.

Set Up Dimension Members and Properties

The consolidation of investments requires that the following dimension properties be included:

- `IS_CONSOL` **of the AUDIT dimension**
  If the value of this property is set to `Y`, the data source participates in the consolidation run. If this property is set to `N`, the data source does not participate in the consolidation run.
- `DATASRC_TYPE` **of the AUDIT dimension**
  Data sources that have this property set to `A` (automatic) can be used as destination data sources in business rules.
- `IS_CONVERTED` **of the AUDIT dimension**
  If the value of this property is set to `Y`, the data source participates in the currency translation. If this property is set to `N`, the data source does not participate in the currency translation.
- `Group` **type dimension**
  The dimension of type `group` represents the relationship of entities for a given consolidation result. In our examples, the name of this dimension is called SCOPE.

Setup of Models

Create a legal model and assign a rate model and ownership model to it. The rate model contains the Category, Currency (Input_Currency), Account (R_ACCT), Entity (R_ENTITY), and Time dimensions. The rate model is used to store currency exchange rates. Ownership is a non-reporting model that is used to store ownership data. The ownership model contains the Scope, Category, Time, Entity, Intco, and Ownership Account (O_ACCT) dimensions. The Ownership Account dimension contains the percentages, methods, and parent positions.

Set Relevant Model Parameters

Set the parameters for the ownership model as described in Table 9.6. You can define these parameters in the Models view under the Dimensions and Models domain in the Administration workspace.

Model Parameter	Description
Non-interco member in Ownership	Specify a third-party member in the Intercompany dimension to use in the ownership model. All transactions to third parties (external) are posted to this intercompany member (e.g., I_NONE).
Parent/child property used for hierarchy of groups	This is the property name in the Group dimension used for the hierarchy of groups.

Table 9.6 Model Parameters Set for the Ownership Model

Ownership Model

The Ownership view under the Consolidation Monitor domain in the Web Client is used for entering the ownership information. The ownership model stores the ownership and consolidation percentages of each entity with respect to a group. The ownership model contains the dimensions listed in Table 9.7. Use the same Category, Time, Group, and Entity dimension names in the consolidation and ownership models.

Dimensions	Description
O_ACCT	Account
CATEGORY	Category
ENTITY	Entity
SCOPE	Group
INTCO	Intercompany
TIME	Time

Table 9.7 Dimensions in the Ownership Model

The O_ACCT dimension is used to specify the following:

- **Ownership percentage (POWN)**
  This represents the percent owned and is also referred to as the financial interest rate.

- **Percentage consolidation (PCON)**
  This represents the percentage of company that should be considered for ownership and is also referred to as the consolidation rate.
- **Percentage control (PCTRL)**
  This represents the percentage control.
- **Consolidation method (METHOD)**
  This represents the method used in consolidation. For example, 100 = full (or purchase method), 20 = equity method, 50 = proportionate method, and 101 = holding company.

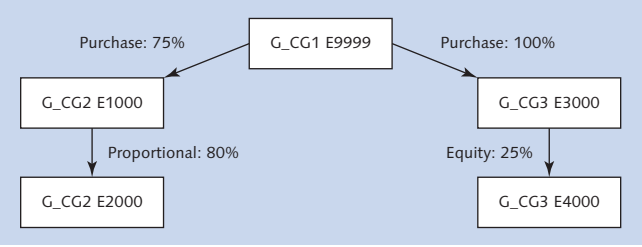


Figure 9.38 Ownership Structure (Example)

Consider the ownership structure of an organization as depicted in Figure 9.38. There are three consolidation groups—G_CG1, G_CG2, and G_CG3. Figure 9.39 shows the dimension members of the SCOPE dimension (dimension of type `group`) for the three groups. Notice that G_CG2 and G_CG3 have G_CG1 as parent group. The ownership information in the ownership model is maintained against the consolidation groups. You can also specify an entity property to each consolidation group where consolidated results for each group can be stored. During consolidation, if the `STORE_ENTITY` property of the group is set to `Y` and the entity property contains a valid entity, the system will post group-level elimination values to this entity.

The holding companies for consolidation groups G_CG1, G_CG2, and G_CG3 are E9999, E1000, and E3000, respectively.

SCOPE Dimension Members - GROUPS

ID	EVDESCRIPTION	CURRENCY_TYPE	ENTITY	GROUP_CURRENCY	PARENT_GROUP	STORE_ENTITY
G_H1	Investments	G		USD	G_H1	
G_CG1	Group 1	G		USD	G_H1	
G_CG2	Group 2	G		USD	G_CG1	
G_CG3	Group 3	G		USD	G_CG1	
NON_GROUP	Group Input	N				

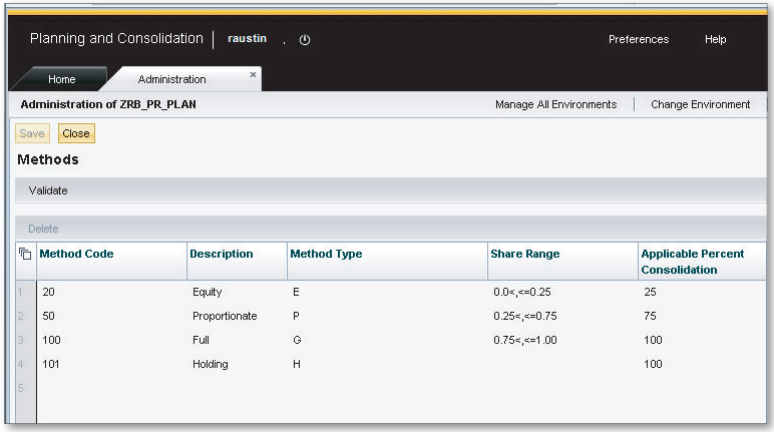
Figure 9.39 Dimension Members of the SCOPE dimension (Dimension of Type "Group")

Configure the Consolidation Business Rules at the Environment Level

Shortly we'll discuss how to enter ownership information, but first let's review the global definition for consolidation methods and method-based multiplier that is available at the environment level. The legal consolidation model uses the following definitions at the environment level to execute consolidation tasks:

- **Consolidation methods**
  These refer to different types of consolidation methods that are available for consolidation. Recall that we discussed 100 = full method, 20 = equity method, 50 = proportionate method, and 101 = holding company. These entries are configured in this table. The METHODS rule type link is accessed from the BUSINESS RULES view under the RULES domain in the ADMINISTRATION workspace.

  Figure 9.40 shows a list of methods. The share ranges defined here are used when calculating the system-generated ownership values and percentages.



	Method Code	Description	Method Type	Share Range	Applicable Percent Consolidation
1	20	Equity	E	0.0<,<=0.25	25
2	50	Proportionate	P	0.25<,<=0.75	75
3	100	Full	G	0.75<,<=1.00	100
4	101	Holding	H		100
5					

Figure 9.40 Global Methods

- **Method-based multiplier**
  The METHOD-BASED MULTIPLIER interface provides the options to define the rules that are referenced in the MULTIPLIER field in the eliminations and adjustments rule type for a consolidation model. The METHOD-BASED MULTIPLIERS RULE TYPE link is accessed from the BUSINESS RULES view under the RULES domain in the ADMINISTRATION workspace. Figure 9.41 shows an example of a multiplier rule to split equity into group and minority based on ownership percentages.

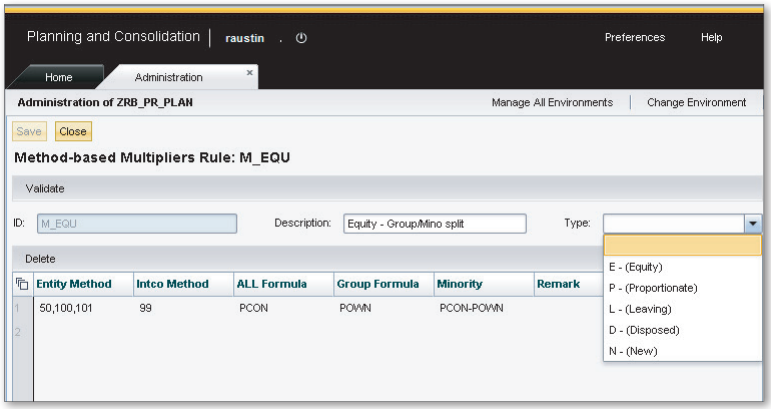


Figure 9.41 Method-Based Multiplier Rule

Method-Based Multiplier Rules

You can define a rule to only apply to a specific method by selecting the type from the dropdown list when creating the rule.

- **Method-based multiplier rule formula**

  Formulas are associated with a method-based multiplier rule that determines how the destination account amounts will be calculated.

  Figure 9.42 displays a method-based multiplier rule on how to split the ownership data based on various methods. This table contains the Entity Method, Intco Method, ALL Formula, Group Formula, Minority Formula, and Remark fields. You maintain this rule for each entity method. A value of 99 in the INTCO Method field means the formula is valid for any intercompany member. For the purchase method (consolidation method = 100), you set "ALL" Formula to 1 to mean that you consider 100% of the amount in the source account. In the Group Formula field, you specify POWN to mean that only the percentage owned is applied to the destination group account. In the Minority Formula field, you specify 1-POWN to mean the percentage that is not owned is applied to the minority shares account.

  Note that the Method-based Multiplier Rule formula table contains the formula for the calculation of destination accounts. The destination accounts are defined in the eliminations and adjustments business rules table that we will discuss in the next section.

Entity Method	INTCO Method	"ALL" formula	Group Formula	Minority Formula	Remark
100	99	1	POWN	1-POWN	
20	99	POWN	POWN		
50	99	POWN	POWN		

Figure 9.42 Method-Based Multiplier Rule for Different Methods

Input Methods, Position, and Percentages in the Ownership Model

The Ownership Manager is used to maintain the ownership method, percent control, percent consolidation, and percent ownership at the group level. The Ownership Manager was called the Dynamic Hierarchy Editor in prior versions of SAP BPC. The ownership hierarchy is identified with a context that includes the Category, Time, and Scope (group) dimension members. This allows multiple organizational structures to be created for different contexts to support changes to ownership. Ownership hierarchies can also be copied into a different context.

The ownership data for the group can be entered directly in the Ownership Manager. The Ownership Manager Editor view can be accessed from the Consolidation Central domain in the Home workspace. The ownership structures created using the Ownership Manager are identified with the Category, Time and Scope dimensions. To edit ownership, select the context (category, time, and group members). The data for the selection is displayed. Then click on the Edit menu option to enter ownership data.

In the Ownership Manager Editor, you also have an option to allow the system to calculate/generate the ownership values. The values calculated by the system are displayed in the Generated column. The calculated values are stored in the system with the suffix _SYS (as in POWN_SYS, PCTRL_SYS, METHOD_SYS, and PCON_SYS). To allow the system to calculate ownership in the Ownership Manager Editor, follow these steps:

1. Enter cross ownership data for percentage control (PCTRL) and percentage ownership (POWN) using the input form, as shown in Figure 9.43. The group is set to NON_GROUP. For example, in the organization structure scenario illustrated earlier in Figure 9.38, because E9999 owns 75% of E1000, the entity value is set to E9999, the intercompany value is set to I_E1000 (this is the intercompany dimension member associated with entity E1000), and the ownership is set to 0.75.

OWNERSHIP

		NON_GROUP	NON_GROUP	NON_GROUP	NON_GROUP	NON_GROUP
		I_E9999	I_E1000	I_E2000	I_E3000	I_E4000
POWN	E9999		0.75		1	
	E1000			0.8		
	E2000					
	E3000					0.25
	E4000					
PCTRL	E9999		1		1	
	E1000			0.8		
	E2000					
	E3000					0.25
	E4000					

POWN	Percentage Ownership
PCTRL	Percentage Control

Figure 9.43 Ownership Model—Enter POWN (Percentage Ownership) and PCTRL (Percentage Control) Data in Input Form That Will Be Used for Calculation of Ownership

2. Click on the CALCULATE menu option in the OWNERSHIP MANAGER EDITOR (Figure 9.44). This will display the CALCULATE screen that will allow you to choose options for calculation of ownership.
3. Choose the method for calculating ultimate percentage of ownership. You could choose either the DIRECT SHARE METHOD or the GROUP SHARE METHOD; the latter is used to calculate ownership for a group.
4. Select the CALCULATION METHOD to use after the system calculates the values: UPDATE METHOD, UPDATE % CONSOLIDATION, and OVERWRITE CURRENT OWNERSHIP VALUES WITH CALCULATED VALUES.
5. The PCTRL entry entered in Step 1 will be used to generate the METHOD_SYS and PCON_SYS values. Recall the consolidation method business rule that we discussed earlier (refer to Figure 9.40). During calculation of ownership, the system looks up the SHARE RANGE in the consolidation methods business rule table with the value entered in PCTRL and determines the corresponding METHOD CODE (METHOD_SYS) and APPLICABLE PERCENT CONSOLIDATION (PCON_SYS) values.
6. The POWN entry entered in Step 1 will be used to derive the POWN_SYS. The value of POWN_SYS depends on the method that you used to calculate the ultimate percentage ownership. You could choose either the DIRECT SHARE METHOD or the GROUP SHARE METHOD. In the DIRECT SHARE METHOD, the value of POWN_SYS is set to value of POWN, whereas in the GROUP SHARE METHOD, the POWN_SYS is calculated for the group by multiplying the rates through the hierarchy.

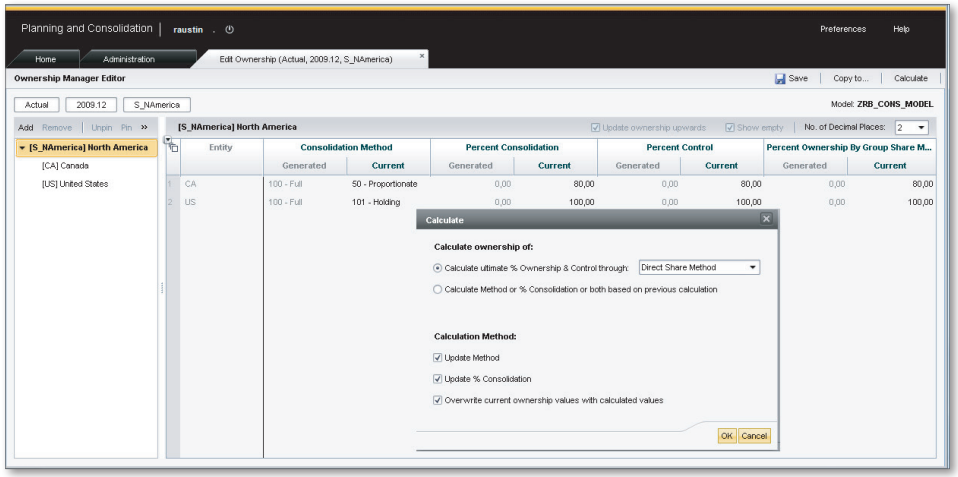


Figure 9.44 Calculate Ownership

For our example, we will enter ownership information in the Ownership Manager Editor for all groups. The ownership data is entered in the ownership model, as shown in Figure 9.45. Note that the ownership information is maintained for a context that consists of Category, Scope, and Time dimension members.

Let's look at some examples to understand how this information is maintained. Consider consolidation group G_CG1 (refer back to Figure 9.38). In this group, entity E9999 is the holding company, so we've set the value of METHOD of entity E9999 in consolidation group G_CG1 to "101." Entity E9999 owns 75% of E1000 and 100% of E3000 using the purchase method. So we've set the value of METHOD for entity E1000 and E3000 in the consolidation group G_CG1 to "100". Entity E1000 owns 80% of E2000 using the proportional method. Because the holding company of G_CG1 only holds 75% of E1000, and E1000 holds 80% of E2000, the percentage ownership of E2000 in the group G_CG1 is 0.75 × 0.80 = 0.60.

Similarly, consider consolidation group G_CG2. In this group, entity E1000 is the holding company, so we've set the value of METHOD of entity E1000 in consolidation group G_CG2 to "101". Entity E1000 owns 80% of E2000 using the proportional method. So we've set the value of PCON and POWN for entity E2000 in the consolidation group G_CG2 to 0.80.

OWNERSHIP

	G_CG1	G_CG1	G_CG1	G_CG2	G_CG2	G_CG2	G_CG3	G_CG3	G_CG3
	METHOD	PCON	POWN	METHOD	PCON	POWN	METHOD	PCON	POWN
E9999	101	1	1						
E1000	100	1	0.75	101	1	1			
E2000	50	0.6	0.6	50	0.8	0.8			
E3000	100	1	1				101	1	1
E4000	20	0.25	0.25				20	0.25	0.25

ICP=I_NONE

Figure 9.45 Ownership Data Entered in the Ownership Manager at the Group Level

Now that we've have entered the ownership information, we need to create the eliminations and adjustments business rules for the consolidation model to create adjustment entries during consolidation.

Configure Eliminations and Adjustment Business Rule in the Legal Model

For a consolidation model, the eliminations and adjustments business rule defines the calculations for consolidation. These rules link back to the method-based multiplier rule for calculating amounts and posting values. The eliminations and adjustment rule is used to read each source account that needs to be consolidated/eliminated and, depending on the fractions specified in the method-based multiplier rule, will post to the destination account, group account, and minority account.

The Eliminations and Adjustments business rule link is accessed from the Business Rules view under the Rules domain in the Administration workspace. This rule has two tabs: General and Details.

Table 9.8 lists the fields in the General tab of the eliminations and adjustments rule.

Field	Description
Adjustment ID	This is the unique identifier for the rule.
Description	This is a brief description of the business rule.
Source Audit	This is the audit member of the data that is read for consolidation. The audit member can be a base member, a parent member, or a list of members filtered using the `DIMLIST` property.

Table 9.8 Eliminations and Adjustments Business Rule Table—General Tab

Field	Description
Destination Audit	The elimination data created during the consolidation process is booked to this destination audit member. This data source should be a base member where the data source type = A (Automatic)
Group Type Filter	This is the CONSO_TYPE property value from the Group/Currency dimension.
Entity Property Filter	This field is used for performing intercompany elimination. The system looks for existence of this property in the Entity and Intercompany (INTCO) dimensions and performs eliminations only if the ENTITY and INTCO partners have the same value in this property.
Adjustment Type	This field can have the following values: ▶ Blank = All. ▶ E = Apply the rule only to entities that use the equity method. ▶ P = Apply the rule only to entities that use the proportional method. ▶ L = Apply the rule only to entities that use the leaving method. ▶ N = Apply the rule only to entities that use the new method.
Adjustment Level	This integer specifies the sequence of execution of rules.
Other Dimension Filter	This specifies the source data region that is consolidated (e.g., CATEGORY=ACTUAL).
Forced Destination Members	This can be used to force a Destination dimension member to contain a specific value (e.g., ENTITY=HQ).

Table 9.8 Eliminations and Adjustments Business Rule Table—General Tab (Cont.)

Table 9.9 lists fields in the Details tab of the eliminations and adjustments rule.

Field	Description
Source Account	This specifies the source data region to read data. The source account can be a base member, a parent member, or a list of members filtered using the DIMLIST property.
Source Flow	This specifies the data region to read data. The source flow can be a base member, a parent member, or a list of members filtered using the DIMLIST property.
Reverse Sign	If set to Y, this field will reverse the amount calculated. Please remember that because the adjustment entries pertain to elimination, the value calculated is negative by default. For example, if both the source accounts and the target accounts are asset accounts AST, then the result of the elimination will be negative by default.
Destination ALL Account	This specifies the destination account. Please refer to the explanation of math used to calculate the amount for this account (below).
Destination Group Account	This specifies the group account to be used. Please refer to the explanation of math used to calculate the amount for this account (below).
Destination Minority Account	This specifies minority account to be used. Please refer to the explanation of math used to calculate the amount for this account (below).
Destination Equity Account	If you use the equity method, this specifies the equity account to be used. Please refer to the explanation of math used to calculate the amount for this account (below).
Destination Flow	The consolidation data created during the consolidation process is booked to this destination flow member. This destination flow should be a base member.
Multiplier	This is the valid multiplier defined at the global definition level for consolidation models. This rule ID links back to the method-based multiplier rules table for calculating amounts.

Table 9.9 Eliminations and Adjustments Business Rule Table—Detail Tab

Field	Description
Force Closing	If set to `Y`, this field will create an extra record with closing flow (`FLOW=F_CLO`; `F_CLO` has flow type value = `CLOSING`).
Swap Entity—INTCO	If selected, this field will swap the Entity dimension with the INTCO dimension. This is especially used during calculation of goodwill. Please refer to the example of usage of this field later in this section.
Forced INTCO Member	This can be used to force a destination dimension INTCO member to contain a specific value.
Remark	This is a brief description of the business rule.

Table 9.9 Eliminations and Adjustments Business Rule Table—Detail Tab (Cont.)

Please note that the multiplier selected in the eliminations and adjustments business rule table links back to the method-based multiplier rules table. When the consolidation process is run for a consolidation group, the system does the following:

1. It reads ownership information pertaining to the consolidation group from the ownership model.
2. It reads each eliminations and adjustments business rule and determines the base amount upon which automatic adjustments are to be calculated (source data based on source accounts).
3. The elimination and adjustment rule identifies the method-based multiplier.
4. Based on the method-based multiplier rule and the method of consolidation assigned to a given entity, the system determines fractions to post to the destination account, group account, and minority account. The fraction for the destination account, group account, and minority accounts are in the All Formula, Group Formula, and Minority Formula fields, respectively, of the method-based multiplier table.
5. The system calculates the amounts to be posted to the destination account, group account, and minority account by multiplying the base amount and the fraction for each account determined in step 4.

6. The system posts the amounts to the specific destination account, group account, and minority account specified in the eliminations and adjustments table.

In summary, the settings made in the DETAILS tab of the eliminations and adjustments rules are used to read each source account that needs to be consolidated/eliminated, and depending on the fractions specified in the method-based multiplier table, posts to the destination account, group account, and minority account. The key information that is used in the process of determining these postings is the method by which each entity is consolidated in the consolidation group, the method-based multiplier rule assigned to each account, and the elimination calculation as specified in each rule.

Let's consider an example to understand how to use the eliminations and adjustments business rules. Figure 9.38, shown previously in the chapter, depicts the ownership tree. Consider consolidation group G_CG2. In this consolidation group, entity E1000, the holding company in the group, acquires 80% of entity E2000 using the proportional method. Let's say that E1000 acquires E2000 for $50,000. Figure 9.46 displays the books of company E1000, company E2000, and the total, elimination, and consolidated group balances.

We'll use the accounts specified in Table 9.10 and the data sources specified in Table 9.11 in our example.

Accounts	Description
1100	Cash
1200	Investments in subsidiaries (proportional)
1300	Goodwill
3100	Treasury stock—purchase
3200	Treasury stock—equity
3300	Treasury stock—proportional
3400	Common stock
3500	Retained earnings

Table 9.10 Eliminations and Adjustments Example—Account Master Data

Data Source	Data Source Type
INPUT	I
AJ_ELIM1	A
AJ_ELIM2	A
AJ_ELIM3	A
AJ_PROP	A

Table 9.11 Eliminations and Adjustments Example—Data Source Master Data

We'll load two records into the consolidation system that represent the investment of entity E1000 and the equity of entity E2000. Table 9.12 lists the records. We'll also load all initial balances for entities E1000 and E2000 as displayed in Figure 9.46. For this illustration, these records can be entered into the system using input templates.

Proportional Method

	Company E1000	Company E2000	Total	Elimination	Group
Goodwill				$18,000	$18,000
Cash	$100,000	$40,000	$140,000	-$8,000	$132,000
Investments	$50,000		$50,000	-$50,000	$0
Total	$150,000	$40,000	$190,000	-$40,000	$150,000
Common Stock	-$130,000	-$40,000	-$170,000	$40,000	-$130,000
Retained earnings	-$20,000		-$20,000		-$20,000
Minority Interest					
Total	-$150,000	-$40,000	-$190,000	$40,000	-$150,000

Company E1000 owns 80% of company E2000 for $50,000

Figure 9.46 Consolidation of G_CG2 (Proportional Method Example)

ACCOUNT	1200	3300
INTCO	I_E2000	I_E1000
ENTITY	E1000	E2000
TIME	2010.DEC	2010.DEC
FLOW	F_CLO	F_CLO

Table 9.12 Investment of Entity E1000 and Equity of Entity E2000

DATA SRC	INPUT	INPUT
CURRENCY	LC	LC
GROUP	NON_GROUP	NON_GROUP
CATEGORY	C_100	C_100
AMOUNT	50000	40000

Table 9.12 Investment of Entity E1000 and Equity of Entity E2000 (Cont.)

The eliminations and adjustment rules need to do the following:

1. Eliminate the investment account of E1000 for ownership that is common between the parent and subsidiary.
2. Eliminate the common stock of E2000 for ownership that is common between the parent and subsidiary.
3. Book excess value paid to goodwill. In this case, the book value of the common stock of company E2000 is $40,000. Company E1000 acquired 80% of company E2000. So company E1000 acquired $32,000 of the book value of the stock. The excess amount of $18,000 ($50,000 – $32,000) paid by company E1000 is booked to goodwill.
4. Company E2000 has $40,000 in a cash account. Company E1000 owns 80% of company E2000. The remaining 20% (100% – 80%) of the cash should be eliminated. The amount eliminated from the cash account should be $8,000.

To accomplish these steps, we'll create the following method-based multiplier rules, as shown in Table 9.13. Note that ZR3 is a method-based multiplier rule that has been defined to use only for the proportional method. Specifically, the type for the rule is set to Proportionate Method.

Rule ID	Description	Entity Method	INTCO Method	ALL formula	Group Formula	Minority Formula
ZR1	Stock Holders Equity	50	99	POWN	POWN	
ZR2	Shares: Subsidiary	101	99	1	1	

Table 9.13 Method-Based Multiplier Rules Used for Eliminations and Adjustment

Rule ID	Description	Entity Method	INTCO Method	ALL formula	Group Formula	Minority Formula
ZR3	Proportionate Method	50	99	1-POWN		

Table 9.13 Method-Based Multiplier Rules Used for Eliminations and Adjustment (Cont.)

We'll configure the elimination and adjustment business rule for consolidating the investment using the proportional method. We'll first set the header table that will contain the destination data sources to which adjustment entries will be posted (see Table 9.14). For example, the adjustment entry for ADJ1 adjustment ID is posted to the AJELIM1 data source.

Adjustment ID	Destination Data Source	Adjustment Type	Remark
ADJ1	AJ_ELIM1		Eliminates investments
ADJ2	AJ_ELIM2		Calculates goodwill
ADJ3	AJ_ELIM3		Calculates goodwill
ADJ4	AJ_PROP	Proportional	Eliminates portion not owned

Table 9.14 Eliminations and Adjustment Rule Header

Next, we'll configure the detail table that will link to the method-based multiplier rule to calculate and post to accounts (see Table 9.15).

Adjustment ID	Source Account	Reverse	Destination ALL	Destination Group	Destination Minority	Rule ID	Swap Entity—INTCO
ADJ1	1200		1200			ZR2	
ADJ2	1200	X	1300			ZR2	
ADJ3	3300	X	1300			ZR1	X
ADJ1	3300		3300			ZR1	

Table 9.15 Eliminations and Adjustment Rule Detail

The first entry (ADJ1) is used to eliminate the investments in subsidiaries (proportional) account. This adjustment entry uses rule ZR2, reads the amount in account 1200 (investments in subsidiaries [proportional]), and posts the same amount (because the ALL formula = 1) to the same account and data source AJ_ELIM1 with a negative value. Because the consolidation process creates an elimination entry, it sets the destination ALL account amount to a negative value even though the adjustment posts to the same account. Also, remember the signs related to accounts (AST +, LEQ –, INC –, and EXP +) when reading source accounts and posting to destination accounts.

The next two entries (ADJ2 and ADJ3) are used to calculate goodwill. ADJ2 uses rule ZR2 and reads the amount in account 1200 and posts the same amount to account 1300 (goodwill) and data source AJ_ELIM2. ADJ3 uses rule ZR1 and reads the amount in account 3300 (treasury stock—proportional) and posts the fraction of the amount owned (because ALL formula = POWN) and posts to account 1300 and data source AJ_ELIM3. The net effect of the two adjustments is equal to "value of the investment"—"value of treasury stocks (proportional) that are owned." This results in the goodwill. Note that ADJ3 has the Swap Entity INTCO checkbox selected. This ensures that the system will read the transaction that contains the treasury stock information posted to E2000 and INTCO I_E1000 and will post to E1000 and intercompany I_E2000. We need the swap entity and INTCO to post goodwill to E1000.

The fourth entry (ADJ1) is used to eliminate the remaining balance in the equity account. This adjustment entry uses rule ZR1 and reads the amount in account 3300 and posts the fraction owned (because ALL formula = POWN) to the same account and data source AJ_ELIM1 with a negative value.

Finally, the adjustment entry ADJ4 would execute and eliminate a portion of the balances of E2000 that is not owned by E1000. This would eliminate balances in the cash account and treasury stock (proportional) accounts. The eliminations are posted to the AJ_PROP data source.

After execution of the consolidation process for consolidation group G_CG2, we would expect to see the results tabulated in Table 9.16.

Entity	Account	Account Description	INPUT	AJ_ ELIM1	AJ_ ELIM2	AJ_ ELIM3	AJ_PROP	TOTAL
E1000	1100	Cash	100,000					100,000
E2000	1100	Cash	40,000				–8,000	32,000
E1000	1200	Investments in Subsidiaries—Proportional	50,000	–50,000				0
E1000	1300	Goodwill			50,000	–32,000		18,000
E1000	3100	Treasury Stock—Purchase						0
E1000	3200	Treasury Stock—Equity						0
E1000	3400	Common Stock	130,000					130,000
E1000	3500	Retained Earning	20,000					20,000
E2000	3300	Treasury Stock—Proportional	40,000	–32,000			–8,000	0

Table 9.16 Consolidation of Investment Results (Proportional Method)

Execute Consolidation Process

You are now ready to execute the consolidation process. SAP BPC 10.0 offers an intuitive interface to view, monitor, and execute the consolidation process. The consolidation process is managed from the Consolidation Central domain in the Home workspace of the Web Client. Click on the Consolidation Monitor view

to display the status for a category, time, and group selection (Figure 9.47). The screen provides the interface to perform the following activities:

- **Display status of consolidation**
  The entire consolidation process for an organization can be displayed and monitored from this interface.
- **Set and display work status**
  WORK STATUS must be enabled for the model before work status can be set.
- **Execute and monitor status of currency translation**
  The rate model applicable for consolidation must contain the exchange rate to run currency translation.
- **Execute and monitor status of consolidation**
  The ownership structure and data must be maintained in the ownership model before data is consolidated.

Two security tasks are related to consolidation:

- **View consolidation monitor**
  This task provides access to the CONSOLIDATION MONITOR view.
- **Run consolidation monitor**
  This task provides access to the currency conversion and the consolidation process from the CONSOLIDATION MONITOR view.

When using the currency translation process from the CONSOLIDATION MONITOR view, the member in the Entity type dimension of the rate model should be called GLOBAL for the currency conversion to execute correctly. You are not able to use custom logic when executing the currency conversion and consolidation from the CONSOLIDATION MONITOR view because these programs are called directly from the interface, and logic script that may contain custom code is not called.

The currency translation and consolidation process can be run in incremental or full mode. When the process is executed in incremental mode, the process will run the program only for those entities that were modified since the last execution of the program. The incremental mode process works faster because the program runs only for the entities that have changed since the last run. However, the incremental mode does not work in the following cases:

- Rates are changed in the rate model
- Ownership data has changed
- Business rules changes have occurred

In these cases, the process needs to be run in full mode.

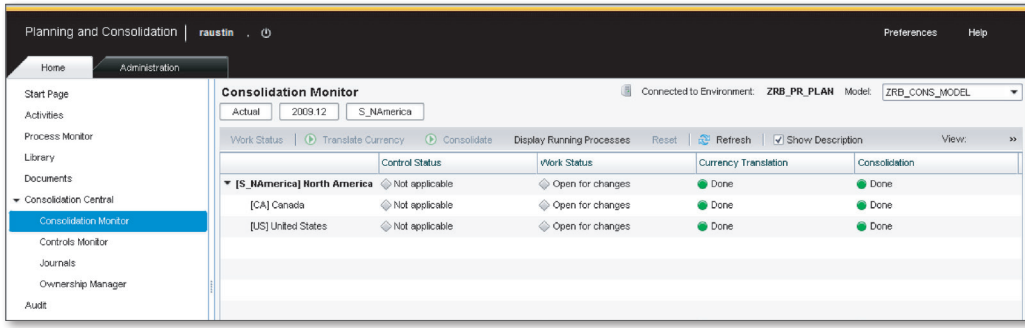


Figure 9.47 Consolidation Monitor View

9.4 Intercompany Elimination—Eliminations and Adjustments Business Rule

In the previous section, when discussed the consolidation of investments, we studied the usage of the ownership model, the consolidation methods, method-based multipliers rules, and the eliminations and adjustments business rule. In this section, we'll see how to use these rules to perform intercompany elimination. You can use intercompany elimination to eliminate intercompany transactions such as intercompany revenues and COGS, intercompany accounts receivable (AR) and accounts payable (AP), intercompany dividends, royalties, and so on. The consolidation logic used in consolidating investments is also used to eliminate intercompany transactions.

This method of intercompany elimination is used in legal consolidation. The elimination postings are generated when the consolidation process is executed for a group. Intercompany elimination applies elimination directly to the entity, unlike U.S. elimination, which posts elimination to an elimination entity that is associated with a common parent of entities in the entity hierarchy, between which an intercompany transaction takes place. The reverse amount is posted to a plug account (also called an elimination account or offset account).

Let's learn how this elimination works with an example. Say entity E1 sells goods to entity E2 for $1000. As a result of the transaction, E1 posts an AR transaction for $1,000 specifying E2 (e.g., INTCO = I_E2) as the trading partner. E2 posts an AP transaction for the same amount, specifying E1 as the trading partner. The elimination process eliminates the AR transaction of E1 for $1,000. The other side of the entry is posted to a plug/offset account. Similarly, the process eliminates the AP transaction of E2 for $1,000. The other side of the entry is posted to a plug/offset account. As a result of the transactions, assuming that the same plug/offset account is used for elimination of AR and AP, the plug/offset account would net to 0.

The elimination entry is posted to the same entity to which the original transaction was posted at the group level. The intercompany eliminations process posts the elimination entries to a destination data source specified in the eliminations and adjustments business rules table. This is done so that the initial value in the source account is not overwritten.

You'll follow these steps to perform intercompany eliminations:

1. Set up dimensions and properties.
2. Set up the ownership model. (We discussed the steps to set up this model in Section 9.3.4 when we studied the consolidation of investments.)
3. Set up the consolidation methods table, method-based multipliers rule, and eliminations and adjustment rules.
4. Execute the consolidation process from the Consolidation Monitor view.

Let's walk through the setup processes.

Set Up Dimensions and Properties

The consolidation of investments requires you to include the following dimension properties:

- `IS_CONSOL` **of DATASRC dimension**
  If the value of this property is set to `Y`, the data source participates in the consolidation run. If this property is set to `N`, the data source does not participate in the consolidation run.
- `DATASRC_TYPE` **of the DATASRC dimension**
  Data sources that have this property set to `A` (automatic) can be used as destination data sources in business rules.

- **`IS_CONVERTED` of the DATASRC dimension**
  If the value of this property is set to `Y`, the data source participates in the currency translation. If this property is set to `N`, the data source does not participate in the currency translation.
- **Intercompany dimension**
  To support intercompany elimination, the SAP BPC application should include this dimension with the type `I`.
- **`Entity` property of the Intercompany dimension**
  The value of this property should match the member ID of the Entity dimension.
- **`TYPELIM` of the Account dimension**
  This property should contain a valid elimination type as defined in the elimination and adjustment business rule. This property value is used to filter specific accounts that are subject to specific business rules.

Set Up Ownership Model

We discussed the steps to set up the ownership model in Section 9.3.4 when we studied consolidation of investments. Follow the steps given there to assign entities to consolidation groups.

Set Up Business Rules

The steps to set up the business rules for intercompany elimination are similar to the steps we used to set up rules for the consolidation of investments. Use the consolidation methods table, the method-based multiplier table, and the eliminations and adjustment header and detail tables.

For intercompany elimination, use method-based multiplier rule ZR4—IC eliminations (Table 9.17). The rule contains formulas in the method-based multiplier table.

Rule ID	Description	Entity Method	INTCO Method	ALL Formula	Group Formula	Minority Formula
ZR4	IC Elimination	100	99	1	1	

Table 9.17 Rule ZR4—Intercompany Elimination (For Full Ownership)

In the General tab for the eliminations and adjustment rule, set the destination data source to which the elimination entries should be posted. Table 9.18 has this

entry to specify the destination data source. In this case, the elimination entries are posted to data source AJ_ELIM.

Adjustment ID	Source Data Source	Destination Data Source
ADJICP		AJ_ELIM

Table 9.18 Intercompany Elimination – Header

In the DETAILS tab for the eliminations and adjustment rule, eliminate intercompany AR and AP accounts. Post the other side of the entry to plug/offset the account. Refer to Table 9.19 for the configuration. AR represents all intercompany accounts receivable accounts, AP represents all intercompany accounts payable accounts, and ARAPPLUG represents the plug/offset account.

Source Account	Reverse Sign	Destination ALL account	Destination Group account	Destination Minority Account	Rule ID
AR		AR	APARPLUG		ZR4
AP		AP	APARPLUG		ZR4

Table 9.19 Intercompany Elimination – Detail

Execute Consolidation Process

Execute the consolidation process for eliminating the intercompany transactions from the CONSOLIDATION MONITOR view in the Web Client.

Now you should understand how to set up intercompany elimination using the eliminations and adjustments business rule. In the next section, we'll discuss the use of journals and see how to create and use journal templates in SAP BPC.

9.5 Journals

In simple scenarios, some companies enter manual journal entries to post entries related to the consolidation of investments. You can also use journals for making manual adjustments to data in a model; these entries are sometimes referred to

as *top-side entries*. During the review and reconciliation of financial data, variances may be detected; to fix these variances, you post differences as a journal entry.

Journals are stored in journal tables. You must create a journal template, which you use to create and maintain journals, before a journal entry can be maintained for a model. You can maintain only one journal template for a model

A user should be granted sufficient authorizations to create and post journal entries; we recommend creating a task profile that provides the necessary authorization for users to perform various journal tasks. The following tasks relate to providing access for users to manage journals: manage journals, edit journals, post journals, view journals, unpost journals, lock/unlock journals, and reopen journals. You can perform the following types of journal activities if you have the necessary authorization:

- **Create a journal entry**
  When you create a journal entry, a unique journal ID is created.
- **Open a journal entry**
  Journals created earlier can be reopened.
- **Copy journals**
  Existing journal entries can be copied to create new ones.
- **Delete journals**
  Journal entries that have not been posted can be deleted.
- **Post journals**
  Journals can be posted to make them available in the model for reporting.
- **Unpost journals**
  A journal entry can be unposted that was previously posted. Unposting a journal reverses the entry in the model and allows a user to modify the journal and post it again.
- **Ungroup journals**
  Journal entries can be grouped together when they are created. This option provides the ability to ungroup journals.
- **Reopen journals**
  Reopening a journal is the process of opening a prior journal transaction and posting it to a different set of accounts. To do this, the system should be open for reopening journals at the model level, and the source and destination accounts for the journal entries reopened should be defined.

- **Report journals**
  A journal entry can be reported using this option.
- **Lock and unlock journals**
  A journal entry can be locked or unlocked if the user has the authorization for those actions. When a journal entry is locked, no further changes are possible.

9.5.1 Create Journal Template

Let's look at an example showing how to create a journal template in SAP BPC. You can access the journal template for a model from the JOURNALS view under the FEATURES domain in the ADMINISTRATION workspace (Figure 9.48).

A journal template can be created for a planning or a consolidation model. The menu options listed in the JOURNALS view enable a journal template to be maintained. Only one journal template can be created for a model.

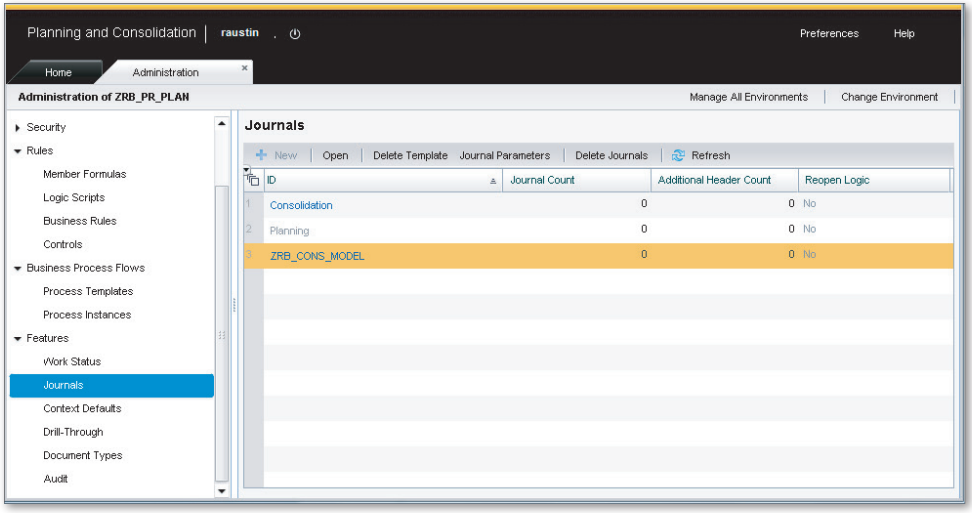


Figure 9.48 Creating a Journal Template—Part A

1. Select the model for which you want to create a journal template, and click the NEW menu option. The NEW menu option is not enabled if a template exists for the model. The existing template has to be deleted if a new template is to be created. An existing journal template cannot be deleted until all of the journals entries created for the model using that template are deleted. Selecting the NEW

menu option presents the options to define the configuration for the template (Figure 9.49).

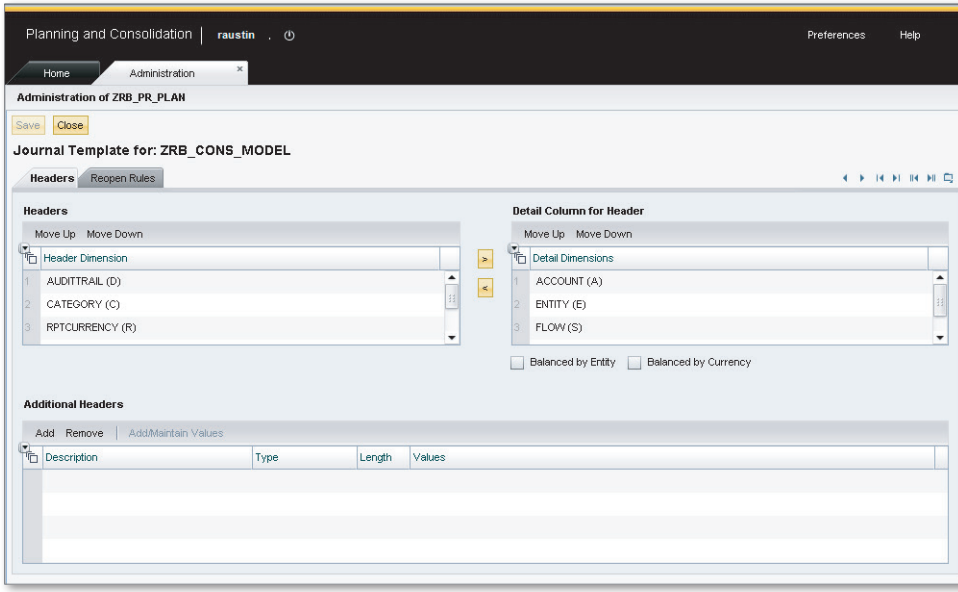


Figure 9.49 Creating a Journal Template—Part B

2. Configure the journal template using these options:
   - In the HEADERS tab, select the dimension to be displayed in the HEADER and DETAIL areas of a journal. Journal entries can be balanced by entity and/or currency. When the checkbox for BALANCE BY ENTITY is checked, the journal sums entries by entity and ensures it is balanced (i.e., that the debit amount equals the credit amount). This also applies when the checkbox for BALANCE BY CURRENCY is checked. When both BALANCE BY ENTITY and BALANCE BY CURRENCY are checked, all combinations of entity and currency are subtotaled and ensured they are balanced. Additional header fields can be added here.
   - In the REOPEN RULES tab, you specify the rules for reopening journals. Specify the dimension member and property values based on account, entity, flow, and intercompany dimension. They constitute the source journal entries. A journal that is reopened based on the rule specified in the source can be posted to a destination. When the reopened journal entries are to be posted to a destination, a destination rule is specified.

- Some journal parameters can be defined in the ADMINISTRATION page using the JOURNAL PARAMETERS menu option. You can set the parameters to enforce balanced journals and select whether journals can be reopened.

You should now understand how to create a journal template. Another important task is managing journals.

9.5.2 Manage Journals

To manage journals, follow these steps:

1. Journal entries can be viewed and managed from the JOURNALS landing page. Access the JOURNALS landing page by selecting the JOURNALS view under the CONSOLIDATION CENTRAL domain in the HOME workspace of the Web Client (Figure 9.50).

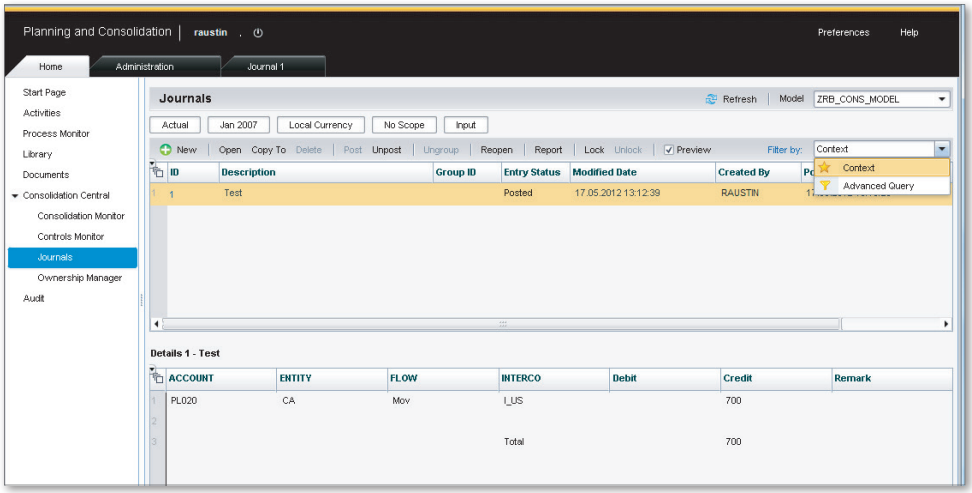


Figure 9.50 Journal Landing Page

2. Select the model for which you want to manage journals in the upper-right corner of the JOURNALS view. You can view journals by selecting the CONTEXT or ADVANCED QUERY options. When the CONTEXT option is selected, the journals are displayed based on the member selections in the context area. When ADVANCED QUERY is selected as the filter option, a query dialog box is presented to specify the criteria for selecting the journals (Figure 9.51). To view the details

of a journal, select the journal in the upper area. The details of the journal are displayed in the bottom if the Preview checkbox is selected in the menu area.

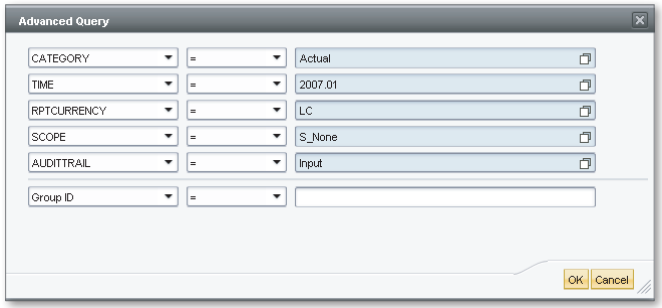


Figure 9.51 Advanced Query Filter Criteria

The journals shown on the landing page display the status, date posted (if a journal was posted), and who posted the document. The menu options that are enabled for a journal depend on the status of the journal. For example, a journal with a saved status will have the Post and Delete options enabled.

3. To create a new journal, click on the New menu option. When the option to create a journal is selected from an EPM report, the Web Client interface to create a journal entry is displayed (Figure 9.52). You have two options when creating a journal:
   - Multiple Header can be checked when you need to generate more than one journal entry with the same values but different header selections. When you select the Multiple Headers checkbox, you are able to select a dimension that belongs to the header area in your journal template and select one or more members for that dimension. When the values entered are saved, the entries are grouped together using a common group ID.
   - Multiple Values can be checked when you want to generate journal entries with different values for different header selections. When you select the Multiple Values checkbox, you are able to select a dimension that is in the header area in your journal template and select one or members for that dimension. The settings provide the interface to enter different values for each of the selected members.

4. To open a journal, select the journal, and click on the OPEN menu option.
5. To delete a journal, select a journal, and click on the DELETE menu option.
6. To post or unpost a journal, click on the POST or UNPOST menu options. The POST journal option make available the journal entry entered in the journal template for reporting in the model.

Additional menu options are available to ungroup, reopen, report, lock, and unlock. The menu options that are available for a journal depend on the status of the journal.

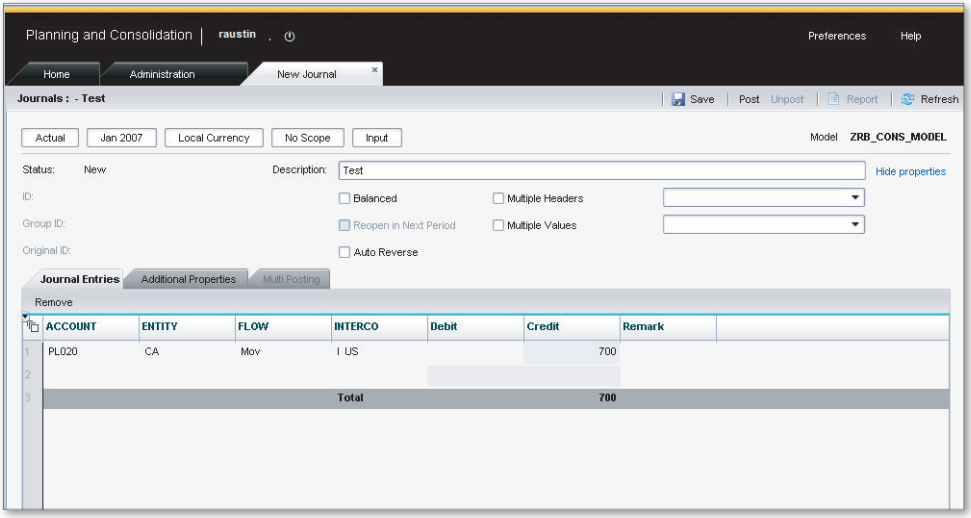


Figure 9.52 Creating a New Journal Entry

Next, we'll discuss the use of a BAdI for managing journals.

9.5.3 Journal BAdI

SAP BPC 10.0 for NetWeaver introduced a new BAdI for managing journals. This BAdI performs calculations and/or creates additional records when it is saved or posted. The name of the enhancement spot is `UJJ_CALCULATED_AMOUNT`. A developer can create code based on this enhancement to manage journals depending on business requirements.

Note

Journals do not post delta values. If you want to make a change to a journal that has been posted, you must unpost the journal, make the change, and then post the journal. During concurrency locking, the package size is set to 99,999,999 for journals to avoid locking.

The `ENABLE_JRN` property can be used in any dimension to allow use of the dimension member during the creation of journals. If the value of this property for a dimension member is set to `Y`, then this member can be used in journal entries.

Data Manager packages are available to export and import journals.

9.6 Summary

In this chapter, you learned the steps in the consolidation process and studied how to use business rules in SAP BPC to perform common tasks in a planning and consolidation model.

We used scenarios to define the steps to configure business rules to perform carry forward balance, account transformation, and validation (control). We also studied the intercompany matching process that is used to reconcile intercompany transactions reported by an entity and its trading partner. We used the intercompany booking business rule to book mismatches in intercompany transactions. We learned about two methods of eliminating intercompany transactions—U.S. elimination and elimination using the eliminations and adjustments business rule—and discussed the business rules used in these methods.

Next, we discussed the purchase, equity, and proportional methods used in the consolidation of investments. We detailed the consolidation methods, method-based multipliers, and the eliminations and adjustment rules that are used to consolidate investments.

Finally, we discussed the use of journals and the process of setting and using journals in a model. In the next chapter, we will discuss how to configure business process flows (BPFs) for sequencing execution of business processes and also look at some of the new features and enhancements introduced by SAP in recent support packs of the SAP BPC software.

You can use the Business Process Flow (BPF) feature for sequencing a business process into defined steps for execution inside a planning and consolidation model.

10 Business Process Flow and Enhancements

The release of SAP BPC 10.0 provides several enhanced features to aid in planning and consolidation. Business Process Flow (BPF) is one of the areas where an improved interface is provided via the Web Client to define and use the functions for workflow management. In this chapter, we discuss how to use the BPF feature to organize the execution of a business process and to monitor the status of planning or consolidation processes in the SAP BPC system.

In Section 10.1, we'll discuss how a BPF is defined and managed inside the SAP BPC system.

In Section 10.2, we will also take a look at some of the new features and enhancements introduced by SAP in recent support packs.

Finally, in Section 10.3, we will conclude by summarizing the lessons taught by this book.

10.1 Business Process Flow

SAP Business Process Flows (BPFs) aid in creating, maintaining, and executing a sequence of steps to fulfill the requirement of a business process. BPFs help to monitor and understand the current status of a planning or consolidation process. For example, the creation of plan data or the consolidation of financial data in an organization is composed of a list of steps. A BPF provides the interface to define the steps and the features to execute them.

For users who are familiar with the SAP NetWeaver Business Planning and Simulation tool, this feature is similar to the Status and Tracing System (STS) that supports a workflow model for a variety of applications. This is especially relevant

for planning and consolidation processes that involve several steps and where the process involves a review before the plan data is approved or the consolidated data is considered final.

BPF is defined as a sequence of steps with each step corresponding to an action. When all of the steps defined in the BPF are completed, the planning or consolidation process is realized. Each BPF is geared toward a particular process; for example, a BPF may be defined for the yearly sales planning process and may consist of steps where data is input, reviewed by management, and approved.

BPF supports the following features:

- As a central menu to access the functions in a model; for example, from a BPF, you can access an input form, a report, and so on.
- As a monitoring tool for multiple levels of review and approval before data is considered final. This is especially helpful in the case of company-wide planning that may go through multiple iterations and approvals.
- As a tool to send email notifications after completion of a task in the process.

The BPF features inside SAP BPC support the creation of a template to define the steps inside an SAP BPC model and execute an instance of the template. Two views are available to execute and monitor the BPF process: the Activities view and Process Monitor view. The Activities view displays the activities for an instance and provides the interface to complete the actions associated with it. The Process Monitor view allows you to monitor the entire BPF across the enterprise in one glance.

Next, we'll discuss the steps for configuring a BPF.

10.1.1 Creating a Business Process Flow Template

Creating a BPF template is the first step in developing a BPF for your application. You create and maintain BPF templates in the Web Client. Follow these steps to create a new BPF template (keeping in mind that you first need to obtain the necessary authorization):

1. To create a template, select the Process Templates view under the Business Process Flows domain in the Administration workspace inside the Web Client (Figure 10.1). To create a template, click on the New menu button.

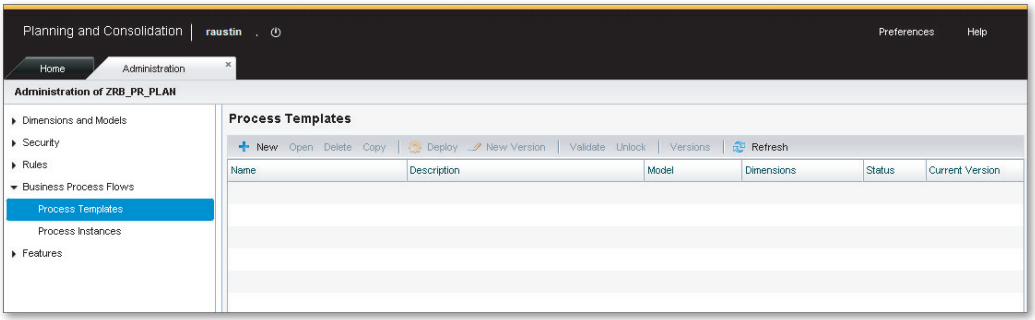


Figure 10.1 Creating a Business Process Flow Template—Part A

2. The screen to create a BPF template is composed of two tabs: the PROCESS SETTINGS and ACTIVITIES tabs (Figure 10.2). In the PROCESS SETTINGS tab, define the settings for the template. You define the following in this tab:

   - NAME and DESCRIPTION
     Enter a name and description for the template.

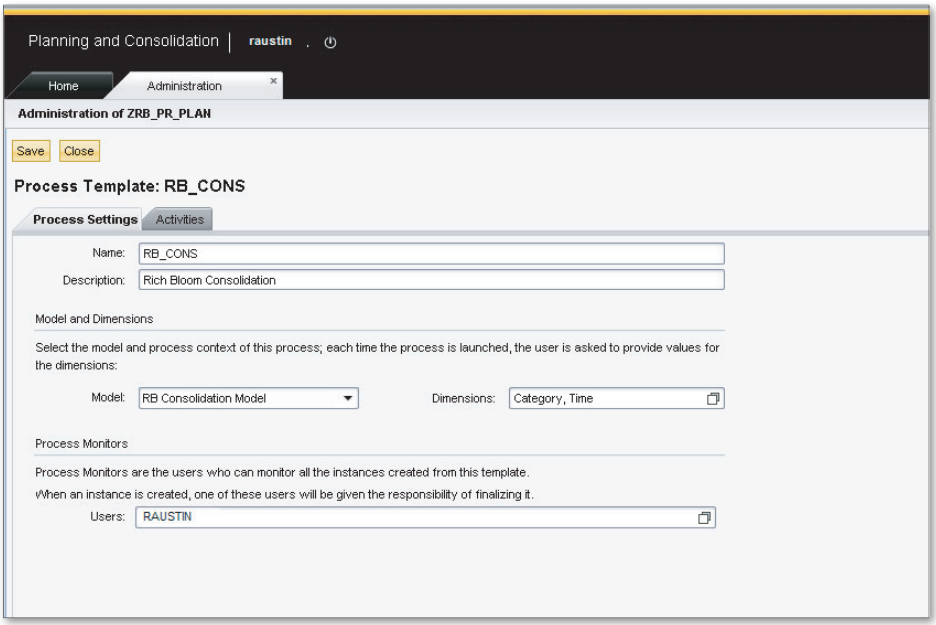


Figure 10.2 Creating a Business Process Flow Template—Part B

- Model and Dimensions
  Identify the model for which this template will be used. Select the dimensions to identify a BPF instance. When a BPF is run, it is associated with a unique instance that is based on the dimension members specified here. We have selected Category and Time dimensions for our example. A BPF instance is created from a BPF template, and more than one BPF instance can be created for each template. The Time dimension is one of the dimensions that is automatically selected here as one of the dimensions and cannot be deselected.
- Users
  Select the user who will be responsible for monitoring the instance created for this template. This user will also be executing the finalize process for an instance created for this template.

3. You are now ready to define the activities for the templates. (Activities were called "steps" in the 7.x versions of SAP BPC.) Click on the Activities tab, and then click on the New menu button to create an activity for the template (Figure 10.3).
   - Name and Instruction
     Enter a name and instruction for the activity.
   - Dimensions and Users
     In this section, you identify the driver dimension and select the member(s) to use for the driver dimension. The member will be used as the context for the activity. Specify the property of the driving dimension that identifies the performer of the activity for each member. The term "performer" replaced the term "owner" in the 7.x version of SAP BPC. A performer can be a single user, more than one user, or a team of users. If a review of the activity is required, check the Require Reviewer checkbox, and select the property of the driver dimension that will be used by the system to identify the reviewer for each dimension member.
   - Opening Criteria
     This setting determines when an activity is open. Two options are available for configuring the open criteria: All or Matched. When the Opening Criteria is set to All, the activity can be started only when all of the previous activities for all process contexts have been completed. When the setting is set to Matched, the activity can be started for a specific context if the previous activity for the same context is complete. Check the Allow Reopen box

when you want the flexibility for an activity to be reopened even after it is complete.

- Workspaces
  Click on Create to define the actions associated with the activity.

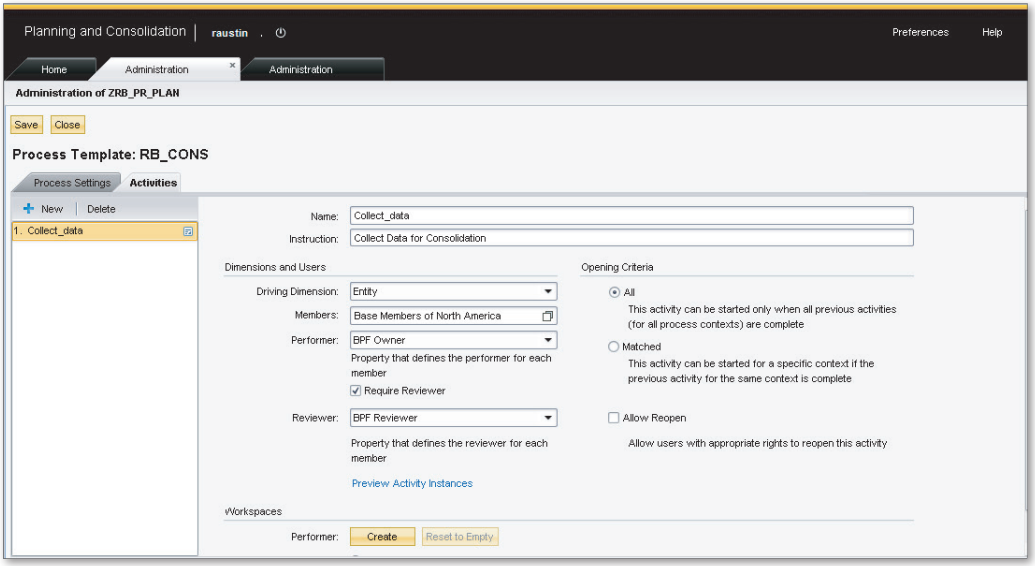


Figure 10.3 Creating a Business Process Flow Template—Part C

4. Three menu options are available to add actions to the activity workspace: Add Content, Add New Report, and Add Hyperlinks. The Add Content option lets you select an existing report, input form, or workspace to the activity workspace. With Add New Report, you can create a new report. Finally, Add Hyperlinks provides a set of predefined actions to select from, including the following:
   - Run Data Manager packages
   - Set work status
   - Run consolidation functions
   - Link to external URLs
   - Link to BPC library and unstructured documents
   - Open audit functions

Figure 10.4 shows an example of a hyperlink added using the ADD HYPERLINKS option.

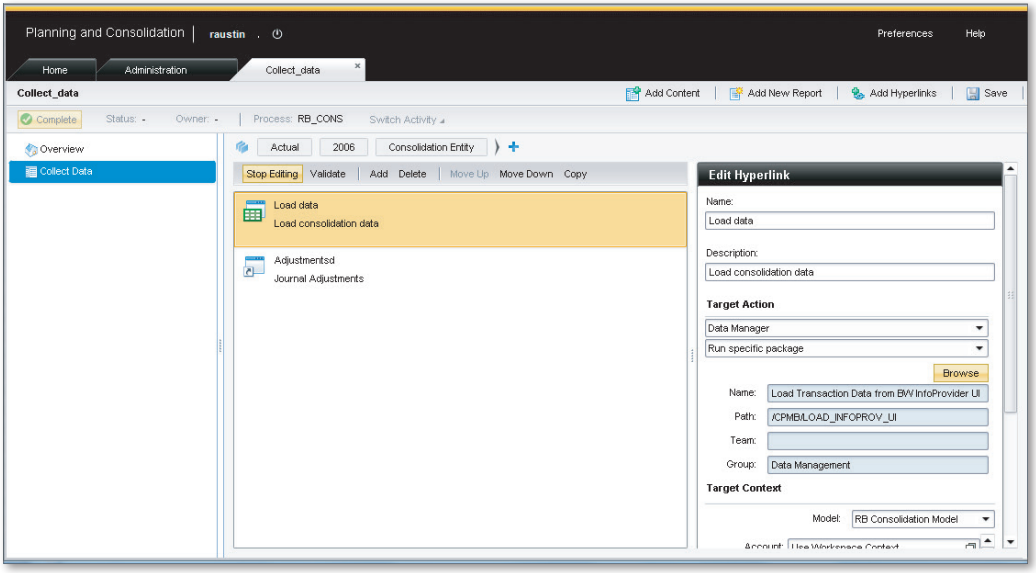


Figure 10.4 Creating a Business Process Flow Template—Part D

5. An action can include one or more subactions, and each subaction can be linked to one or more related actions. After adding the content, click the SAVE button.
6. You can create additional activities and corresponding actions as required for your BPF. When a template is initially created, it is in draft version status. The template has to be validated before it can be used. Validate the template using the VALIDATE menu option.
7. After validating the template, click on the DEPLOY menu option. You can now create an instance of the template.
8. After a template is deployed, you cannot change the template unless you use the NEW VERSION menu option to create a new version for the template. There are additional menu options to lock, archive, and copy a template with another name. You can archive a template only if there are no active instances of the template.

Now that you've seen how to create a BPF template, we'll discuss how to create a BPF instance.

10.1.2 Creating a Business Process Flow Instance

The user who has been identified as the process owner or performer of the template can create a BPF instance from a BPF template. Instance creation is wizard-driven and consists of five steps (Figure 10.5), as follows:

1. The first is the SELECT PROCESS step. Select the template from the list of steps, and click on NEXT.
2. Select the instance owner in the SELECT OWNER step. The system will default to the process owner or performer that we defined when creating the template, but the owner can be changed to a different user. Click on NEXT.
3. Select the members that will uniquely identify the instance in the SELECT CONTEXT step. If there is another active instance running with the same dimension member selection for the BPF template, the system will not allow an instance to be created. Click on NEXT.

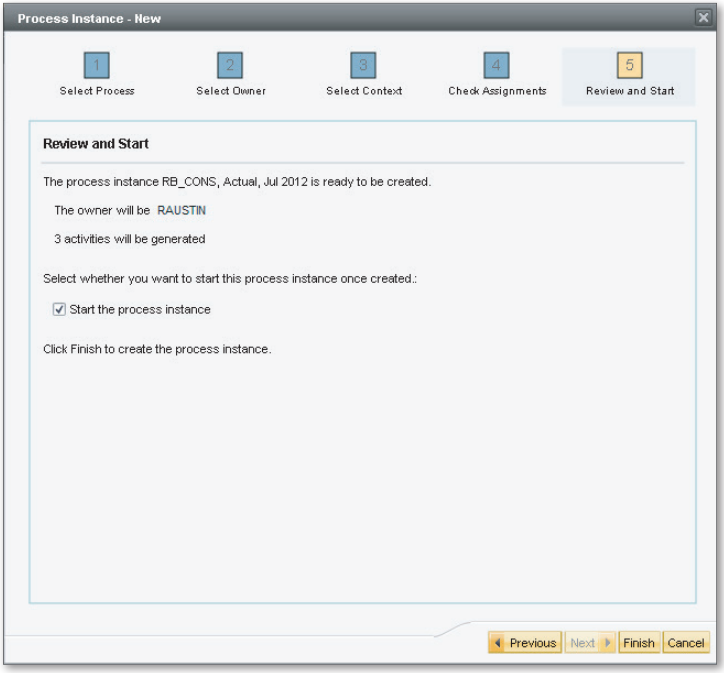


Figure 10.5 Creating a Business Process Flow Instance

4. In the Check Assignments step, the performer and reviewer for the instance that will be created are displayed. Here you have the options to select a new performer and reviewer. Click on Next.
5. The Review and Start is the final step in the creation of the instance (Figure 10.5). Select the checkbox if you want to start the instance immediately. If this checkbox is not checked, the instance has a suspended status and must be manually started using the Start menu option. Click on Finish.

A confirmation message indicates that the instance was created. After the BPF instance has been created, it's ready for use.

10.1.3 Executing Activities

After an instance is created, the activities for the instance can be executed from the Activity view in the Home workspace by the user identified as the performer (Figure 10.6). Click on the Activity view. The top-right area displays the status of the activities for the instance. Selecting the status in the top area displays the activities in the lower area of the screen. Only the activities the user is involved in are listed here. The user can perform the actions for an activity displayed in the lower area by clicking on the activity or by clicking the Open menu button. This displays the list of actions for the activity (Figure 10.7).

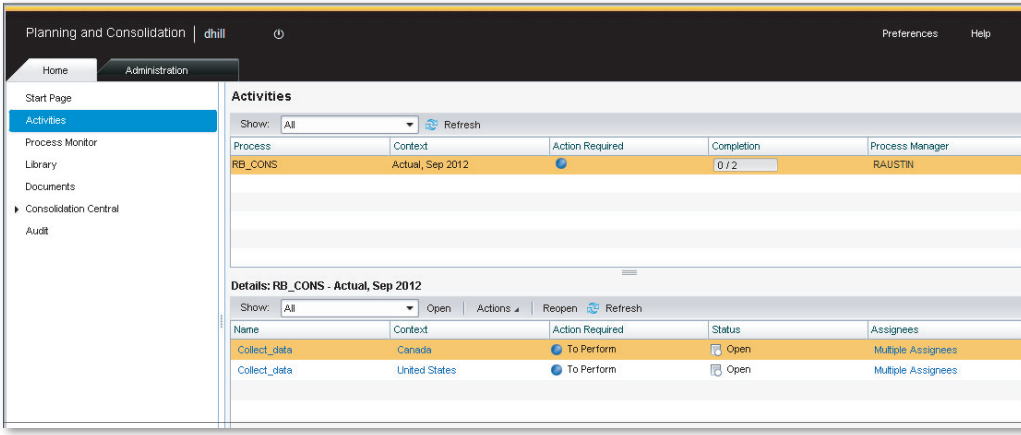


Figure 10.6 Activities for an Instance

After the activity is completed, click the COMPLETE button to complete the activity, or click the SUBMIT button to send it to a reviewer when a review is required of the actions performed. If the activity requires review, the reviewer for the activity can APPROVE or REJECT it (Figure 10.8). The email process can be enabled in the SAP NetWeaver BW system to automatically send an email to the reviewer when an action has been submitted by a performer. After the reviewer reviews and approves the activity, the status of the activities is updated (Figure 10.9).

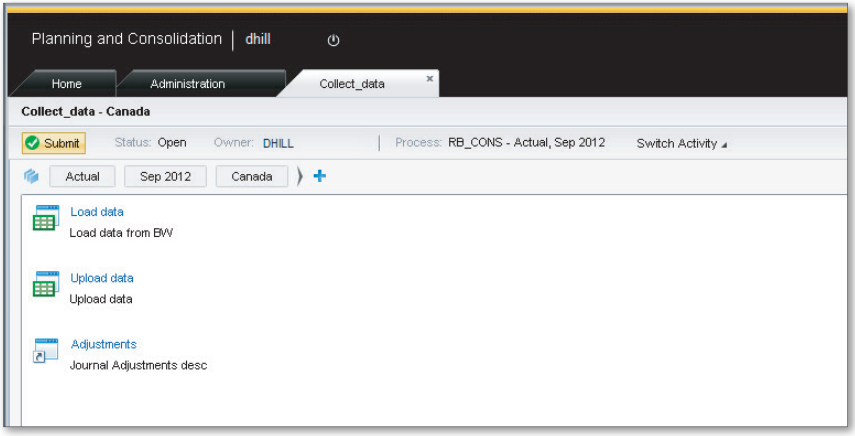


Figure 10.7 Executing Actions for an Activity

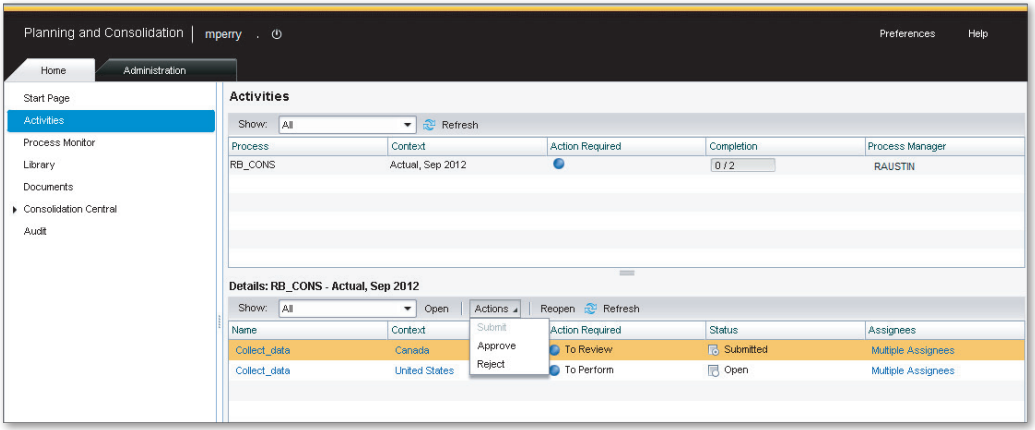


Figure 10.8 Reviewing an Activity

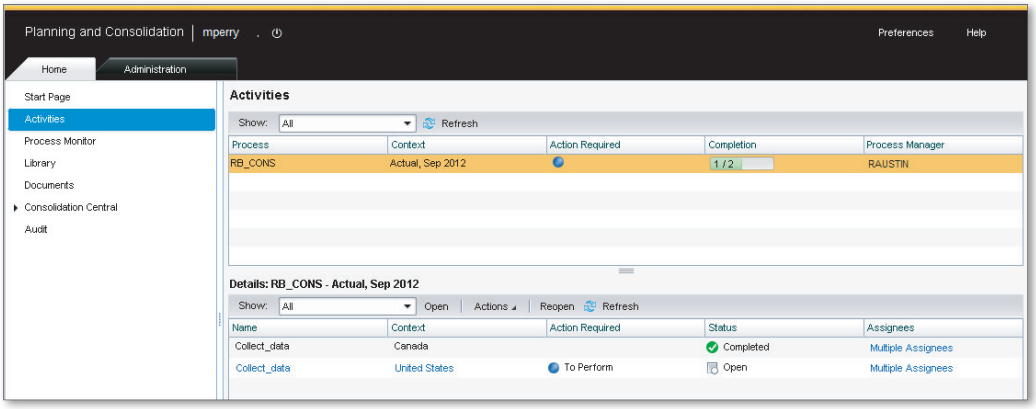


Figure 10.9 Summary Activity Status for an Instance

10.1.4 Process Monitor

The Process Monitor view provides the percentage completion and status of an instance (Figure 10.10). This is available to the user who has been selected as the instance owner (performer) of the template. You can display the activities for an instance by selecting and clicking on it. The details of an instance can also be displayed by selecting the Open menu option in the Process Monitor view. The process owner can finalize an instance after all of the activities are completed from this view.

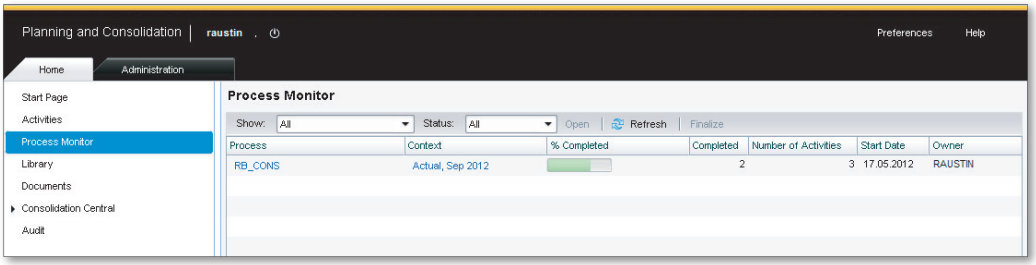


Figure 10.10 Process Monitor View

We have seen how BPF helps to organize the execution and to monitor the status of the planning or consolidation processes. In the next section, we look at some new features introduced in recent support packages of SAP BPC 10.0 version for NetWeaver.

10.2 New Features for SAP NetWeaver

Several new features have been introduced in recent support packs for SAP BPC for NetWeaver. In this section, we'll explore some of the changes introduced in support packages (SPs) 6, 7, and 8.

10.2.1 SAP HANA

SAP HANA for SAP BPC for NetWeaver is delivered with SP6. SAP HANA is delivered for superior performance on high volume transactions. In the regular SAP BPC, significant time is spent on query parsing and calculation. With SP6, the entire calculation and aggregation can be done on SAP HANA, making query execution faster.

10.2.2 Firefox as Web Browser

Firefox 8.0 is additionally supported as the browser that can be used to access SAP BPC on the web from SP7. The EPM plugin for Firefox can be installed from the download center in the Web Client to launch the EPM tool for reporting.

10.2.3 Drill-Through for Web Reports

The drill-through feature allows SAP BPC users to directly access information in a source system via a URL. It lets users jump from an SAP BPC report to an external URL or an SAP NetWeaver BW system by passing parameters. SP7 extended the functionality of drill-through web reports.

10.2.4 Disaggregate Data from Parent Node Cells Using BAdI

Data cannot be planned at a parent node level in SAP BPC. However, there is often a need to plan at a node level (hierarchy node) and to apportion that data to the base members based on some logic. A new method called `CUSTOMIZE_PROPERTIES` in BAdI enhancement `UJQ_SHARED_QUERY` is now available to enable data entry for hierarchy nodes (with the property value `CALC = N`) when entering data in an input form. The data entered at the node level can then be allocated to base members using logic when data is updated in the write-back BAdI. This is available from SP7.

10.2.5 Reporting on Archived Audit Data

There is a new feature introduced in SP7 to report on archived audit data. This enables the audit data in the system to be kept to a minimum and allows for archiving of older audit data. The archived data can be then reported.

10.2.6 Defer Layout in a Web Report

When you are creating or modifying an SAP BPC report in EPM, this feature lets you defer the update of the report until after it is complete before applying any changes. SP8 provides this feature for web reports developed in the Web Client.

10.2.7 Export Web Report to CSV file

SP8 offers the ability to export data displayed in a web report as a CSV file.

10.2.8 Zero Suppression Option in a Web Report

SP8 offers the ability to suppress zero data in a web report offering the same functionality when developing an SAP BPC report using the EPM Excel add-in.

10.3 Conclusion

The contents of this book were distributed into four segments. In the first segment, which consisted of Chapter 1 and Chapter 2, we provided an introduction to financial planning and consolidation. We offered an overview of EPM and discussed the use of the SAP BPC application to support the financial planning and consolidation process.

In the second segment, which consisted of Chapter 3, Chapter 4, and Chapter 5, we delved into the concepts and terminologies that are used in the SAP NetWeaver BW and SAP BPC systems. We presented a case study for a model company, Rich Bloom, which is a clothing retailer headquartered in the United States that has a presence in Germany and England as well. We discussed the development of data models and the configuration of objects to support the requirements of this model company. We also discussed the configuration of input forms that allows users to plan and report.

In the third segment, which consisted of Chapter 6, Chapter 7, Chapter 8, and Chapter 9, we presented topics on business logic, process management, collaboration, and supporting tools that can help in configuring the SAP BPC model to meet your organization's planning and consolidation requirements. In this segment, we discussed in detail topics related to consolidation such as using business rules to perform specific consolidation tasks.

In the fourth segment, found in Chapter 10, we looked at BPF and how it can help to execute a multi-step planning or consolidation process. We also looked at some of the new features introduced in the recent support packs for the SAP NetWeaver system.

After reading this book, you should understand how SAP BPC serves as a complete solution for an organization's planning and consolidating needs. The intuitive Excel-based interface used for reporting, coupled with powerful customization features, makes it a sound value proposition. In addition, the ability of SAP BPC to integrate with SAP NetWeaver BW in the SAP NetWeaver version provides additional power to leverage the architecture of SAP NetWeaver BW, which is based on a star schema. This architecture provides not only enhanced performance but also a secure environment for managing the planning and consolidation process.

It has been a pleasure to present this book to you. We hope you've found the information useful and informative, and we look forward to helping you with your training and project implementation needs in the future.

The Authors

Sridhar Srinivasan is a solution architect for Zebra Consulting, Inc., in Houston, Texas. He holds a bachelor's degree in engineering from the College of Engineering in Guindy, Chennai, India. He graduated from the Master of Business Administration program at Duke University's Fuqua School of Business with a specialization in finance.

Sri has over two decades of experience in information technology and has been working with SAP products since 1995. He is certified by SAP AG for the SAP BuinessObjects Business Intelligence and SAP Business Planning and Consolidation (BPC) products. He has been directly involved with implementations of solutions in these areas for clients in the United States and has functioned as a lead consultant in managing the entire lifecycle of SAP BPC project implementations, from the project preparation phase through post-implementation support. His expertise is in the functional, technical, and project management areas of the implementation.

Sri lives with his family in Houston, Texas. He can be reached at *sri.srinivasan@zebraconsultinginc.com*.

Kumar Srinivasan is a senior business intelligence/SAP business planning and consolidation consultant for Zebra Consulting, Inc., in Houston, Texas. He holds a master's degree in finance from Loyola College in Chennai, India. He is also a certified cost accountant.

Kumar has almost two decades of experience in information technology and has been working with SAP products since 1999. He is certified by SAP AG for the SAP BusinessObjects Business Intelligence and SAP Business Planning and Consolidation (BPC) products. Kumar has expertise providing solutions in the areas of SAP NetWeaver Business Warehouse (BW), SAP Business Planning and Simulation (BPS), SAP BI Integrated Planning (BI-IP), and SAP BPC. He provides diverse solutions to clients in areas that cover both back-end

configuration and development of front-end user interfaces. His articles on topics related to SAP NetWeaver BW and BPC have been published in "BWExpert."

Before working with Zebra Consulting, Inc., Kumar was employed with The World Bank as an information officer. During this time, he worked with relational database management systems, including Ingress, Informix, and Oracle. He also has extensive experience in the development of applications using messaging tools such as Lotus Notes.

Kumar lives with his family in Cary, North Carolina. He can be reached at *kshrini77@yahoo.com*.

Sri and Kumar are the authors of *SAP NetWeaver BI Integrated Planning for Finance*, published by SAP PRESS in 2007.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Service Pages

The following sections contain notes on how you can contact us.

Praise and Criticism

We hope that you enjoyed reading this book. If it met your expectations, please do recommend it, for example, by writing a review on http://www.sap-press.com. If you think there is room for improvement, please get in touch with the editor of the book: emily.nicholls@galileo-press.com. We welcome every suggestion for improvement but, of course, also any praise!

You can also navigate to our web catalog page for this book to submit feedback or share your reading experience via Facebook, Google+, Twitter, email, or by writing a book review. Simply follow this link: http://www.sap-press.com/978-1-59229-445-9.

Supplements

Supplements (sample code, exercise materials, lists, and so on) are provided in your online library and on the web catalog page for this book. You can directly navigate to this page using the following link: http://www.sap-press.com/978-1-59229-445-9. Should we learn about typos that alter the meaning or content errors, we will provide a list with corrections there, too.

Technical Issues

If you experience technical issues with your e-book or e-book account at SAP PRESS, please feel free to contact our reader service: customer@sap-press.com.

About Us and Our Program

The website http://www.sap-press.com provides detailed and first-hand information on our current publishing program. Here, you can also easily order all of our books and e-books. For information on Galileo Press Inc. and for additional contact options please refer to our company website: http://www.galileo-press.com.

Legal Notes

This section contains the detailed and legally binding usage conditions for this e-book.

Copyright Note

Your Rights as a User

Digital Watermark

Trademarks

The common names, trade names, descriptions of goods, and so on used in this publication may be trademarks without special identification and subject to legal regulations as such.

Limitation of Liability

Regardless of the care that has been taken in creating texts, figures, and programs, neither the publisher nor the author, editor, or translator assume any legal responsibility or any liability for possible errors and their consequences.