ORACLE
JD EDWARDS ENTERPRISEONE

EnterpriseOne Supply Chain Business Modeler 8.12.1 Implementation Guide

May 2007

ORACLE

EnterpriseOne Supply Chain Business Modeler 8.12.1 Implementation Guide
SKU E1_APPS8121SBM-B 0507

Contents

About This Documentation Preface

JD Edwards EnterpriseOne implementation guides provide you with the information that you need to implement and use JD Edwards EnterpriseOne applications from Oracle.

This preface discusses:

- JD Edwards EnterpriseOne application prerequisites.
- Application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common fields in implementation guides.

Note. Implementation guides document only elements, such as fields and check boxes, that require additional explanation. If an element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common fields for the section, chapter, implementation guide, or product line. Fields that are common to all JD Edwards EnterpriseOne applications are defined in this preface.

JD Edwards EnterpriseOne Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use JD Edwards EnterpriseOne applications.

You might also want to complete at least one introductory training course, if applicable.

You should be familiar with navigating the system and adding, updating, and deleting information by using JD Edwards EnterpriseOne menus, forms, or windows. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your JD Edwards EnterpriseOne applications most effectively.

Application Fundamentals

Each application implementation guide provides implementation and processing information for your JD Edwards EnterpriseOne applications.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals implementation guide. Most product lines have a version of the application fundamentals implementation guide. The preface of each implementation guide identifies the application fundamentals implementation guides that are associated with that implementation guide.

The application fundamentals implementation guide consists of important topics that apply to many or all JD Edwards EnterpriseOne applications. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals implementation guides. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Download and order printed documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on Oracle's PeopleSoft Customer Connection website. Through the Documentation section of Oracle's PeopleSoft Customer Connection, you can download files to add to your Implementation Guides Library. You'll find a variety of useful and timely materials, including updates to the full line of JD Edwards EnterpriseOne documentation that is delivered on your implementation guides CD-ROM.

Important! Before you upgrade, you must check Oracle's PeopleSoft Customer Connection for updates to the upgrade instructions. Oracle continually posts updates as the upgrade process is refined.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Downloading and Ordering Printed Documentation

In addition to the complete line of documentation that is delivered on your implementation guide CD-ROM, Oracle makes JD Edwards EnterpriseOne documentation available to you via Oracle's website. You can:

- Download PDF files.
- Order printed, bound volumes.

Downloading PDF Files

You can download PDF versions of JD Edwards EnterpriseOne documentation online via the Oracle Technology Network. Oracle makes these PDF files available online for each major release shortly after the software is shipped.

See Oracle Technology Network, http://www.oracle.com/technology/documentation/psftent.html.

Ordering Printed, Bound Volumes

You can order printed, bound volumes of selected documentation via the Oracle Store.

See Oracle Store, http://oraclestore.oracle.com/OA_HTML/ibeCCtpSctDspRte.jsp?section=14021

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs)	Implement, Optimize + Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
Implementation guides support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in implementation guides:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and JD Edwards EnterpriseOne or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).

Typographical Convention or Visual Cue	Description
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Visual Cues

Implementation guides contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the JD Edwards EnterpriseOne system.

Note. Example of a note.

If the note is preceded by *Important!,* the note is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Cross-References

Implementation guides provide cross-references either under the heading "See Also" or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: "(FRA) Hiring an Employee"

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in implementation guides:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in implementation guides:

- USF (U.S. Federal)
- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about implementation guides and other Oracle reference and training materials. Please send your suggestions to your product line documentation manager at Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. Or email us at appsdoc@us.oracle.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

Common Fields Used in Implementation Guides

As of Date	The last date for which a report or process includes data.
Business Unit	An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Enter up to 30 characters of text.
Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.

Once, Always, and **Don't Run**	Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Request ID	An ID that represents a set of selection criteria for a report or process.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
SetID	An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.
Short Description	Enter up to 15 characters of text.
User ID	An ID that represents the person who generates a transaction.

EnterpriseOne Supply Chain Business Modeler Preface

This preface discusses:

- Products discussed in this implementation guide.
- Related documentation.

Note. This implementation guide documents only page elements that require additional explanation. If a page element is not documented with the process or task in which it is used, then it either requires no additional explanation or is documented with the common elements for the section, chapter, or implementation guide.

Products Discussed in this Implementation Guide

This implementation guide refers to these products:

- EnterpriseOne Supply Chain Business Modeler.
- Enterprise Supply Chain Management.
- EnterpriseOne Supply Chain Management.
- World Manufacturing Management.
- EnterpriseOne Demand Forecasting.
- EnterpriseOne Demand Consensus.
- EnterpriseOne Strategic Network Optimization.
- EnterpriseOne Production and Distribution Planning.
- EnterpriseOne Production Scheduling.
- EnterpriseOne Production Scheduling-Process.
- EnterpriseOne Sales and Operations Planning.
- EnterpriseOne Order Promising.

Companion Documentation

The Supply Chain Business Modeler implementation guide provides you with information about how to implement and use your EnterpriseOne Supply Chain Business Modeler system. However, additional essential information describing deployment and supplemental third-party software options resides in the EnterpriseOne Supply Chain Planning Hardware and Software Requirements Guide. You should be familiar with the contents of this guide.

CHAPTER 1

Getting Started

This chapter provides an overview of Supply Chain Business Modeler and discusses:

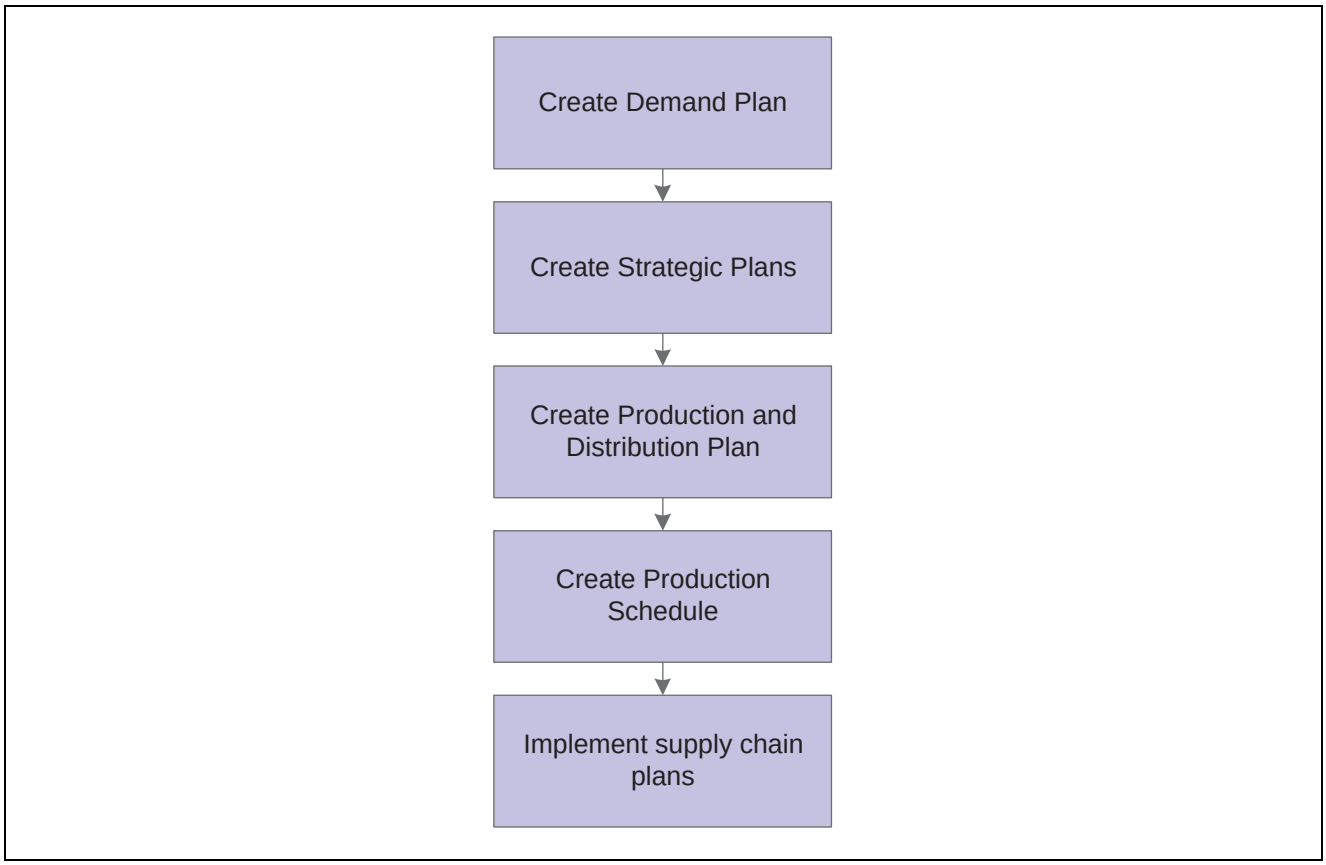
- Supply Chain Business Modeler business processes.
- Supply Chain Business Modeler integrations.
- Supply Chain Business Modeler implementation.

Supply Chain Business Modeler Overview

Supply Chain Business Modeler is a configurable supply chain data warehouse that enables you to marshall and convert data to the representation required by EnterpriseOne Supply Chain Management and Supply Chain Planning systems. With Supply Chain Business Modeler, you can easily transfer data from a supply chain management system to one or more EnterpriseOne Supply Chain Planning systems. You can then create optimized supply chain plans and transfer the plans to other Supply Chain Planning systems for further refinement. You can also transfer plans to supply chain management systems for implementation.

Supply Chain Business Modeler Business Processes

By enabling you to transfer data between EnterpriseOne Supply Chain Manufacturing, EnterpriseOne Supply Chain Planning, and other systems, Supply Chain Business Modeler supports many business processes, including the planning phase of the EnterpriseOne production business process. The following process flow illustrates this business process:



Planning phase of the Production business process

Supply Chain Business Modeler also supports the Planning and Performance Management, Marketing, Order Fulfillment, and other business processes.

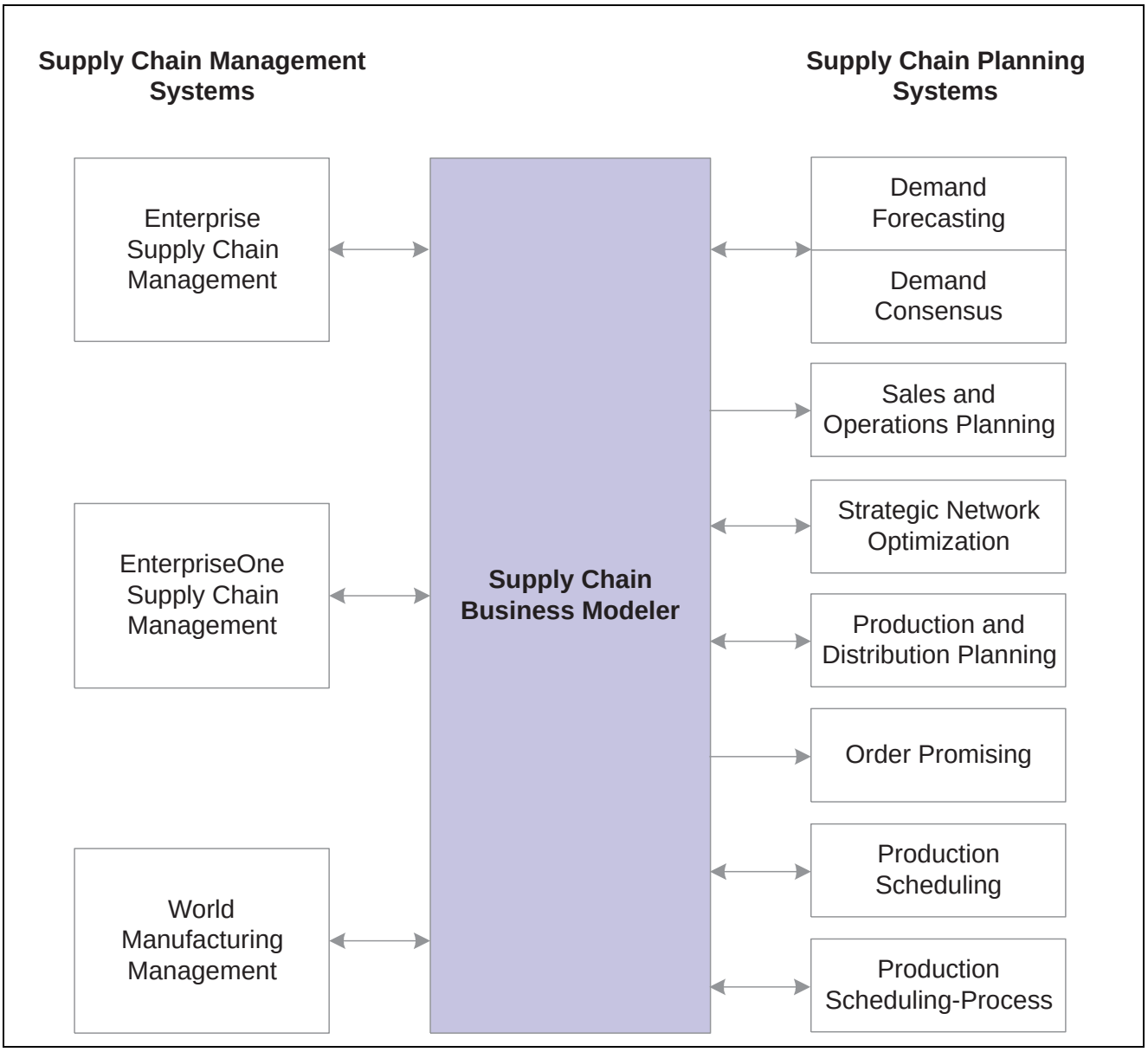
See Also

Understanding Supply Chain Business Modeler

Supply Chain Business Modeler Integrations

Supply Chain Business Modeler is used to integrate EnterpriseOne Supply Chain Management and EnterpriseOne Supply Chain Planning systems.

The following graphic shows the EnterpriseOne Supply Chain Management and Supply Chain Planning Systems that are integrated using Supply Chain Business Modeler:



Supply Chain Management Systems

Supply Chain Management systems, such as Enterprise Supply Chain Management, EnterpriseOne Supply Chain Management, and World Manufacturing Management, provide Supply Chain Business Modeler with the supply chain data that Supply Chain Planning systems need to create optimal supply chain plans. The data includes information about items, branches, inventory policies, and manufacturing processes.

After EnterpriseOne Supply Chain Planning systems create optimal supply chain plans using the data, you can transfer the supply chain plans through Supply Chain Business Modeler to a Supply Chain Management system for further refinement or implementation.

Demand Forecasting and Demand Consensus

Using Supply Chain Business Modeler, you can transfer supply chain data from EnterpriseOne Supply Chain Management systems into Demand Consensus and Demand Forecasting. You can then use Demand Consensus and Demand Forecasting to create enterprise demand forecasts based on the data.

After creating supply chain plans in Demand Forecasting and Demand Consensus, you can use Supply Chain Business Modeler to transfer the plans to another EnterpriseOne Supply Chain Planning system for further refinement or to a Supply Chain Management system for further refinement or implementation.

Strategic Network Optimization

Using Supply Chain Business Modeler, you can transfer supply chain data from Supply Chain Management systems and demand forecasts from Demand Forecasting and Demand Consensus into Strategic Network Optimization. You can then use Strategic Network Optimization to create inventory build targets and sourcing recommendations based on the data.

After creating supply chain plans in Strategic Network Optimization, you can use Supply Chain Business Modeler to transfer the plans to another EnterpriseOne Supply Chain Planning system for further refinement or to a Supply Chain Management system for further refinement or implementation.

Production and Distribution Planning

Using Supply Chain Business Modeler, you can transfer supply chain data from Supply Chain Management systems and supply chain plans from Demand Forecasting, Demand Consensus, and Strategic Network Optimization into Production and Distribution Planning. You can then use Production and Distribution Planning to create net deployment requirements and net production requirements based on the data.

After creating supply chain plans in Production and Distribution Planning, you can use Supply Chain Business Modeler to transfer the plans to another EnterpriseOne Supply Chain Planning system for further refinement or to a Supply Chain Management system for further refinement or implementation.

Production Scheduling and Production Scheduling-Process

Using Supply Chain Business Modeler, you can transfer supply chain data from Supply Chain Management systems and supply chain plans from Demand Forecasting, Demand Consensus, Strategic Network Optimization, and Production and Distribution Planning into Production Scheduling and Production Scheduling - Process. The EnterpriseOne Supply Chain Planning production scheduling applications can then produce optimal production schedules for meeting the production targets.

Sales and Operations Planning

Using Supply Chain Business Modeler, you can transfer supply chain data from Demand Forecasting, Demand Consensus, and Strategic Network Optimization to Sales and Operations Planning.

Sales and Operations Planning uses data from Supply Chain Business Modeler, but does not provide optimized plans for refinement by other applications.

Order Promising

Using Supply Chain Business Modeler, you can transfer supply chain data from a Supply Chain Management system to create a model in Order Promising.

Order Promising uses data from Supply Chain Business Modeler, but does not provide optimized plans for refinement by other applications.

Supply Chain Business Modeler Implementation

The Supply Chain Business Modeler implementation process can be divided into the following steps:

- Installing Supply Chain Business Modeler.
- Creating extract areas and data folders.
- Importing supply chain data into the Base model.
- Setting up models.
- Creating calendar and changeover data.
- Setting up import scenarios, export scenarios, and model generation scenarios.
- Automating business processes.

Installing Supply Chain Business Modeler

Step	Reference
Install Supply Chain Business Modeler.	EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for Windows, EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for UNIX, or EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for Linux.

Creating Extract Areas and Data Folders

Step	Reference
Create extract areas.	"Managing Extract Areas and Data Folders," Creating, Modifying, and Deleting Extract Areas.
Create data folders.	"Managing Extract Areas and Data Folders," Creating, Renaming, and Deleting Data Folders.

Importing Supply Chain Data into the Base Model

Step	Reference
Prepare data for importing into Supply Chain Business Modeler.	"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler."
Set up how operation and routing names are displayed, if desired.	"Importing Data into Supply Chain Business Modeler," Setting Operation and Routing Display Names.
Create import scenario for importing data into the Base model.	"Importing Data into Supply Chain Business Modeler," Creating, Modifying, and Deleting Import Scenarios.
Run import scenario for importing data into the Base model.	"Importing Data into Supply Chain Business Modeler," Running Import Scenarios.

Setting Up Models

Step	Reference
Create planning models.	"Setting Up Models," Creating, Modifying, and Deleting Planning Models.
Set model properties.	"Setting Up Models," Setting Model Properties.

Creating Calendar and Changeover Data

Step	Reference
Create and publish calendars that represent resource availability.	"Creating Calendars that Model Resource Availability."
Create and publish changeover patterns.	"Creating Changeover Patterns and Groups."

Setting Up Import, Export, and Model Generation Scenarios

Step	Reference
Create import scenarios to support business processes.	"Importing Data Into Supply Chain Business Modeler," Creating, Modifying, and Deleting Import Scenarios.
Create export scenarios to support business processes.	"Exporting Data From Supply Chain Business Modeler," Creating, Modifying, and Deleting Export Scenarios.
Create model generation scenarios to support business processes.	"Creating Supply Chain Planning Application Models," Creating, Modifying, and Deleting Model Generation Scenarios.

Automating Business Processes

Step	Reference
Create scripts for automating business processes.	“Using the Business Modeler Shell,” Automating Processes Using Scripts.

CHAPTER 2

Understanding Supply Chain Business Modeler

This chapter discusses:

- Business processes supported by Supply Chain Business Modeler.
- Supply Chain Business Modeler integration systems.

Business Processes Supported by Supply Chain Business Modeler

Supply Chain Business Modeler supports many business processes, including these four processes:

- The Planning and Performance Management business process.
- The Marketing business process.
- The Production business process.
- The Order Fulfillment business process.

The Planning and Performance Management Business Process

The planning and performance management business process involves forecasting, planning, and adjusting business strategies in order to create operational business plans and budgets.

EnterpriseOne Supply Chain Planning systems support this business process by enabling you to forecast demand and plan how to meet the demand in a supply chain. For example, Demand Forecasting and Demand Consensus enable you to create accurate demand forecasts. Strategic Network Optimization enables you to create "what-if" analyses, and plan for meeting changes in supply, demand, and capacity.

Supply Chain Business Modeler supports this business process by transferring data between Supply Chain Planning and Supply Chain Management systems. To create accurate demand forecasts and business plans, Supply Chain Planning systems require data from Supply Chain Management and other systems. Supply Chain Business Modeler enables you to transfer supply chain data between systems as often as is required to support the planning and performance management business process. In addition, Supply Chain Business Modeler provides a standard supply chain representation in which data from multiple supply chain systems can be merged in a central data warehouse.

The Marketing Business Process

The marketing business process involves planning, creating, executing, and measuring marketing activities.

EnterpriseOne Supply Chain Planning systems support the marketing business process by enabling you to create "what-if" scenarios for analyzing the potential effects of marketing campaigns in a supply chain. For example, Demand Consensus enables you to create demand forecasts that show the effects of promotions and sales campaigns. Strategic Network Optimization enables you to model and analyze the effects of promotions on supply chain production and distribution plans.

Supply Chain Business Modeler supports this business process by enabling you to transfer data easily between supply chain systems. To model the effects of marketing initiatives in a supply chain, Supply Chain Planning systems require data from Supply Chain Management and other systems. Supply Chain Business Modeler also enables you to pass plans and forecasts to lower level Supply Chain Planning systems so that you can determine how to meet increased demand in a supply chain due to marketing campaigns.

The Production Business Process

The production business process involves planning, deploying, producing, and assessing production in an organization. Supply Chain Business Modeler and Supply Chain Planning systems support the planning phase of the Production business process.

Using advanced algorithmic technology, EnterpriseOne Supply Chain Planning systems solve problems and create optimal supply chain plans at particular planning levels. For example, Strategic Network Optimization creates strategic plans for entire enterprises, while Production Scheduling creates detailed production schedules for individual plants. These supply chain plans can reduce costs, increase efficiencies, and increase profitability for organizations.

To create optimal supply chain plans, EnterpriseOne Supply Chain Planning systems require accurate, up-to-date supply chain data, including information about items, branches, customers, and demand. Furthermore, each Supply Chain Planning system requires data at a particular level of abstraction and granularity. For example, a high-level demand planning system such as Demand Forecasting requires data with less detail and a longer horizon than an operation planning system such as Production Scheduling.

Supply Chain Business Modeler supports the production business process by transforming data from Supply Chain Management systems to the representation and level of abstraction required by EnterpriseOne Supply Chain Planning systems. After creating optimized plans based on the supply chain data, you can use Supply Chain Business Modeler to convert the plan data to the representation required by a Supply Chain Management system, and transfer the optimized plans to the Supply Chain Management system for further refinement and implementation.

Furthermore, Supply Chain Business Modeler can transform optimized plans from one Supply Chain Planning system to the representation required by other Supply Chain Planning systems. As a result, you can create long-term plans in one Supply Chain Planning system and then refine the plans in other Supply Chain Planning systems to create detailed, short-term plans. For example, using Supply Chain Business Modeler and Supply Chain Planning products, you can:

1. Transfer supply chain data from a Supply Chain Management system into Demand Forecasting and Demand Consensus.
2. Use Demand Forecasting and Demand Consensus to create long-range demand forecasts by location.
3. Transfer the demand forecasts from Demand Forecasting and Demand Consensus into Strategic Network Optimization.
4. Use Strategic Network Optimization to determine the quantities of items to produce at particular branches by specific dates.
5. Transfer the inventory build targets from Strategic Network Optimization into Production and Distribution Planning.
6. Use Production and Distribution Planning to determine where and when to create the required items.

7. Transfer the supply chain plans from Production and Distribution Planning into Production Scheduling.
8. Use Production Scheduling to create detailed production schedules for meeting the production requirements.

Using Supply Chain Business Modeler, you can then transfer the optimal production schedules to a Supply Chain Management system for further refinement and implementation.

The Order Fulfillment Business Process

The order fulfillment business process involves capturing, fulfilling, and settling goods sold. Order Promising supports this business process by determining accurate and profitable dates for fulfilling customer sales orders. By promising sales orders during the order entry process, Order Promising can increase an organization's customer service levels and revenues.

To make accurate delivery commitments during the order entry process, Order Promising requires current information about on-hand inventory, planned receipts, and production capacity in a supply chain. Supply Chain Business Modeler supports this process by transforming data from a Supply Chain Management system to the representation required by Order Promising. Based on this data, Order Promising can determine accurate available-to-promise and capable-to-promise dates in real-time during the order entry process.

Supply Chain Business Modeler Integration Systems

This section discusses:

- Integrating systems using Supply Chain Business Modeler.
- Creating data in Supply Chain Business Modeler.
- Automating business processes using the Business Modeler Shell.
- Customizing Supply Chain Business Modeler integration systems.
- Integrating systems in global enterprises using Supply Chain Business Modeler.
- Maximizing the performance of integrated supply chain systems

Integrating Systems Using Supply Chain Business Modeler

Supply Chain Business Modeler makes it easy to integrate supply chain systems by:

- Importing data from supply chain systems into Supply Chain Business Modeler.
- Transforming data to the representation required by other supply chain systems.
- Exporting the data for use in other supply chain systems.

Data that you import data into Supply Chain Business Modeler must be in Supply Chain Planning Extensible Markup Language (XML) format. In Supply Chain Planning XML format, data is defined using tags and is divided into separate documents or packages. Each package includes related data that must be transferred together to ensure the integrity and reliability of the data. You can create data packages in Supply Chain Planning XML format by:

- Extracting data from Supply Chain Management systems in Supply Chain Planning XML format.
- Using EnterpriseOne Supply Chain Planning Connectors to convert data from EnterpriseOne Supply Chain Planning systems to Supply Chain Planning XML format.

- Extracting data from systems and converting the data into Supply Chain Planning XML format using Extract, Transform and Load (ETL) tools.

After you import data into Supply Chain Business Modeler, you can transform the data to the representation required by other supply chain planning and management systems. Each Supply Chain Management and Supply Chain Planning system requires data at a different level of detail. For example, a high-level planning system such as Demand Forecasting requires data with less detail and a longer horizon than an operation planning system such as Production Scheduling.

To transform the data to the required representation, you can populate the appropriate model in Supply Chain Business Modeler with the data. Models are subsets of predefined data objects that represent a supply chain at the level of detail that is required by particular Supply Chain Management or EnterpriseOne Supply Chain Planning systems.

The following table describes the five levels of models in the Supply Chain Business Modeler database, and their associated supply chain systems:

Supply Chain Business Modeler Model Level	Model Level Description	Associated Supply Chain Systems
Demand	Contains data required for creating demand forecasts. Data is saved in day, week, month, calendar quarter, or year time periods.	• Demand Consensus • Demand Forecasting
Strategic	Contains data required for creating sourcing guidelines, capacity requirements, and inventory build targets. Data is saved in day, week, month, calendar quarter, or year time periods.	Strategic Network Optimization
Tactical	Contains data required for creating net production requirements, purchase order recommendations, and deployment plans. Data is saved in daily time periods.	• Production and Distribution Planning • Order Promising • Tactical Network Optimization
Operation	Represents supply chain at the high level of granularity that is required for creating detailed production schedules. Data for individual supply chain events is saved in the Operation model. The data is event-based and is not summarized in time periods.	• Production Scheduling • Production Scheduling - Process
Base	Represents supply chain at the high level of granularity that is used in a supply chain management system. Data for individual supply chain events is saved in the Base model. The data is event-based and is not summarized in time periods.	Supply Chain Management systems: • Enterprise Supply Chain Management • EnterpriseOne Supply Chain Management • World Manufacturing Management

To populate a model with data, you can do the following:

- Refresh data in the model.

  When you refresh data in a *reference* model, the system populates the model with data from a connected model, including the snapshot date of the data in the source model. This process can involve detecting which data has changed in the model and aggregating data, depending on the source and destination models.

  When you refresh data in a *planning* model, the system populates the model with data from the reference model at the same level in the model hierarchy.

- Publish data to the model.

  When you publish plans from a *reference* model, the system populates the connected reference model that is lower in the model hierarchy with the optimized plan data. Any existing optimized plan data is cleared from the destination model.

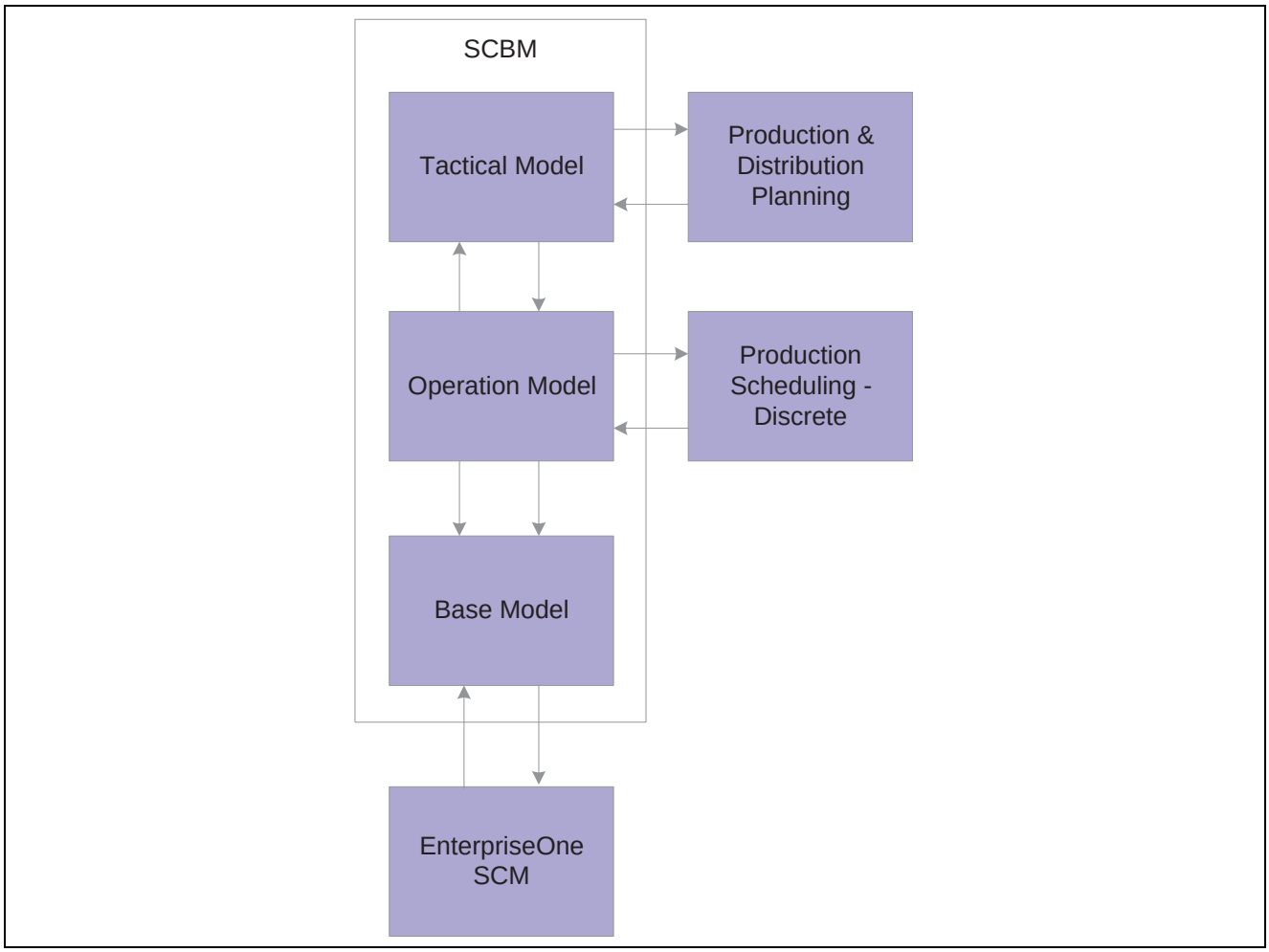
  When you publish plans from a *planning* model, the system populates the reference model at the same model level with optimized plan data from the planning model. In the destination reference model, the system clears any optimized plan data for branches that are assigned to the planning model. Data for other branches is not cleared from the reference model.

When you refresh data in a model or publish plans to a model, the system populates only the destination model with data; the data does not move automatically to any other models. For example, if you publish data from the Tactical model, it is populated only in the Operation model. To transfer data between models that are not directly connected, you must populate each connected model between the two models with the data.

After you populate the appropriate model with data, you can export the data for use in an associated supply chain planning or management system. For example, you can:

1. Import data into the Base model from a supply chain management system.
2. Refresh data in the Operation and Tactical model levels, and export data from the Tactical model level for use in Production and Distribution Planning.
3. In Production and Distribution Planning, create recommendations of cost-optimal inventory deployments and net production requirements for meeting demand.
4. Import plans from Production and Distribution Planning into the Tactical model.
5. Publish the plans to the Operation model level and export the data for use in Production Scheduling.
6. In Production Scheduling, create optimal production schedules for producing recommended inventory.
7. Import schedules from Production Scheduling into the Operation model level.
8. Publish the data to the Base model and export the data for use in a Supply Chain Management system.

The following diagram illustrates the flow of data in this example:



Data flow

See Also

"Setting Up Models," Understanding Reference and Planning Models

Creating Data in Supply Chain Business Modeler

In addition to importing data into Supply Chain Business Modeler, you can use Supply Chain Business Modeler editors to create supply chain data that is used by EnterpriseOne Supply Chain Planning systems but is not available in Supply Chain Management systems.

The Supply Chain Business Modeler calendar editor enables you to create calendars that represent the availability of resources in a supply chain. For example, you can model the downtime that is associated with performing scheduled maintenance on a machine, or the premium time that is associated with paying overtime wages for a shift. You can then publish the resulting schedule of events to Supply Chain Business Modeler models for exporting to external systems.

The Supply Chain Business Modeler changeover editor enables you to specify the time and cost associated with changeovers. For example, you can specify the time and cost involved in changing a paint sprayer from spraying red paint to spraying green paint. The changeover editor in Supply Chain Business Modeler is similar to the changeover editor in Production Scheduling. However, having a changeover editor in Supply Chain Business Modeler provides a central location for defining changeover rules that apply to an entire enterprise.

Supply Chain Business Modeler also includes a group editor, which enables you to create resource and operation groups and then create changeover patterns that apply to groups of operations or resources rather than to individual resources or groups. For example, instead of specifying the time and cost involved in changing a specific paint sprayer from spraying red paint to spraying green paint, you can specify the time and cost required to change any paint sprayer at a branch from spraying dark shades of paint to spraying lighter shades of paint. Groups that you create in the group editor are currently only used for creating changeover patterns in Supply Chain Business Modeler.

After creating calendars, changeover patterns, and groups in Supply Chain Business Modeler, you can publish data from the editors to the Base model and transform it to the representations required by specific supply chain systems.

See Also

"Creating Calendars that Model Resource Availability"

"Creating Changeover Patterns and Groups"

Automating Business Processes Using the Business Modeler Shell

Supply Chain Business Modeler includes both a graphical user interface and a command shell called the Business Modeler Shell. While the graphical user interface makes it easy to set up Supply Chain Business Modeler integration systems, the Business Modeler Shell enables you to automate business processes.

The Business Modeler Shell is an extension of the Tool Command Language (Tcl) shell, and enables you to run Supply Chain Business Modeler processes from a command prompt. You can also run other Tcl commands from the Business Modeler Shell.

To automate business processes, you can create and run scripts of Business Modeler Shell and other commands. For example, you can create a script that transfers data from a Supply Chain Management system to a Supply Chain Planning system, creates supply chain plans in the Supply Chain Planning system, and transfers supply chain plans to the Supply Chain Management system for implementation. You can then schedule the script to run automatically, as often as required to support the Production business process. Similarly, you can create and schedule scripts that transfer data from a Supply Chain Management system to EnterpriseOne Order Promising as often as required to support the Order Fulfillment business process.

See Also

"Using the Business Modeler Shell"

"Business Modeler Commands"

Customizing Supply Chain Business Modeler Integration Systems

Supply Chain Business Modeler and EnterpriseOne Supply Chain Planning Connectors provide a standard, upgradeable integration solution for a core set of data that is available in almost all Supply Chain Management and Supply Chain Planning implementations. This standard integration solution makes it easy for you to implement and upgrade Supply Chain Business Modeler integration systems.

You can also create and run model generation scenarios in Supply Chain Business Modeler to specify how Supply Chain Planning applications use data from Supply Chain Business Modeler. When you create a model generation scenario, you can select options for displaying and using the data in a particular Supply Chain Planning product. You can then run the model generation scenario to create a model generation rules (.mgr) file. The .mgr file specifies how the Supply Chain Planning product should display or use the data, and can be called when you run a connector to refresh data in a Supply Chain Planning product.

To further address the unique aspects of Supply Chain Management and Supply Chain Planning implementations, you can use Extract, Transform, and Load (ETL) tools. Using ETL tools, you can extract data from supply chain systems, and convert the data to Supply Chain Planning XML format. You can then merge these values with existing data in Supply Chain Business Modeler models. For example, transportation details such as carriers and lead-time are defined in the core transportation module in some Supply Chain Management implementations, and in the advanced transportation module in other systems. Using ETL tools, you can ensure that the correct data is available, regardless of the data location in Supply Chain Management systems.

Merging new data with existing data in Supply Chain Business Modeler is also useful because Supply Chain Management systems do not contain all data required by Supply Chain Planning systems. Because Supply Chain Management systems focus on business process execution while Supply Chain Planning systems focus on planning for the future, these systems do not have exactly the same data model. However, you can use ETL tools to create data packages with missing data values, and then merge the data with existing model data.

Integrating Systems in Global Enterprises Using Supply Chain Business Modeler

Supply Chain Business Modeler is used as a conduit to pass data to and from various applications at different planning levels, which could span multiple locations within a global enterprise.

Because locations in a global enterprises can have different languages, it is imperative for Supply Chain Business Modeler to handle data in languages other than English. Supply Chain Business Modeler internally represents data using the Unicode standard, a character coding system that supports diverse languages. Supply Chain Business Modeler can import data with encoding formats that are supported by the Xerces library, including utf-8, utf-16, latin-1, and ASCII. Supply Chain Business Modeler exports data in Supply Chain Planning XML format using utf-8 encoding. Numbers and dates must be in standard XML schema data types. In the graphical user interface (GUI) and in Business Modeler Shell Tcl commands, you can include international characters in the names of elements such as calendars, changeover patterns, groups, planning models, and import and export scenarios. File names should be in ASCII format, or another format that can be easily used in the locale of the operating system.

Locations in global enterprises can also have different Supply Chain Management systems. When you use data from multiple systems in Supply Chain Business Modeler and Supply Chain Planning applications, you must ensure that the data presents a reliable, accurate view of the enterprise. To do so, you must accommodate differences such as the following:

- Time zone.

  To accommodate time zone differences, you might have to convert data from different time zones to a standard time zone for an enterprise.

- Date formats.

  Storing dates in standard ISO format can be useful so you do not have to transform dates from location-specific formats.

- Different systems might use the same codes to represent different items or resources.

For example, different Supply Chain Management systems might use the same short item number to refer to different items. To solve this problem, you could select an alternate item number to use or change item codes from one system to be unique by adding a character to the front of each item number from one system. To perform global planning, you could use an ETL tool to convert multiple codes for the same item to the same code.

- Currencies.
- Business processes.

  If you create plans for locations with separate Supply Chain Management systems, you must be careful to send the resulting plans to the correct Supply Chain Management system for implementation. For example, if you create net production requirements in Production and Distribution Planning for an entire enterprise, you must be careful to direct the plans to the appropriate Supply Chain Management systems, especially if the plans span locations. For example, a transportation plan could have a source location in one Supply Chain Management system and a destination location in another system.

See Also

XML Schema Part 2: Datatypes Second Edition, http://www.w3.org/TR/xmlschema-2

Maximizing the Performance of Integrated Supply Chain Systems

EnterpriseOne Supply Chain Planning 8.12.1 systems, including Supply Chain Business Modeler, can export and import data in compressed .gzip format. This compressed format reduces the size of data files that are transferred between systems, and reduces the time required for transferring data between systems.

To further reduce the time required for transferring data between systems, you can reduce the amount of data that you extract and transfer using the following guidelines:

- Only extract data from a Supply Chain Management system that is required to support business planning processes.

  For example, if you are only using EnterpriseOne Production Scheduling to create production schedules, you do not need to extract sales order history data. Similarly, if you do not extract data for expired operations, you will not spend additional time transferring unnecessary data between systems.
- Run multiple extracts from a Supply Chain Management system to create relatively small data files, and use incremental imports to add the data into Supply Chain Business Modeler.
- Only extract transactional data that is consistent with enterprise data extracts.

  Supply Chain Business Modeler does not import transactional data for items that are not defined in the Base package in a model. If you extract transactional data for items that are not defined in the Base package, you will spend time transferring data that cannot be used.
- Extract static data less frequently than dynamic data.

  For example, you could extract static data on weekends and dynamic data every day. Alternatively, you could extract static and dynamic data every day, except for customer data. Customer data is usually quite stable and very large.

CHAPTER 3

Understanding Data for Importing into and Exporting from Supply Chain Business Modeler

This chapter discusses:

- Supply Chain Planning XML format.
- Supply Chain Planning XML data packages.
- Compressed Supply Chain Planning XML files.
- Supply Chain Planning XML format versions.
- XML Schema Definitions.
- Sample data packages.

Supply Chain Planning XML Format

Data is imported into and exported from Supply Chain Business Modeler in Supply Chain Planning XML format. Supply Chain Planning XML format is designed for integrating supply chain products and is based on the Extensible Markup Language (XML) version 1.0 standard that is officially recommended by the World Wide Web Consortium as of 1998. Unlike flat file data that uses tabs or other characters as content delimiters, data in XML format uses tags to define the data.

Data in Supply Chain Planning XML format is divided into separate XML documents, or packages. Each package includes related XML data that must be imported into or exported from Supply Chain Business Modeler together to ensure the integrity and reliability of the data. For example, the Supplier package contains information about suppliers, groups of suppliers, and the length of time necessary to obtain items from suppliers. Data in Supply Chain Business Modeler models is also grouped into these packages.

There are three categories of data packages in Supply Chain Planning XML:

- Enterprise data.

  Enterprise data packages include information about the supply chain model structure. This data does not change frequently, and is traditionally called static data.

- Transactional data.

  Transactional data packages include information about supply and demand in a supply chain. This data changes frequently, and is traditionally called dynamic data.

- Optimized plan data.

  Optimized plan packages usually include data that originates from EnterpriseOne Supply Chain Planning products. You can send this data through Supply Chain Business Modeler to a supply chain management system for action or to other Supply Chain Planning products for further refinement.

Within each package are elements and attributes that contain data. Elements and attributes in XML documents are similar to tables and fields in relational databases. When you import a Supply Chain Planning XML package, Supply Chain Business Modeler populates the corresponding package of objects in the destination model with data from elements and attributes in the Supply Chain Planning XML data package.

Supply Chain Business Modeler exports all data in Supply Chain Planning XML format. You can produce data in Supply Chain Planning for importing into Supply Chain Business Modeler using the following methods:

- Creating data extracts in Supply Chain Planning XML format from Enterprise Supply Chain Management, EnterpriseOne Supply Chain Management, or World Manufacturing Management.
- Creating data extracts in Supply Chain Planning XML format using EnterpriseOne Supply Chain Planning application connectors.
- Extracting data from systems and transforming the data using Extract, Transform, and Load (ETL) tools.

See Also

XML Schema Definitions

Supply Chain Planning XML Format Versions

Supply Chain Business Modeler can import data in four Supply Chain Planning XML format versions:

Supply Chain Planning XML 3.3 format	Includes data package changes that correspond to Supply Chain Business Modeler 8.12.1 data model changes.
Supply Chain Planning XML 3.2 format	Includes data package changes that correspond to Supply Chain Business Modeler 8.12 and 8.11.1 data model changes.
Supply Chain Planning XML 3.1 format	Includes data package changes that correspond to Supply Chain Business Modeler 8.11 data model changes.
Supply Chain Planning XML 3.0 format	Introduced with Supply Chain Business Modeler 8.10. In data packages in Supply Chain Planning 3.0 format, detailed information is nested within its related header data.
Supply Chain Planning XML 2.0 format	Introduced with Supply Chain Business Modeler 2.0. In data packages in Supply Chain Planning XML 2.0 format, general information appears in separate Supply Chain Planning XML document sections from detailed information.

Supply Chain Planning XML 3.3 Features

In addition to features that are available in Supply Chain Planning XML 3.2, 3.1 and 3.0, Supply Chain Planning XML 3.3 format includes changes to the Base and Strategic models to support changes to Production Scheduling, Strategic Network Optimization, and integration with the Oracle E-Business Suite. Key changes are:

- Customer Sourcing object has been removed from the Customer package and replaced with the Customer Transport Mode and Customer Item Transport Mode objects in all models.
- Currency Forecast package has been added to the Base, Operational, Tactical and Strategic models. This package includes the Currency Forecast and Currency object, and is used in strategic planning.

- Supplier Item Branch object has been added to the Suppliers package in the Base, Operation, Tactical, and Strategic models.
- Customer Product Substitution object has been added to the Customer package in the Base, Operation, Tactical, and Strategic models.
- Implementing Resource object has been added to the Manufacturing package in the OperationBor object in the Base, Operation, Tactical, and Strategic models.
- Assigned Attribute (for Items), Resource Attribute, and Attribute Value objects have been added to the Base package in the Base, Operation, and Tactical models.
- Machine Assigned Attribute object has been added in the Machine object to the Manufacturing package in the Base, Operation, and Tactical models.
- Work Order Implementing Resource and Next Work Order objects have been added to the Work Orders package in the Base, Operation, and Tactical models.
- Work Order Schedule package has been added to the Base and Operation models. This package includes Work Order Schedule, Pegged Work Order, Satisfied Demand, Work Order Schedule Operation, Work Order Schedule Bor, Planned Resource, Actual Resource, Work Order Consumed Item BOM, and Work Order Produced Item BOM objects.
- Fixed Production package has been added in the Strategic model to replace the Work Orders package.
- Work Orders package and all objects have been removed in the Strategic model.

Detailed changes to the Base, Operation and Tactical models are shown below. The models affected are referenced as follows:

- B — Base
- D — Demand
- O — Operation
- S — Strategic
- T — Tactical
- * — links back to an object

Model	Package	Object	Field	Type of Change
B, O, T	Base	Resource Attribute	attributeCode	Added.
B, O, T	Base	Resource Attribute	attributeDescription	Added.
B, O, T	Base	Resource Attribute	AttributeValueList	Added.
B, O, T	Base	Attribute Value	resourceAttribute*	Added.
B, O, T	Base	Attribute Value	attributeValueCode	Added.
B, O, T	Base	Attribute Value	attributeValueDescription	Added.
B, O, T	Base	Assigned Attribute	item*	Added.
B, O, T	Base	Assigned Attribute	attributeCode	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Base	Assigned Attribute	attributeValueCode	Added.
B, O, T	Base	Item	assignedAttributeList	Added.
B, O, T	Base	Item	relaxMaximum	Added.
B, O, T	Base	Item	relaxMinimum	Added.
B, O, T, S	Currency Forecast	Currency		New object added.
B, O, T, S	Currency Forecast	Currency	conversion	Added.
B, O, T, S	Currency Forecast	Currency	currencyCode	Added.
B, O, T, S	Currency Forecast	Currency	currencyForecast*	Added.
B, O, T, S	Currency Forecast	Currency	currencyName	Added.
B, O, T, S	Currency Forecast	Currency Forecast		New object added.
B, O, T, S	Currency Forecast	Currency Forecast	currencyForecastCode	Added.
B, O, T, S	Currency Forecast	Currency Forecast	currencyForecastName	Added.
B, O, T, S	Currency Forecast	Currency Forecast	currencyList	Added.
B, O, T, S	Currency Forecast	Currency Forecast	forecastDate	Added.
B, O, T, S	Currency Forecast	Currency Forecast	forecastSource	Added.
B, O, T, S	Customer	Customer	customerSourcingList	Removed.
All	Customer	Customer Item Transport Mode		Object Added.
All	Customer	Customer Item Transport Mode	available	Added.
All	Customer	Customer Item Transport Mode	customerTransportMode*	Added.
All	Customer	Customer Item Transport Mode	item*	Added.
All	Customer	Customer Item Transport Mode	minimumShipment	Added.
All	Customer	Customer Item Transport Mode	preference	Moved from the Customer Sourcing object.

Model	Package	Object	Field	Type of Change
All	Customer	Customer Item Transport Mode	shipMultiple	Added.
S	Customer	Customer Product Substitution		New object added.
B, O, T, S	Customer	Customer Product Substitution	available	Added.
S	Customer	Customer Product Substitution	branchCode	Added.
S	Customer	Customer Product Substitution	cost	Added.
S	Customer	Customer Product Substitution	customerCode	Added.
S	Customer	Customer Product Substitution	item	Added.
S	Customer	Customer Product Substitution	multiple	Added.
S	Customer	Customer Product Substitution	preference	Added.
S	Customer	Customer Product Substitution	ratio	Added.
S	Customer	Customer Product Substitution	substituteItem	Added.
B, O, T, S	Customer	Customer Sourcing		Object removed.
All	Customer	Customer Transport Mode		New object.
All	Customer	Customer Transport Mode	available	Added.
All	Customer	Customer Transport Mode	branch	Moved from the Customer Sourcing object.
All	Customer	Customer Transport Mode	customer	Moved from the Customer Sourcing object.
All	Customer	Customer Transport Mode	customerItemTransportModeList	Added.

Model	Package	Object	Field	Type of Change
All	Customer	Customer Transport Mode	costPerVolumeUnit	Added.
All	Customer	Customer Transport Mode	costPerWeightUnit	Added.
All	Customer	Customer Transport Mode	description	Added.
All	Customer	Customer Transport Mode	leadTime	Moved from the Customer Sourcing object.
All	Customer	Customer Transport Mode	leadTimeUom	Moved from the Customer Sourcing object.
All	Customer	Customer Transport Mode	transportModeCode	Added.
All	Customer	Customer Transport Mode	volumeCapacity	Added.
All	Customer	Customer Transport Mode	volumeCapacityUom	Added.
All	Customer	Customer Transport Mode	weightCapacity	Added.
All	Customer	Customer Transport Mode	weightCapacityUom	Added.
B, O, T	Deployment Plan	Deployment Plan Detail	transferOrderDetail	Changed from transferOrder-Code.
B, O, T	Deployment Plan	Deployment Plan Detail	lineItem	Removed.
B, O, T	Enterprise Forecast	Enterprise Forecast	forecastName	Added.
S	Fixed Production	Fixed Production		New object added.
S	Fixed Production	Fixed Production	branch*	Added.
S	Fixed Production	Fixed Production	item*	Added.
S	Fixed Production	Fixed Production	quantity	Added.
B, O, T	Manufacturing	Crew	enforceActivityStartTime	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Manufacturing	Crew	prebuildTarget	Added.
B, O, T	Manufacturing	Crew	prebuildTargetUOM	Added.
B, O, T	Manufacturing	Crew	throughputPercentage	Added.
B, O, T	Manufacturing	Crew	throughputPercentageDescription	Added.
B, O, T, S	Manufacturing	Implementing Resource		New object added.
B, O, T, S	Manufacturing	Implementing Resource	capacityRequired	Added. Was originally in Operation Bor object.
B, O, T, S	Manufacturing	Implementing Resource	operationBor	Added.
B, O, T, S	Manufacturing	Implementing Resource	required	Added.
B, O, T, S	Manufacturing	Implementing Resource	resourceCode	Added. Was originally in Operation Bor object.
B, O, T	Manufacturing	Machine	capacityType	Added.
B, O, T	Manufacturing	Machine	changeoverPrebuildMaximum	Added.
B, O, T	Manufacturing	Machine	changeoverPrebuildMaximumTimeUnit	Added.
B, O, T	Manufacturing	Machine	enforceActivityStartTime	Added.
B, O, T	Manufacturing	Machine	machineAssigned AttributeList	Added.
B, O, T	Manufacturing	Machine	minimumCapacity	Added.
B, O, T	Manufacturing	Machine	minimizeChangeoversBetweenCycles	Added.
B, O, T	Manufacturing	Machine	minimumCycleTime	Added.
B, O, T	Manufacturing	Machine	minimumCycleTimeUOM	Added.
B, O, T	Manufacturing	Machine	minimizeIdleTime	Added.
B, O, T	Manufacturing	Machine	prebuildTarget	Added.
B, O, T	Manufacturing	Machine	prebuildTargetUOM	Added.
B, O, T	Manufacturing	Machine	pullForwardWindow	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Manufacturing	Machine	pullForwardWindowTimeUnit	Added.
B, O, T	Manufacturing	Machine	schedulingObjective	Added.
B, O, T	Manufacturing	Machine	throughputPercentage	Added.
B, O, T	Manufacturing	Machine	throughputPercentageDescription	Added.
B, O, T	Manufacturing	Machine	useChangeoverPrebuildMaximum	Added.
B, O, T	Manufacturing	Machine Assigned Attribute		New object added.
B, O, T	Manufacturing	Machine Assigned Attribute	attributeCode	Added.
B, O, T	Manufacturing	Machine Assigned Attribute	attributeValueCode	Added.
B, O, T	Manufacturing	Machine Assigned Attribute	machine*	Added.
B, O, T, S	Manufacturing	Operation	operationName	Added.
B, O, T	Manufacturing	Operation	type	Added.
B, O, T, S	Manufacturing	Operation BOR	capacityRequired	Moved to implementing-Resource.
B, O, T, S	Manufacturing	Operation BOR	resourceCode	Moved to implementing-Resource.
B, O, T, S	Manufacturing	Operation BOR	durationResource	Added.
B, O, T	Manufacturing	Operation BOR	implementingResourceList*	Added.
B, O, T, S	Manufacturing	Operation BOR	isMinimumRunByTime	Added.
B, O, T, S	Manufacturing	Operation BOR	lotMultipleResource	Added.
B, O, T, S	Manufacturing	Operation BOR	resourceGroupName	Added.
B, O, T	Manufacturing	Operation Set	consistentResourceAssignmentGroupID	Added.
B, O, T, S	Manufacturing	ResourceSet	resourceSetName	Added.
B, O, T	Manufacturing	Routing	preferAdjacentOperations	Added.
B, O, T, S	Manufacturing	Routing	routingName	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Manufacturing	Routing	workOrderUnitsofEffort	Added.
B, O, T	Manufacturing	Tool	enforceActivityStartTime	Added.
B, O, T	Manufacturing	Tool	prebuildTarget	Added.
B, O, T	Manufacturing	Tool	prebuildTargetUOM	Added.
B, O, T	Manufacturing	Tool	throughputPercentage	Added.
B, O, T	Manufacturing	Tool	throughputPercentageDescription	Added.
B, O, T	Master Production Plan	Master Production Plan Detail	workOrder*	Changed from workOrderCode.
B, O, T	Master Production Plan	Master Production Plan Detail	routing*	Changed from routingCode.
B, O, T	Purchase Orders	Purchase Order	purchaseOrderName	Added.
B, O, T	Purchase Plan	Purchase Plan Detail	purchaseOrder*	Changed from purchaseOrder-Code.
B, O, T	Sales Orders	Sales Order	orderClass	Added.
B, O, T	Sales Orders	Sales Order	salesOrderName	Added.
B, O, T	Sales Orders	Sales Order Details	orderClass	Added.
B, O, T	Sales Orders	Sales Order Details	shipToCustomer*	Renamed from shipToCustomer-Code.
B, O, T	Sales Orders	Sales Order History	billToCustomer*	Changed from customerCode.
B, O, T	Supplier	Supplier	supplierItemBranchList	Added.
B, O, T, S	Supplier	Supplier Item	branch	Moved to Supplier Item Branch.
B, O, T	Supplier	Supplier Item	demandCoverageOrder	Moved to Supplier Item Branch object.
S	Supplier	Supplier Item	demandCoverageOrder	Removed.

Model	Package	Object	Field	Type of Change
B, O, T, S	Supplier	Supplier Item	demandCoverageUnits	Moved to Supplier Item Branch object.
B, O, T, S	Supplier	Supplier Item	fixedOrderQuantity	Moved to Supplier Item Branch object.
B, O, T, S	Supplier	Supplier Item	leadTimeUom	Moved to Supplier Item Branch object.
B, O, T , S	Supplier	Supplier Item	maximumQuantity	Changed from maximumPeriod-Order.
B, O, T, S	Supplier	Supplier Item	minimumQuantity	Changed from minimumPeriod-Order.
B, O, T, S	Supplier	Supplier Item	minmaxOrderPeriod	Moved to Supplier Item Branch object.
B, O, T, S	Supplier	Supplier Item	minmaxOrderPeriodUnit	Moved to Supplier Item Branch object.
B, O, T, S	Supplier	Supplier Item	orderPolicy	Moved to Supplier Item Branch.
B, O, T, S	Supplier	Supplier Item	preference	Removed.
B, O, T, S	Supplier	Supplier Item	premiumCost	Moved to Supplier Item Branch.
B, O, T, S	Supplier	Supplier Item	premiumLeadTime	Moved to Supplier Item Branch.
B, O, T	Supplier	Supplier Item	receivingCalendar	Moved to Supplier Item Branch.
S	Supplier	Supplier Item	receivingCalendar	Removed.
B, O, T	Supplier	Supplier Item	shippingCalendar	Moved to Supplier Item Branch.
S	Supplier	Supplier Item	shippingCalendar	Removed..

Model	Package	Object	Field	Type of Change
B, O, T, S	Supplier	Supplier Item	standardLeadTime	Moved to Supplier Item Branch.
B, O, T, S	Supplier	Supplier Item Branch		New object.
B, O, T, S	Supplier	Supplier Item Branch	branch*	Moved from the Supplier Item object.
B, O, T	Supplier	Supplier Item Branch	demandCoverageOrder	Added.
B, O, T, S	Supplier	Supplier Item Branch	demandCoverageUnits	Added.
B, O, T	Supplier	Supplier Item Branch	fixedOrderQuantity	Added.
B, O, T, S	Supplier	Supplier Item Branch	item*	Added.
B, O, T, S	Supplier	Supplier Item Branch	leadTimeUom	Moved from the Supplier Item object.
B, O, T, S	Supplier	Supplier Item Branch	maximumQuantity	Added.
B, O, T, S	Supplier	Supplier Item Branch	minimumQuantity	Added.
B, O, T	Supplier	Supplier Item Branch	minmaxOrderPeriod	Added.
B, O, T	Supplier	Supplier Item Branch	minmaxOrderPeriodUnit	Added.
B, O, T, S	Supplier	Supplier Item Branch	orderMultiple	Added.
B, O, T	Supplier	Supplier Item Branch	orderPolicy	Moved from the Supplier Item object.
B, O, T, S	Supplier	Supplier Item Branch	premiumCost	Moved from the Supplier Item object.
B, O, T, S	Supplier	Supplier Item Branch	premiumLeadTime	Moved from the Supplier Item object.

Model	Package	Object	Field	Type of Change
B, O, T	Supplier	Supplier Item Branch	receivingCalendar	Moved from Supplier Item object.
B, O, T	Supplier	Supplier Item Branch	shippingCalendar	Added.
B, O, T, S	Supplier	Supplier Item Branch	standardCost	Added.
B, O, T, S	Supplier	Supplier Item Branch	standardLeadTime	Moved from the Supplier Item object.
B, O, T, S	Supplier	Supplier Item Branch	supplier*	Added.
B, O, T	Transfer Orders	Transfer Order	orderClass	Added.
B, O, T	Transfer Orders	Transfer Order	transferOrderName	Added.
Operation Planning	Transfer Orders	Transfer Order In	orderClass	Added.
Operation Planning	Transfer Orders	Transfer Order In	transferOrderName	Added.
Operation Planning	Transfer Orders	Transfer Order Out	orderClass	Added.
Operation Planning	Transfer Orders	Transfer Order Out	transferOrderName	Added.
B, O, T	Work Orders	Next Work Order		New object added.
B, O, T	Work Orders	Next Work Order	maxSeparation	Added.
B, O, T	Work Orders	Next Work Order	minSeparation	Added.
B, O, T	Work Orders	Next Work Order	precedenceConstraint	Added.
B, O, T	Work Orders	Next Work Order	timeUnit	Added.
B, O, T	Work Orders	Next Work Order	workOrder*	Added.
B, O, T	Work Orders	Next Work Order	workOrderCode	Added.
B, O, T	Work Orders	Work Order	nextWorkOrderList	Added.
B, O, T	Work Orders	Work Order	offHoldDate	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Work Orders	Work Order	orderClass	Added.
B, O, T	Work Orders	Work Order	parentWorkOrderBranch*	Changed from parent-WorkOrderBranch-Code.
B, O, T	Work Orders	Work Order	parentWorkOrderCode	Changed to parent-WorkOrder.
B, O, T	Work Orders	Work Order	releaseinERP	Added.
B, O, T	Work Orders	Work Order	routing*	Changed from routingCode.
B, O, T	Work Orders	Work Order	status	New values: Released, UnReleased, Cancelled, Pending, WIP, Complete, On Hold.
B, O, T	Work Orders	Work Order	workOrderName	Added.
B, O, T	Work Orders	Work Order Bor	capacityRequired	Removed from object.
B, O, T	Work Orders	Work Order Bor	endTimeOffset	Removed from object.
B, O, T	Work Orders	Work Order Bor	leadTimeOffset	Removed from object.
B, O, T	Work Orders	Work Order Bor	resourceCode	Removed from object.
B, O, T	Work Orders	Work Order Bor	resourceType	Removed from object.
B, O, T	Work Orders	Work Order Bor	durationResource	Added.
B, O, T	Work Orders	Work Order Bor	lotMultiple	Added.
B, O, T	Work Orders	Work Order Bor	lotMultipleResource	Added.
B, O, T	Work Orders	Work Order Bor	resourceGroupName	Added.
B, O, T	Work Orders	Work Order Bor	workOrderImplementingResourceList	Added.

Model	Package	Object	Field	Type of Change
B, O, T	Work Orders	Work Order Implementing Resource		New Object added.
B, O, T	Work Orders	Work Order Implementing Resource	capacityRequired	Added.
B, O, T	Work Orders	Work Order Implementing Resource	endTimeOffset	Added.
B, O, T	Work Orders	Work Order Implementing Resource	leadTimeOffset	Added.
B, O, T	Work Orders	Work Order Implementing Resource	resourceCode	Added.
B, O, T	Work Orders	Work Order Implementing Resource	resourceType	Added.
B, O, T	Work Orders	Work Order Implementing Resource	workOrderBor*	Added.
B, O, T	Work Orders	Work Order Consumed Item BOM	parentWorkOrder*	Changed from parentWorkOrder-Code
B, O, T	Work Orders	Work Order Consumed Item BOM	parentWorkOrderBranch*	Changed from parentWorkOrder-BranchCode.
B, O, T	Work Orders	Work Order Consumed Item BOM	quantityIssued	Added.
B, O, T	Work Orders	Work Order Consumed Item BOM	scaleParts	Added.
B, O, T	Work Orders	Work Order Produced Item	parentWorkOrder*	Changed from parentWorkOrder-Code.
B, O, T	Work Orders	Work Order Produced Item	parentWorkOrderBranch	Changed from parentWorkOrder-BranchCode.

Model	Package	Object	Field	Type of Change
B, O, T	Work Orders	Work Order Operation	durationResourceOrSet	Added.
B, O, T	Work Orders	Work Order Operation	firmStatus	
B, O, T	Work Orders	Work Order Operation	lotMultipleResourceOrSet	Added as a fixed attribute to support a PS enhancement.
B, O, T	Work Orders	Work Order Operation	nextOperation	Added.
B, O, T	Work Orders	Work Order Operation	operation*	Changed from operationCode.
B, O, T	Work Orders	Work Order Operation	scaleDuration	Added.
B, O	Work Order Schedule	Actual Resource		New Object.
B, O	Work Order Schedule	Actual Resource	capacityRequired	Added.
B, O	Work Order Schedule	Actual Resource	resourceCode	Added.
B, O	Work Order Schedule	Actual Resource	resourceDescription	Added.
B, O	Work Order Schedule	Actual Resource	resourceType	Added.
B, O	Work Order Schedule	Actual Resource	workOrderScheduleBor*	Added.
B, O	Work Order Schedule	Pegged Work Order		New object added.
B, O	Work Order Schedule	Pegged Work Order	branch*	Added.
B, O	Work Order Schedule	Pegged Work Order	completionDate	Added.
B, O	Work Order Schedule	Pegged Work Order	description	Added.
B, O	Work Order Schedule	Pegged Work Order	item*	Added.

Model	Package	Object	Field	Type of Change
B, O	Work Order Schedule	Pegged Work Order	peggedWorkOrderCode	Added.
B, O	Work Order Schedule	Pegged Work Order	quantity	Added.
B, O	Work Order Schedule	Pegged Work Order	requestedDate	Added.
B, O	Work Order Schedule	Pegged Work Order	routing*	Added.
B, O	Work Order Schedule	Pegged Work Order	satisfiedDemandList	Added.
B, O	Work Order Schedule	Pegged Work Order	startDate	Added.
B, O	Work Order Schedule	Pegged Work Order	workOrderSchedule*	Added.
B, O	Work Order Schedule	Pegged Work Order	workOrderScheduleOperationList	Added.
B, O	Work Order Schedule	Planned Resource		New Object.
B, O	Work Order Schedule	Planned Resource	capacityRequired	Added.
B, O	Work Order Schedule	Planned Resource	resourceCode	Added.
B, O	Work Order Schedule	Planned Resource	resourceDescription	Added.
B, O	Work Order Schedule	Planned Resource	resourceType	Added.
B, O	Work Order Schedule	Planned Resource	workOrderScheduleBor*	Added.
B, O	Work Order Schedule	Satisfied Demand		New object.
B, O	Work Order Schedule	Satisfied Demand	availableDate	Added.
B, O	Work Order Schedule	Satisfied Demand	demandCode	Added.
B, O	Work Order Schedule	Satisfied Demand	demandType	Added.

Model	Package	Object	Field	Type of Change
B, O	Work Order Schedule	Satisfied Demand	lineItem	Added.
B, O	Work Order Schedule	Satisfied Demand	peggedWorkOrder*	Added.
B, O	Work Order Schedule	Satisfied Demand	quantity	Added.
B, O	Work Order Schedule	Satisfied Demand	requestedDate	Added.
B, O	Work Order Schedule	Work Order Schedule		New object
B, O	Work Order Schedule	Work Order Schedule	createdDate	Added.
B, O	Work Order Schedule	Work Order Schedule	description	Added.
B, O	Work Order Schedule	Work Order Schedule	horizonUnits	Added.
B, O	Work Order Schedule	Work Order Schedule	workOrderScheduleName	Added.
B, O	Work Order Schedule	Work Order Schedule	peggedWorkOrderList	Added.
B, O	Work Order Schedule	Work Order Schedule	planningHorizonLength	Added.
B, O	Work Order Schedule	Work Order Schedule	planningPeriodEndDate	Added.
B, O	Work Order Schedule	Work Order Schedule	planningPeriodStartDate	Added.
B, O	Work Order Schedule	Work Order Schedule	source	Added.
B, O	Work Order Schedule	Work Order Schedule	workOrderScheduleCode	Added.
B, O	Work Order Schedule	Work Order Schedule Bor		New object added.
B, O	Work Order Schedule	Work Order Schedule Bor	actualResourceGroupName	Added.
B, O	Work Order Schedule	Work Order Schedule Bor	actualResourceList	Added.

Model	Package	Object	Field	Type of Change
B, O	Work Order Schedule	Work Order Schedule Bor	plannedResourceGroupName	Added.
B, O	Work Order Schedule	Work Order Schedule Bor	plannedResourceList	Added.
B, O	Work Order Schedule	Work Order Schedule Bor	resourceSet	Added.
B, O	Work Order Schedule	Work Order Schedule Bor	workOrderScheduleOperation*	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM		New object added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	actualItem	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	actualItemDescription	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	consumedItemSet	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	operationSetName	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	plannedItem	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	plannedItemDescription	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	quantity	Added.
B, O	Work Order Schedule	Work Order Consumed Item BOM	workOrderScheduleOperation*	Added.
B, O	Work Order Schedule	Work Order Produced Item BOM		New object added.

Model	Package	Object	Field	Type of Change
B, O	Work Order Schedule	Work Order Produced Item BOM	description	Added.
B, O	Work Order Schedule	Work Order Produced Item BOM	item*	Added.
B, O	Work Order Schedule	Work Order Produced Item BOM	operationSetName	Added.
B, O	Work Order Schedule	Work Order Produced Item BOM	quantity	Added.
B, O	Work Order Schedule	Work Order Produced Item BOM	workOrderScheduleOperation*	Added.
B, O	Work Order Schedule	Work Order Schedule Operation		New object added.
B, O	Work Order Schedule	Work Order Schedule Operation	durationUnit	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	operation*	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	operationSetName	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	peggedWorkOrder*	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	scheduledCompletionDate	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	scheduledStartDate	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	workOrderScheduleBorList	Added.

Model	Package	Object	Field	Type of Change
B, O	Work Order Schedule	Work Order Schedule Operation	workOrderScheduleConsumedItemBom	Added.
B, O	Work Order Schedule	Work Order Schedule Operation	workOrderScheduleProducedItemBom	Added.

Note. For complete Supply Chain Planning XML 3.3 format documentation, see the XML schema definitions using the Schema Viewer that ships with Supply Chain Business Modeler 8.12.1.

Supply Chain Planning XML 3.2 Features

In addition to features that are available in Supply Chain Planning XML 3.1 and 3.0, Supply Chain Planning XML 3.2 format includes the following features:

- Automated method of customizing operation and routing display names upon import.
- Resource capacity time calculations.
- Enhancements to support Strategic Network Optimization.
- Optimized plan documentation.
- Improved lead time modeling.
- Other enhancements to support changes in the Supply Chain Business Modeler data model.

To support customized operation and routing display names, particularly useful for Production Scheduling, Supply Chain Planning XML 3.2 includes the following elements:

- The Operation element in Manufacturing data package includes a new operationName element.
- The Routing element in Manufacturing data package includes a new routingName element.

When exporting capacity values for crew, machine, and tool resources from the Tactical and Strategic models, Supply Chain Business Modeler now calculates the amount of time available on the resource in uptime and in premium time in each time period rather than exporting the number of available crews, machines, or tools. To support these resource capacity calculations, Supply Chain Planning XML 3.2 includes the following elements:

- The Crew, Machine, and Tool elements in Manufacturing data packages for Tactical and Strategic level models include a new element: premiumCapacity. This element specifies the time available on the resource during premium time in the time period.
- In Supply Chain Planning XML 3.2, effective-dated capacity data for Crew, Machine, and Tool elements in Manufacturing data packages for Tactical level models is saved in daily time buckets. In Supply Chain Planning XML 3.1, effective-dated capacity data for crews, machines, and tools appears in a compressed form, in bucket elements that can span multiple model time periods.

To support Strategic Network Optimization, Supply Chain Planning XML 3.2 includes the following changes in data packages for the Strategic model:

- The Enterprise Forecast element has been removed from the Demand package in the Transactional Data category, and an Enterprise Forecast package with a header element and detail element has been added in the Optimized Plans category. Users can now transfer several forecast versions as optimized plans rather than including a single forecast as a Demand element.

- The Enterprise Forecast Detail element in the Enterprise Forecast package and in the Sales element in the Demand package now includes a Customer element.
- A Work Orders data package has been added.
- In the Purchase Orders and Transfer Orders packages, fixed shipDate and arriveDate elements have been added to the primary key, and the quantity element has been changed from a time-varying to a fixed field. In Supply Chain Planning XML 3.1, the time period start date represented the ship date of items, and the arrive date was not specified.
- In Supply Chain Planning XML 3.1, effective-dated data for the Strategic model appears in single bucket elements that can span multiple model time periods. In Supply Chain Planning XML 3.2, to make it easier to build models in Strategic Network Optimization, effective-dated data appears in bucket elements for each model time period.

  For example, this sample data shows inventory policy data in Supply Chain Planning XML 3.2 format:

```
<inventoryPolicy>
  <itemCode>9797702</itemCode>
  <branchCode>GUNNISON</branchCode>
  <forecastConsumptionRule>Greater-Forecast</forecastConsumptionRule>
  <forecastConsumptionTimeFence>4</forecastConsumptionTimeFence>
  <timeFenceUnit>Months</timeFenceUnit>
  <coverageUnit>Months</coverageUnit>
  <bucketList>
  <bucket startDate="2004-01-01T00:00:00" endDate="2004-02-01T00:00:00">
    <maximumLevel>0</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
    <maximumCoverage>0.333333333333333</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>Yes</useCoverForMaximum>
  </bucket>
  <bucket startDate="2004-02-01T00:00:00" endDate="2004-03-01T00:00:00">
    <maximumLevel>2000</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
    <maximumCoverage>0</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>No</useCoverForMaximum>
  </bucket>
  <bucket startDate="2004-03-01T00:00:00" endDate="2004-04-01T00:00:00">
    <maximumLevel>2000</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
    <maximumCoverage>0</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>No</useCoverForMaximum>
  </bucket>
  <bucket startDate="2004-04-01T00:00:00" endDate="2004-05-01T00:00:00">
    <maximumLevel>2000</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
```

```
    <maximumCoverage>0</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>No</useCoverForMaximum>
  </bucket>
```

To enable planners to annotate optimized plans and document background information for what-if scenarios, Supply Chain Planning XML 3.2 includes the following features:

- A forecastCode element has been added to the Enterprise Forecast, Inventory Build Targets, and Sourcing Recommendations packages in any models where these packages exist.
- In Optimized Plan data packages for all models, a description field has been added to each optimized plan object.

Model	Optimized Plan Objects with Description Fields
Demand	enterpriseForecast
Strategic	enterpriseForecast, inventorySafetyTargets, inventoryBuildTargets, sourcingRecommendation
Tactical	netDeploymentRequirements, netProductionRequirements, enterpriseForecast, inventorySafetyTargets, inventoryBuildTargets, purchaseOrderRecommendation, sourcingRecommendation
Operation	productionSchedule, netDeploymentRequirements, netProductionRequirements, enterpriseForecast, inventorySafetyTargets, inventoryBuildTargets, purchaseOrderRecommendation, sourcingRecommendation
Base	detailedProductionPlan, deploymentPlan, masterProductionPlan, enterpriseForecast, inventorySafetyTargets, inventoryBuildTargets, purchasePlan, sourcingRecommendation

To support improved lead time modeling in Supply Chain Business Modeler, Supply Chain Planning XML 3.2 includes the following changes in data packages for all models but the Demand model:

- The Customer Sourcing element in the Customer data package now includes a time-varying leadTime element and a fixed leadTimeUom element.
- In the Supplier Item element in the Supplier data package, the standardLeadTime and premiumLeadTime elements are now time-varying.
- In the Transport Mode element in the Distribution data package, the leadTime and returnTripTime elements are now time-varying.

The following features in Supply Chain Planning XML 3.2 format were introduced to support improvements to the Supply Chain Business Modeler data model:

- The Crew, Machine, and Tool elements in Manufacturing data packages for all models except the Demand model include two new elements: a time-varying offloadThreshold element and a fixed offloadWindow element. These elements specify the loads at which EnterpriseOne Production Scheduling should schedule the operations on alternate resources, if alternate resources are available.
- In the Base package for all models but the Demand model, the Item Branch element now includes a storageLotSize element.
- In the Base package for all models, the FromUOM element in the Standard UOM element has been changed from a key field to a fixed field.

- In the Base package for all models, the Item Group element now includes an optional planningUom element.
- In the Operation BOR package for all models except the Demand model, the minimumRunQuantity element type has been changed from a string to a double. As a result, the default value for this element has been changed from an empty string to 0.
- In Sales Order packages for the Base, Operation, and Tactical models, the priority element in the Sales Order Detail has been changed from an integer to a string. This change ensures that values from Supply Chain Management systems can be transferred correctly to EnterpriseOne Supply Chain Planning applications.
- In the Base package for the Demand model, an optional Channel name field has been added to the Channel element.
- In the Item Branch element in the Base package for all models but the Demand model, a holdPeriod element was added and hours was added as an option for shelfLifeUom. Because the holdPeriod element is now in the Item Branch element, the holdPeriod element was removed from the Beginning Inventory object in all models where this object exists.
- In the Beginning Inventory package for the Strategic model, an availableDate element has been added.

Note. For complete Supply Chain Planning XML 3.2 format documentation, see the XML schema definitions using the Schema Viewer that ships with Supply Chain Business Modeler 8.12.1.

Supply Chain Planning XML 3.1 Features

In addition to features that are available in Supply Chain Planning XML 3.0, Supply Chain Planning XML 3.1 format includes the following features:

- Hierarchy data.
- Enhancements associated with Production Scheduling features.
- Enhancements for integration system efficiency.
- Enhancements associated with data model improvements.

To support data aggregation in EnterpriseOne Demand Consensus and Demand Forecasting and related objects in Supply Chain Business Modeler data models, Supply Chain Planning XML 3.1 includes customer, item, and branch hierarchy elements. Supply Chain Business Modeler uses hierarchy information to aggregate data in channels when you refresh data in the demand models. You can also use hierarchy data from Supply Chain Business Modeler to enable other applications, such as Demand Forecasting and Demand Consensus, to aggregate data.

The following elements have been added to the Base package for all model levels:

- Item Hierarchy Level.
- Item Hierarchy Node.
- Branch Hierarchy Level.
- Branch Hierarchy Node.

The following elements have been added to the Customer package for the Base, Operation, Tactical, and Strategic model levels:

- Customer Hierarchy Level.
- Customer Hierarchy Node.

The following elements have been added to the Base package for the Demand model:

- Channel Hierarchy Level.
- Channel Hierarchy Node.

To support Production Scheduling enhancements and related objects in Supply Chain Business Modeler data models, Supply Chain Planning XML 3.1 includes the following elements:

- To support Theory of Constraints functionality in Production Scheduling, the Tool, Machine, and Crew elements in Manufacturing data packages include a new element: capacityConstraintStatus. This element indicates whether the resource is constrained, with possible values of relaxed, ccr (capacity constrained resource), and constrained. The default value is constrained.
- The Supplier element in Supplier data packages includes two new child elements: timeFenceDuration and timeFenceDurationUnits. The time fence specifies when purchases can be delivered to the supplier, and supports purchase order recommendations in Production Scheduling.
- The Supplier Item element in Supplier data packages includes a new time-varying Preference attribute. The Preference can now vary over time.
- The Item Branch element in Base data packages includes new shippingCalendar and receivingCalendar elements.
- The Branch element in Base data packages for the Base, Operation, and Tactical model levels includes a new productionCalendar element.

The following features in Supply Chain Planning XML 3.1 format were introduced to improve the efficiency of Supply Chain Business Modeler integration systems:

- To reduce the size of data files, Supply Chain Planning XML 3.1 uses a compressed format for effective-dated data in packages for Tactical models. Rather than repeating the same data values in multiple consecutive time periods, Supply Chain Planning XML 3.1 data packages include unchanging effective-dated values in time periods that span model time periods.

  For example, in this sample Supply Chain Planning XML 3.1 data, inventory policy values for item 9797702 change only once and are specified in two time buckets— 2004–01–01 to 2004–02–03 and 2004–02–03 to 2004–01–01— instead of being repeated in daily, weekly, monthly, or quarterly time periods.

```
<inventoryPolicy>
  <itemCode>9797702</itemCode>
  <branchCode>GUNNISON</branchCode>
  <forecastConsumptionRule>Greater-Forecast</forecastConsumptionRule>
  <forecastConsumptionTimeFence>6</forecastConsumptionTimeFence>
  <timeFenceUnit>Months</timeFenceUnit>
  <coverageUnit>Months</coverageUnit>
  <bucketList>
  <bucket startDate="2004-01-01T00:00:00" endDate="2004-02-03T00:00:00">
    <maximumLevel>0</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
    <maximumCoverage>10</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>Yes</useCoverForMaximum>
  </bucket>
  <bucket startDate="2004-02-03T00:00:00" endDate="2036-01-01T00:00:00">
    <maximumLevel>2000</maximumLevel>
    <minimumLevel>0</minimumLevel>
    <stockoutCost>5</stockoutCost>
```



```
    <maximumCoverage>0</maximumCoverage>
    <useCoverForSafety>Yes</useCoverForSafety>
    <useCoverForMaximum>No</useCoverForMaximum>
  </bucket>
  </bucketList>
</inventoryPolicy>
```

- Some nested elements in Supply Chain Planning XML are collections of large sets of elements. In Supply Chain Planning XML 3.1, these elements are identified by a new type attribute value: collection. This attribute value enables Supply Chain Business Modeler to process the elements with improved memory usage.

  In the Base model, the following elements include the collection type:

- Time Series.
- Detailed Production Plan.
- Enterprise Forecast.
- Deployment Plan.
- Master Production Plan.
- Inventory Safety Target.
- Inventory Build Target.
- Purchase Plan.
- Sourcing Recommendation Plan.

The following features in Supply Chain Planning XML 3.1 format were introduced to improve the Supply Chain Business Modeler data model:

- The extendedPrice element was removed from the Demand Point History element for the Demand model and the Sales object in the Demand package for the Strategic model. This element was removed because the extendedPrice value is equal to the product of the values of two other element values: unitPrice and quantity.
- The lineItem and unitPrice elements were added to the Sales Order History elements for the Base model.
- In the Base package in the Demand model, the itemUomConversion element is now named itemUom, to be consistent with the naming for other models.
- In the Demand model, the Sales Orders package has been replaced by the Sales package. This package includes the total sales from sales orders in the Base model in each period.
- A capacityRequired element was added in the Work Order Bor element for the Base and Operation models.
- The uomType element in the StandardUom element for all models has a new valid value: Unspecified. The Unspecified value maps to the Count value in previous Supply Chain Planning XML versions.

Note. For complete Supply Chain Planning XML 3.1 format documentation, see the XML schema definitions that are provided with Supply Chain Business Modeler 8.12.1 in the *path*\scp\8.12.1\scbm\docs\xsd\3.1 directory.

See Also

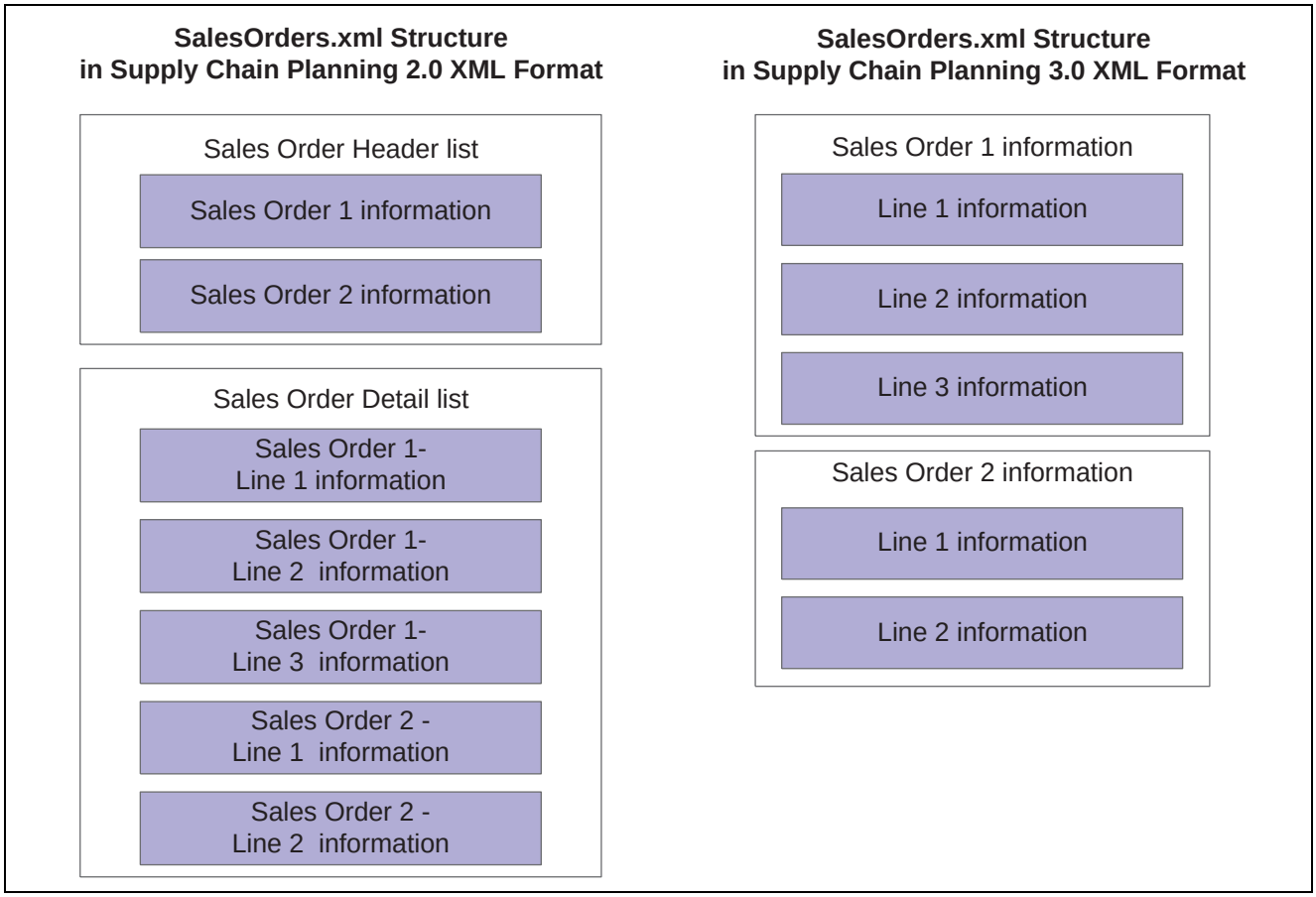
"Understanding Data for Importing into Supply Chain Business Modeler," XML Schema Definitions

Supply Chain Planning XML 3.0 and 2.0 Formats

In data packages in Supply Chain Planning XML 3.0 format, detailed information is nested within its related header data. For example, in a SalesOrders.xml document in Supply Chain Planning XML 3.0 format, information about lines in a sales order is nested within information about the sales order as a whole. All attributes of the sales order must appear before the detailed line information. Supply Chain Planning XML 3.0 format enables systems to stream data more quickly than in Supply Chain Planning XML 2.0 because systems do not have to perform joins on the data to link header and detail information.

In data packages in Supply Chain Planning XML 2.0 format, general information appears in separate Supply Chain Planning XML document sections from detailed information. For example, in a SalesOrders.xml document in Supply Chain Planning XML 2.0 format, customer and shipping information that applies to an entire sales order appears in a Sales Order Header element while detailed sales order information about items, dates and quantities ordered appears in a separate Sales Order Detail element. Systems can stream data in Supply Chain Planning XML 3.0 format more quickly than data in Supply Chain Planning XML 2.0 format. However, Supply Chain Business Modeler 8.12.1 can import and export data in Supply Chain Planning XML 2.0 format to ensure that it supports previously-implemented integration systems.

This diagram illustrates the differences between Supply Chain Planning XML 2.0 and 3.0 format by showing the structure of SalesOrders.xml documents in each format:



SalesOrders.xml structure in Supply Chain Planning XML 2.0 and Supply Chain Planning XML 3.0 format

To determine which elements are nested in documents in Supply Chain Planning XML 3.0 format, you can review the object list in the schema viewer. Elements that are nested in Supply Chain Planning XML documents appear indented beneath their parent objects. For example, the Sales Order Detail object is indented below the Sales Order Header. This indentation indicates that sales order detail information is nested within sales order header information in the Sales Orders package in Supply Chain Planning XML 3.0 format.

See Also

"Setting Up Models," Obtaining Model Information

XML Schema Definitions

For more information about Supply Chain Planning XML format, you can view the XML Schema Definitions (XSDs) that are shipped with Supply Chain Business Modeler. XSDs describe valid data package formats, including the elements that can appear, the order of the elements, and the valid data values in each package. You can reference the XSDs in XML documents to validate the data packages and ensure that they conform to the Supply Chain Planning XML format.

Supply Chain Business Modeler is shipped with XSDs that describe data packages that can be imported in full import scenarios and in incremental import scenarios. Because data in incremental import scenarios is merged with existing model data, data packages for incremental scenarios do not require all data values that are required in full import scenarios.

You can find XSDs for Supply Chain Planning XML format in the following locations:

XSD	Location
For Supply Chain Planning XML - Full import scenarios	In Windows: *path*\scp\8.12.1\scbm\docs\xsd*x.x*\full*model_type* In UNIX: *path*/scp/8.12.1/scbm/docs/xsd/*x.x*/full/*model_type* Where *path* is the directory where Supply Chain Business Modeler is installed, *x.x* is the version of Supply Chain Planning XML, and *model_type* is the type of model that you are importing data into or exporting data from.
For Supply Chain Planning XML - Incremental import scenarios	In Windows: *path*\scp\8.12.1\docs\xsd*x.x*\incremental*model_type* In UNIX: *path*/scp/8.12.1/docs/xsd/*x.x*/incremental/*model_type* Where *path* is the directory where Supply Chain Business Modeler is installed, *x.x* is the version of Supply Chain Planning XML, and *model_type* is the type of model that you are importing data into or exporting data from.

The following sample XSD includes annotated excerpts from a Base package XSD:

```
<!-- Specify that the document uses XML version 1.0 and the -->
 <!-- UTF-8 character set. (Supply Chain Business Modeler can import -->
 <!-- files that use any character set supported by the Xerces XML parser -->
 <!-- including UTF-8, ISO-8859-1, ASCII, EBCDIC, UTF-16, and Win-1252.) -->
 <!-- Specify that elements and data types come from the -->
 <!-- http://www.w3.org/2001/XMLSchema namespace and that elements -->
 <!-- from this namespace begin with xs: -->
 <?xml version=1.0 encoding=iso-8859-1?>
 <xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
 <!-- Specify that the root element of the XML document is a complex -->
 <!-- element called scbm-extract. In this example, this element can -->
 <!-- include itemList, standardUomList, and itemUomList elements. -->
 <!-- Because maxOccurs defaults to 1 and minOccurs=0 for these -->
```

```
 <!-- elements, itemList, standardUomList, and itemUomList can -->
 <!-- appear one or no times in the XML document. The sequence element -->
 <!-- indicates that if the itemList, standardUomList, and -->
 <!-- itemUomList elements appear, they must appear in the order -->
 <!-- specified. The scbm-extract element must have a version -->
 <!-- attribute with a value of 3.0. -->
<xs:element name=scbm-extract>
 <xs:complexType>
 <xs:sequence>
 <xs:element name=provenance type=provenanceType minOccurs=0 maxOccurs=1/>
 <xs:element name=itemList type=itemListType minOccurs=0/>
 <xs:element name=standardUomList type=standardUomListType minOccurs=0/>
 <xs:element name=itemUomList type=itemUomListType minOccurs=0/>
 </xs:sequence>
 <xs:attribute name=version type=xs:string fixed=3.0 use=required/>
 </xs:complexType>
 </xs:element>
 <!-- Specify that elements in the XML document with the provenanceType -->
 <!-- type can include source, comment and timestamp elements. The -->
 <!-- source and comment elements have the scbmString type. The -->
 <!-- timestamp element has the scbmDT type.-->

 <xs:complexType name=provenanceType>
<xs:all>
<xs:element name=source type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=comment type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=timestamp type=scbmDT minOccurs=0 maxOccurs=1 nillable=true/>
</xs:all>
</xs:complexType>
 <!-- Specify that elements in the XML document with the itemListType -->
 <!-- type can include any number of item elements with the -->
 <!-- itemObject type. -->
<xs:complexType name=itemListType>
 <xs:sequence>
 <xs:element name=item type=itemObject minOccurs=0 maxOccurs=unbounded />
 </xs:sequence>
 </xs:complexType>
 <!-- Specify that elements in the XML document with the -->
 <!-- standardUomListType type can include any number of -->
 <!-- standardUom elements with the standardUomObject type. -->
 <xs:complexType name=standardUomListType>
 <xs:sequence>
 <xs:element name=standardUom type=standardUomObject minOccurs=0
maxOccurs=unbounded />
 </xs:sequence>
 </xs:complexType>
 <!-- Specify that elements in the XML document with the itemUomListType -->
 <!-- type can include any number of itemUom elements with the -->
 <!-- itemUomObject type. -->
 <xs:complexType name=itemUomListType>
```



```
 <xs:sequence>
 <xs:element name=itemUom type=itemUomObject minOccurs=0 maxOccurs=unbounded />
 </xs:sequence>
 </xs:complexType>
 <!-- Specify that elements with the itemObject type can include -->
 <!-- itemCode, itemName, alternateItemId, description, planningUom, -->
<!-- shippingUom, weight, weightUom, volume, volumeUom and -->
<!-- storageRequirement elements. The weight and volume elements -->
 <!-- have the scbmDouble type. The remaining elements have the -->
<!-- scbmString type. xs:all specifies that these elements can -->
<!-- appear in any order. minOccurs=1 specifies that the itemCode -->
<!-- and planningUom elements are required. minOccurs=0 specifies -->
 <!-- that an element is not required, while nillable=true -->
 <!-- specifies that an element can appear but be empty. -->
<xs:complexType name=itemObject>
 <xs:all>
<xs:element name=itemCode type=scbmString minOccurs=1 maxOccurs=1/>
<xs:element name=itemName type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=alternateItemId type=scbmString minOccurs=0
maxOccurs=1 nillable=true/>
<xs:element name=description type=scbmString minOccurs=0
maxOccurs=1 nillable=true/>
<xs:element name=planningUom type=scbmString minOccurs=1 maxOccurs=1/>
<xs:element name=shippingUom type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=weight type=scbmDouble minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=weightUom type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=volume type=scbmDouble minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=volumeUom type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
<xs:element name=storageRequirement type=scbmString minOccurs=0
maxOccurs=1 nillable=true/>
</xs:all>
</xs:complexType>
<!-- Specify that elements with the standardUomObject type can -->
<!-- include the toUom, unitType, fromUom, and factor elements -->
<!-- in any order. The toUom, unitType, and factor elements must -->
<!-- appear once because minOccurs=1 and maxOccurs=1 for these -->
<!-- elements. The fromUom element is not required. The toUom, -->
<!-- fromUom, and factor elements have the scbmString type. -->
<!-- The factor element has the scbmDouble type. Possible values -->
<!-- for the toUomType element are: Weight, Volume, Length, Count, -->
<!-- and Area. -->
<xs:complexType name=standardUomObject>
<xs:all>
<xs:element name=toUom type=scbmString minOccurs=1 maxOccurs=1/>
<xs:element name=unitType minOccurs=1 maxOccurs=1>
<xs:simpleType>
<xs:restriction base=xs:string>
 <xs:enumeration value=Weight/>
<xs:enumeration value=Volume/>
<xs:enumeration value=Length/>
```

```
<xs:enumeration value=Count/>
 <xs:enumeration value=Area/>
</xs:restriction>
</xs:simpleType>
</xs:element>
 <xs:element name=fromUom type=scbmString minOccurs=0 maxOccurs=1 nillable=true/>
 <xs:element name=factor type=scbmString minOccurs=1 maxOccurs=1/>
</xs:all>
 </xs:complexType>
 <!-- Specify that elements with the itemUomObject type can -->
 <!-- include the itemCode, toUom, toUomType, and factor elements -->
 <!-- in any order. Each of these elements must appear once because -->
<!-- minOccurs=1 and maxOccurs=1 for these elements. The itemCode -->
 <!-- and toUom elements have the scbmString type. Possible values -->
<!-- for the toUomType element are: Weight, Volume, Length, Count, -->
 <!-- and Area. The factor element has the scbmDouble type. -->
 <xs:complexType name=itemUomObject>
 <xs:all>
 <xs:element name=itemCode type=scbmString minOccurs=1 maxOccurs=1/>
 <xs:element name=toUom type=scbmString minOccurs=1 maxOccurs=1/>
 <xs:element name=toUomType minOccurs=1 maxOccurs=1>
 <xs:simpleType>
 <xs:restriction base=xs:string>
 <xs:enumeration value=Weight/>
 <xs:enumeration value=Volume/>
 <xs:enumeration value=Length/>
 <xs:enumeration value=Count/>
 <xs:enumeration value=Area/>
 </xs:restriction>
 </xs:simpleType>
 </xs:element>
 <xs:element name=factor type=scbmDouble minOccurs=1 maxOccurs=1/>
 </xs:all>
 </xs:complexType>
 <!-- Specify that elements with the scbmString or scbmDouble type -->
 <!-- can accept isNull=true or isNull=false as attributes. -->
<xs:complexType name=scbmString>
 <xs:simpleContent>
 <xs:extension base=xs:string>
<xs:attribute name=isNull type=simpleTrueFalse/>
 </xs:extension>
 </xs:simpleContent>
 </xs:complexType>
 <xs:complexType name=scbmDouble>
 <xs:simpleContent>
 <xs:extension base=xs:double>
 <xs:attribute name=isNull type=simpleTrueFalse/>
 </xs:extension>
 </xs:simpleContent>
 </xs:complexType>
```



```
 <!-- Specify that elements with the scbmDT type is restricted to the -->
 <!-- datetime yyyy-mm-ddTHH:MM:SS format with the specified pattern of -->
 <!-- values. -->
 <xs:simpleType name=scbmDT>
 <xs:union>
 <xs:simpleType>
 <xs:restriction base=xs:dateTime/>
 </xs:simpleType>
 <xs:simpleType>
 <xs:restriction base=xs:string>
 <xs:pattern value=[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]
:[0-5][0-9]:[0-5][0-9] id=OWDateTimeFormat.pattern/>
 </xs:restriction>
 </xs:simpleType>
 </xs:union>
</xs:simpleType>
 </xs:schema>
```

Sample Data Packages

Supply Chain Business Modeler is shipped with sample data packages in Supply Chain Planning XML 3.3 format for full import scenarios. Sample data packages are saved in the *path*\scp\8.12.1\scbm\sample_data\SCP_bikes directory in Windows or the *path*/scp/8.12.1/scbm/sample_data/SCP_bikes directory in UNIX, where *path* is the directory where Supply Chain Business Modeler is installed.

The following sample document in Supply Chain Planning XML 3.0 format is an annotated excerpt from a Base package. This Supply Chain Planning XML document conforms to the structure specified by the sample XSD that is included in this implementation guide.

```
<!-- Specify that the document uses XML version 1.0 and the UTF-8 -->
<!-- character set. (Supply Chain Business Modeler can import files -->
<!-- that use any character set supported by the Xerces XML parser, -->
<!-- including ISO-8859-1, ASCII, EBCDIC, UTF-16, UTF-8, and Win-1252.) -->
<?xml version=1.0 encoding=UTF-8?>
<!-- Specify an element called scbm-extract that has a version -->
<!-- attribute value of 3.2 -->
<scbm-extract version="3.2">
<!-- Specify an element called provenance with source, comment, and -->
<!-- timestamp information. Note: This data is not currently used in -->
<!-- Supply Chain Business Modeler, and is provided as documentation -->
<!-- for the extract. -->
<provenance>
 <source>EnterpriseOne Supply Chain Management</source>
 <comment>base model</comment>
 <timestamp>2003-12-05T11:22:56</timestamp>
</provenance>
<!-- Specify that the scbm-extract element has a child element -->
<!-- called itemList. -->
```

```
<itemList>
<!-- Specify that the itemList element has a child element -->
<!-- called item. Specify the item code, name, alternate item ID, -->
<!-- description, planningUom, shippingUom, weight, weightUom -->
<!-- volume, volumeUom, storageRequirement, and schedulingPrecision. -->
 <item>
  <itemCode>9797700</itemCode>
  <itemName>5900_Road</itemName>
  <alternateItemId>9797700EA</alternateItemId>
  <description>Trek 5900 OCLV 110 Road Bike with Dura-Ace </description>
  <planningUom>EA</planningUom>
  <shippingUom>PL</shippingUom>
  <weight>20</weight>
  <weightUom>LB</weightUom>
  <volume>18</volume>
  <volumeUom>Cubic Feet</volumeUom>
  <storageRequirement>FINISHED GOODS</storageRequirement>
  <schedulingPrecision>0</schedulingPrecision>
 </item>
<!-- Specify another item child element of the itemList element -->
<!-- Specify the item code, name, alternate item ID, description -->
<!-- planningUom, shippingUom, weight, weightUom, volume, volumeUom, -->
<!-- storageRequirement, and schedulingPrecision. -->
 <item>
  <itemCode>9797701</itemCode>
  <itemName>5900_Road_LA</itemName>
  <alternateItemId>9797701EA</alternateItemId>
  <description>Trek 5900 OCLV 110 Road Bike with Dura-Ace Lance Armstrong Limited
   Edition</description>
  <planningUom>EA</planningUom>
  <shippingUom>PL</shippingUom>
  <weight>20</weight>
  <weightUom>LB</weightUom>
  <volume>18</volume>
  <volumeUom>Cubic Feet</volumeUom>
  <storageRequirement>FINISHED GOODS</storageRequirement>
  <schedulingPrecision>0</schedulingPrecision>
 </item>
<!-- Specify another item child element of the itemList element -->
<!-- Specify the item code, name, alternate item ID, description -->
<!-- planningUom, shippingUom, weight, weightUom, volume, volumeUom, -->
<!-- storageRequirement, and schedulingPrecision. -->
 <item>
  <itemCode>9797702</itemCode>
  <itemName>5500_Road</itemName>
  <alternateItemId>9797702EA</alternateItemId>
  <description>Trek 5500 OCLV 120 Road Bike with Dura-Ace</description>
  <planningUom>EA</planningUom>
  <shippingUom>PL</shippingUom>
```



```
  <weight>20</weight>
  <weightUom>LB</weightUom>
  <volume>18</volume>
  <volumeUom>Cubic Feet</volumeUom>
  <storageRequirement>FINISHED GOODS</storageRequirement>
  <schedulingPrecision>0</schedulingPrecision>
 </item>
</itemList>
<!-- Specify that the scbm-extract element has a child element -->
<!-- called standardUomList. -->
<standardUomList>
<!-- Specify that the standardUomList element has a child element -->
<!-- called standardUom. Specify the toUom, unitType, fromUom and -->
<!-- factor of the standardUom. -->
 <standardUom>
  <toUom>KG</toUom>
  <unitType>Weight</unitType>
  <fromUom>LB</fromUom>
  <factor>0.454545454545455</factor>
 </standardUom>
<!-- Specify another standardUomList child element called -->
<!-- standardUom. Specify the toUom, unitType, fromUom and factor -->
<!-- of the standardUom. -->
 <standardUom>
  <toUom>LB</toUom>
  <unitType>Weight</unitType>
  <fromUom>LB</fromUom>
  <factor>1</factor>
 </standardUom>
<!-- Specify another standardUomList child element called -->
<!-- standardUom. Specify the toUom, unitType, fromUom and factor -->
<!-- of the standardUom. -->
 <standardUom>
  <toUom>LT</toUom>
  <unitType>Volume</unitType>
  <fromUom>ML</fromUom>
  <factor>0.001</factor>
 </standardUom>
</standardUomList>
<!-- Specify that the scbm-extract element has a child element -->
<!-- called itemUomList. -->
<itemUomList>
<!-- Specify that the itemUomList element has a child element -->
<!-- called itemUom. Specify the itemCode, toUom, toUomType, and -->
<!-- factor of the itemUom. -->
 <itemUom>
  <itemCode>9797700</itemCode>
  <toUom>EA</toUom>
  <toUomType>Count</toUomType>
  <factor>1</factor>
```

```
 </itemUom>
<!-- Specify another itemUomList child element called itemUom. -->
<!-- Specify the itemCode, toUom, toUomType, and factor. -->
 <itemUom>
  <itemCode>9797700</itemCode>
  <toUom>LB</toUom>
  <toUomType>Weight</toUomType>
  <factor>25</factor>
 </itemUom>
<!-- Specify another itemUomList child element called itemUom. -->
<!-- Specify the itemCode, toUom, toUomType, and factor. -->
 <itemUom>
  <itemCode>9797700</itemCode>
  <toUom>PL</toUom>
  <toUomType>Count</toUomType>
  <factor>6</factor>
 </itemUom>
<!-- Specify another itemUomList child element called itemUom. -->
<!-- Specify the itemCode, toUom, toUomType, and factor. -->
 <itemUom>
  <itemCode>9797701</itemCode>
  <toUom>EA</toUom>
  <toUomType>Count</toUomType>
  <factor>1</factor>
 </itemUom>
</itemUomList>
</scbm-extract>
```

Compressed Supply Chain Planning XML Files

Supply Chain Business Modeler can import and export Supply Chain Planning XML data files in compressed gzip format. When data files are reduced in size using gzip format, you can transfer the data between supply chain systems more quickly than if the files were uncompressed.

Other Supply Chain Planning 8.12.1 applications can also create and process Supply Chain Planning XML files in compressed format. You can import compressed data files from Supply Chain Business Modeler directly into other Supply Chain Planning applications, and import compressed data files from other Supply Chain Planning applications into Supply Chain Business Modeler.

To further optimize supply chain integration systems, you can compress Supply Chain Planning XML files from other sources using the gzip compression utility. For example, before transferring data files from a Supply Chain Management system, you can reduce the size of the data files using the gzip utility. You can then transfer the data files and import them into Supply Chain Business Modeler.

When you run an import scenario, Supply Chain Business Modeler automatically detects whether a data file is compressed or uncompressed and imports the data. When you create an export scenario, you can specify whether you want to export the Supply Chain Planning XML data packages in compressed or uncompressed format.

If a data folder in an import scenario includes a compressed data package and an uncompressed data package with the same name, Supply Chain Business Modeler imports only the uncompressed package. For example, if a data folder includes both a Base.xml and a Base.xml.gz file, Supply Chain Business Modeler imports the Base.xml package.

Supply Chain Planning XML files in gzip format have the .xml.gz extension.

CHAPTER 4

Starting Supply Chain Business Modeler

This chapter discusses how to:

- Start Supply Chain Business Modeler in Windows.
- Start Supply Chain Business Modeler in UNIX.
- Start Supply Chain Business Modeler in Linux.

Note. You can run only one Supply Chain Business Modeler or Business Modeler Shell session with the same database at a time. Supply Chain Business Modeler will not start if another Supply Chain Business Modeler or Business Modeler Shell session is running using the same database.

Starting Supply Chain Business Modeler in Windows

To start Supply Chain Business Modeler in Windows, do one of the following:

- Select Start, Programs, EnterpriseOne Supply Chain Planning 8.12.1, Supply Chain Business Modeler, Supply Chain Business Modeler.
- From a command prompt, navigate to the *path*\scp\8.12.1\common\start directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
run_scbm_scbm.bat
```

Starting Supply Chain Business Modeler in UNIX

To start Supply Chain Business Modeler in UNIX:

1. Set the display to the following value:

   ```
   ipaddress:0.0
   ```

   Where *ipaddress* is the IP address of the machine where you want to display the Supply Chain Business Modeler graphical user interface.

2. Navigate to the *path*/scp/8.12.1/common/start directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

   ```
   ./run_scbm_scbm.sh
   ```

Starting Supply Chain Business Modeler in Linux

To start Supply Chain Business Modeler in Linux:

1. Set the display to the following value:

   ```
   ipaddress:0.0
   ```

   Where *ipaddress* is the IP address of the machine where you want to display the Supply Chain Business Modeler graphical user interface.

2. Navigate to the *path*/scp/8.12.1/common/start directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

   ```
   ./run_scbm_scbm.sh
   ```

CHAPTER 5

Managing Extract Areas and Data Folders

This chapter discusses how to:

- Create, modify, and delete extract areas.
- Create, rename, and delete data folders.

Creating, Modifying, and Deleting Extract Areas

This section provides an overview of extract areas, lists common elements, and lists the workspaces used to create, modify, and delete extract areas.

Understanding Extract Areas

Before you can import data into or export data from Supply Chain Business Modeler, you must create at least one extract area. An extract area is a directory on an operating system that is managed from within Supply Chain Business Modeler. This directory contains subdirectories for storing data in Supply Chain Planning XML format, and Tool Command Language (Tcl) scripts for use in import scenarios and export scenarios.

You can associate more than one import or export scenario with a single extract area. However, users usually create multiple extract areas for logically organizing their data extracts.

When you create an extract area, the system creates a directory in the location on the operating system that you specify. In this directory, the system creates a data subdirectory and a scripts subdirectory. The data subdirectory is the location where the system saves data folders. The scripts subdirectory can store Tcl scripts for running before or after you import or export data.

After you create an extract area, you must add at least one data folder. Data folders are used for storing data in Supply Chain Planning XML format that can be imported into or has been exported from Supply Chain Business Modeler.

Note. If you delete the directory on the operating system where you saved an extract area, the extract area is no longer valid.

File names should be in ASCII format, or another format that can be used by the operating system locale.

Prerequisites

Before you complete the tasks in this section:

- You must obtain sufficient file permissions from the system administrator before you can create extract areas.
- You must ensure that sufficient disk space is available before you can create extract areas.

Common Elements Used in This Section

Name Enter the name of the extract area. Each extract area must have a unique name. Extract area names can only include ASCII characters; they cannot include international characters.

Location Enter the location of the extract area directory on the operating system.

Folder Enter the name of the extract area directory on the operating system. The directory name appears by default from the Name field, but you can enter a different name for the extract area directory. You must specify a directory name that does not already exist in the specified location. The directory name cannot include any characters that are reserved by the operating system: \ / : * ? < > | in Windows and / in UNIX.

Workspaces Used to Create, Modify, and Delete Extract Areas

Window Name	Navigation	Usage
Add Extract Area	• Click the Extract Areas button in the Shortcuts bar. Select File, Extract Area, Add Extract Area. • Click the Extract Areas button in the Shortcuts bar. Right-click the Extract Area Navigator and select Add Extract Area.	Create extract areas.
Extract Area Properties	Click the Extract Areas button in the Shortcuts bar. Right-click the extract area for which you want a new directory, and select Properties.	Change extract area directories.
Extract Area - Rename	• Click the Extract Areas button in the Shortcuts bar. Select the extract area that you want to rename and select Edit, Rename. • Click the Extract Areas button in the Shortcuts bar. Right-click the extract area that you want to rename and select Rename.	Rename extract areas.
Extract Area - Delete	• Click the Extract Areas button in the Shortcuts bar. Select the extract area that you want to delete and select Edit, Delete. • Click the Extract Areas button in the Shortcuts bar. Right-click the extract area that you want to delete and select Delete.	Delete extract areas.

See Also

Creating, Renaming, and Deleting Data Folders

“Importing Data into Supply Chain Business Modeler”

“Exporting Data from Supply Chain Business Modeler”

Creating, Renaming, and Deleting Data Folders

This section provides an overview of data folders and lists the workspaces used to create, rename, and delete data folders.

Understanding Data Folders

Before you can import data into or export data from Supply Chain Business Modeler, you must create at least one data folder. A data folder is an extract area subdirectory that is managed from within Supply Chain Business Modeler. On the operating system, data folders are saved in a data subdirectory in the extract area directory.

Data folders store data in Supply Chain Planning XML format that can be imported into or has been exported from Supply Chain Business Modeler. Data folders help to logically organize XML files within an extract area. For example, in an extract area called "Inbound to Supply Chain Business Modeler", you might create the following data folders:

Data Folder Name	Description
From Supply Chain Management	This data folder contains enterprise and transactional data from a supply chain management system for importing into the Base model.
From DM	This data folder contains inventory safety target and enterprise forecast data from EnterpriseOne Demand Forecasting and Demand Consensus for importing into Supply Chain Business Modeler Demand model.
From SNO	This data folder contains inventory build targets, inventory safety targets, and sourcing recommendations from EnterpriseOne Strategic Network Optimization for importing into Supply Chain Business Modeler Strategic model.
From Cranston PDP	This data folder contains net deployment requirements and net production requirements for the Cranston branch from a Production and Distribution Planning model for importing into a planning model that contains data for Cranston at the Tactical level in Supply Chain Business.
From Gunnison PDP	This data folder contains net deployment requirements and net production requirements for the Gunnison branch from a Production and Distribution Planning model for importing into a planning model that contains data for Gunnison at the Tactical level in Supply Chain Business Modeler.

Data Folder Name	Description
From Cranston PS	This data folder contains production schedule data for the Cranston branch from a Production Scheduling model for importing into a planning model that contains data for Cranston at the Operation level in Supply Chain Business Modeler.
From Gunnison PS	This data folder contains production schedule data for the Gunnison branch from a Production Scheduling model for importing into a planning model that contains data for Gunnison at the Operation level in Supply Chain Business Modeler.

When you create an import or export scenario, you must specify the extract area and data folder to import data from or export data to.

Prerequisites

Before you complete the tasks in this section:

- You must obtain sufficient file permissions from the system administrator before you can create data folders.
- You must ensure that sufficient disk space is available before you can create data folders.

Note. Data folder names can only include ASCII characters; they cannot include international characters.

See Also

"Importing Data into Supply Chain Business Modeler"

"Exporting Data from Supply Chain Business Modeler"

Workspaces Used to Create, Rename, and Delete Data Folders

Window Name	Navigation	Usage
Extract Areas - Add Data Folder	• Click the Extract Areas button in the Shortcuts bar. Select the extract area where you want to add a data folder. Select File, Extract Area, Add Data Folder. • Click the Extract Areas button in the Shortcuts bar. Right-click the extract area where you want to add a data folder and select Add Data Folder.	Create data folders.
Extract Areas - Rename Data Folder	• Click the Extract Areas button in the Shortcuts bar. Select the data folder that you want to rename and select Edit, Rename. • Click the Extract Areas button in the Shortcuts bar. Right-click the data folder that you want to rename and select Rename.	Rename data folders.
Extract Areas - Delete Data Folder	• Click the Extract Areas button in the Shortcuts bar. Select the data folder that you want to delete and select Edit, Delete. • Click the Extract Areas button in the Shortcuts bar. Right-click the data folder that you want to delete and select Delete.	Delete data folders.

CHAPTER 6

Importing Data into Supply Chain Business Modeler

This chapter provides overviews of importing data into Supply Chain Business Modeler, full and incremental import scenarios, importing new data objects in full and incremental import scenarios, updating data in incremental import scenarios, and importing data in Supply Chain Planning XML 2.0 format in full import scenarios, and discusses how to:

- Set operation and routing display names.
- Create, modify, and delete import scenarios.
- Run import scenarios.

Understanding Importing Data into Supply Chain Business Modeler

To import data into Supply Chain Business Modeler, you must create and run import scenarios. When you create an import scenario, you must specify:

- The way you want operation and routing names to be displayed in Supply Chain Planning applications like Production Scheduling.
- The data to import into Supply Chain Business Modeler.

  Data that you import into Supply Chain Business Modeler must be in Supply Chain Planning XML format. The files can be compressed in gzip format, or uncompressed.

  You do not have to import data into all packages in a model. For example, if you have a distribution-only business, you do not need to import data into manufacturing objects when you populate models with data from supply chain management and planning systems.
- The model to populate with the data.
- Any Tool Command Language (Tcl) scripts to run before or after importing the data.

  Tcl scripts enable you to link external processes to the import scenario.
- Whether you want to create a full import scenario or an incremental import scenario.

  In a full import scenario, Supply Chain Business Modeler imports full data packages and replaces data in the model with the imported data.

  In an incremental import scenario, Supply Chain Business Modeler imports incremental data packages and merges the data with existing data in the model. Incremental data packages do not require all data values that are required in full data packages.

Note. Supply Chain Business Modeler is shipped with XML schema definitions that describe full data packages and incremental data packages.

See "Understanding Data for Importing into and Exporting from Supply Chain Business Modeler," XML Schema Definitions.

You can create multiple import scenarios to support different business processes. For example, you could create the following scenarios:

- A full import scenario that imports item, location, customer, and supplier data from a supply chain management system into the Base model.
- An incremental import scenario that runs a script to extract data for a particular business unit from a supply chain management system and merges the data with data from other business units that is already populated in the Base model.
- A full import scenario that initiates a solve in Production Scheduling, exports optimized plans from Production Scheduling, and imports the schedule into a planning model at the Operation level.

  You can then run the scenarios as often as required to support business processes.

Note. Before you can create Supply Chain Business Modeler planning models, you must create and run an import scenario that imports branch data into the Base model.

See Also

Running Import Scenarios

Understanding Operation and Routing Display Names

When using Supply Chain Planning applications such as Production Scheduling, it is helpful to customize the appearance of the operation and routing names to improve reporting.

Without customization, EnterpriseOne Supply Chain Management programs export extremely long operation and routing names that include a concatenation of many fields. In particular, Production Scheduling creates production schedules with detailed operation and routing assignments on the screen where space is at a premium. In the original format, the long operation and routing names might be considered confusing and difficult to read. In addition to the OperationCode and RoutingCode fields, which contain the original EnterpriseOne contents, Supply Chain Business Modeler provides OperationName and RoutingName fields that can be set either within the Supply Chain Business Modeler application, or by a BMSH command. This global setting allows you to customize how operation and routing names are displayed in Supply Chain Planning applications upon data import, including the EnterpriseOne fields to be included and their order.

Alternatively, if you choose to customize the operation and routing display names directly in the EnterpriseOne extract files, the display name setting allows your operationName and routingName customizations to be used upon import instead of the original EnterpriseOne names. Any blank display name fields default to the original operationCode or routingCode contents.

When Supply Chain Business Modeler exports data back to EnterpriseOne, the operation and routing codes are returned in the original format.

See Also

"Setting Operation and Routing Display Names."

Understanding Full and Incremental Import Scenarios

When you create an import scenario, you must specify whether you want to create a full or incremental import scenario.

In a full import scenario, Supply Chain Business Modeler:

- Imports full data packages into a model.

  Full data packages require some data values that are not required in incremental data packages.

- Replaces existing data in the model with data from the full data packages.

  If you import data in Supply Chain Planning XML 3.3, 3.2, 3.1 or 3.0 format, Supply Chain Business Modeler determines which enterprise and transactional data is different in the data packages than in the destination models. Supply Chain Business Modeler then imports only the new or changed enterprise and transactional data, deletes from the model any data that is not in the data packages, and assigns default values for required data that is missing from the data packages.

  If you import data in Supply Chain Planning XML 2.0 format, does not determine which data is new or changed.

In an incremental import scenario, Supply Chain Business Modeler:

- Imports incremental data packages.

  Incremental data packages do not require all data elements that are required in full data packages.

- Merges the data with existing data in the destination model.

  If the model includes data that is not in the incremental packages, Supply Chain Business Modeler does not delete data from the destination model. The only way to "delete" data in an incremental import scenario is to change an existing data value to a value that is equivalent to deleted. For example, you could change a sales order status to be rejected, change a quantity to zero, or change an available value to not available.

After running a full import scenario to populate the Base model with enterprise data from a supply chain management system, you could run an incremental import to add transport mode data or data for different branches from another system. You can also use incremental import scenarios to import large amounts of Supply Chain Planning XML data. For example, if you have ten million records of Supply Chain Planning XML data, you could import five million records using a full import scenario, and add the remaining five million records using an incremental import scenario.

Note. Do not import incremental data packages using full import scenarios. If you import incremental data packages using full import scenarios, Supply Chain Business Modeler assigns default values for data that is missing from the incremental data packages.

If you run an incremental import scenario before you run a full import scenario, the import fails if the data packages refer to data values that do not exist in the model or in the data that you are importing. For example, if you import data that refers to a branch that does not exist in the Branch object in the data or in the model, the incremental import fails.

See Also

"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler," XML Schema Definitions

Understanding Importing New Data Objects in Full and Incremental Import Scenarios

In a full or incremental import scenario, if you import a Supply Chain Planning XML document that includes data with a different primary key than existing model objects, Supply Chain Business Modeler creates a new data object with the data. For example, if you import a Supply Chain Planning XML document that includes data for Sales Order 048091-020505, and this sales order number does not exist in the destination model, Supply Chain Business Modeler creates a new object with the Sales Order 048091-020505 data.

When you add new data objects to a model using a full or incremental import scenario, Supply Chain Business Modeler assigns default values for some missing data values and writes error messages to the error log for other missing data. Data values can be missing if elements are missing from Supply Chain Planning XML files, or if elements are present but have no specified values.

This table describes how Supply Chain Business Modeler handles various types of missing data when you import new data objects:

Data Type	Supply Chain Business Modeler Action
Required	Writes an error to the error log without assigning a default value for the data.
Object reference	Writes an error to the error log without assigning a default value for the data.
Has a default value	Assigns the default value for the data. **Note.** You can view default values for data elements in the schema viewer.
Other	Assigns the default value for the data type.

This table shows the default values that Supply Chain Business Modeler assigns for missing data values that are not required, are not object references, and do not have default values:

Data Type	Default Value
Integer	0
Double	0
String	"" (empty string)
DateTime	"" (empty string)

Note. To determine the data type of an element, you can use the schema viewer that is provided with Supply Chain Business Modeler.

Understanding Updating Data in Incremental Import Scenarios

When you run a full import scenario that includes data with the same primary key as an existing model object, Supply Chain Business Modeler overwrites existing data values in the model and assigns default values as if it was creating a new data object.

When you run an incremental import scenario that includes data with the same primary key as an existing model object, Supply Chain Business Modeler updates the data in that model object with data from the Supply Chain Planning XML documents.

This section discusses:

- Updating fixed data values in existing model objects using incremental imports.
- Updating time-varying data values in existing model objects using incremental imports.
- Updating data in time periods in existing model objects using incremental imports.

Note. To determine each element's data type, you can use the schema viewer that is provided with Supply Chain Business Modeler.

Updating Fixed Data Values In Existing Model Objects Using Incremental Imports

When you run an incremental import scenario of data that has the same primary key as an existing model object, you can update fixed data values in the model object.

This table describes how Supply Chain Business Modeler updates fixed data values in incremental import scenarios:

Supply Chain Planning XML Document Contents	Result
Includes a value for a fixed data element (for example, <weight>20</weight>).	The value from the Supply Chain Planning XML document replaces the existing value in the model object.
Includes an empty fixed data element (for example, <weight></weight>).	The value in the model object remains unchanged.
Does not include a fixed data element.	The value in the model object remains unchanged.

Updating Time-Varying Data in Existing Model Objects Using Incremental Imports

When you run an incremental import scenario with data that has the same primary key as an existing model object, you can update time-varying data in the Base model or in an Operation model. Time-varying data appears in timeVaryingEvent elements in Supply Chain Planning XML documents.

In an incremental import scenario, the Supply Chain Planning XML document must include a timeVaryingEvent for every effective date in an object. Otherwise, all data for that effective date is removed from the destination model object. For example, if a model has inventory policy values for an item at a branch that begin on January 1, 2006 at midnight, and you update the model with a Supply Chain Planning XML file that does not include a timeVaryingEvent element that starts at this date and time, the model object no longer includes data for these dates.

This table describes how Supply Chain Business Modeler updates time-varying data values in incremental import scenarios:

Supply Chain Planning XML Document Contents	Result
Includes a timeVaryingEvent element that has the same effective date as the data object in the model. The timeVaryingEvent element includes an effective-dated element with a new data value.	The value from the Supply Chain Planning XML document replaces the existing value in the model object.
Includes a timeVaryingEvent element that has the same effective date as the data object in the model. The timeVaryingEvent element includes an empty element (for example, <cost></cost>).	The value in the model object remains unchanged.
Includes a timeVaryingEvent element that has the same effective date as the data object in the model. The timeVaryingEvent element does not include a particular element.	The value in the model object remains unchanged.
Includes a timeVaryingEvent element with an effective date that does not exist in the model object.	A time-varying event with the effective date from the Supply Chain Planning XML document is*not*created in the model object.

Updating Data in Time Periods in Existing Model Objects Using Incremental Imports

When you run an incremental import scenario to import data that has the same primary key as an existing model object, you can update data in Tactical, Strategic, and Demand model time periods. Data for time periods in these models appears in bucket elements in Supply Chain Planning XML documents. The following table describes how Supply Chain Business Modeler updates data values in time periods by importing Supply Chain Planning XML documents in incremental import scenarios:

Supply Chain Planning XML Document Contents	Result
Includes a bucketList element that begins after the last time period in the model.	A new time period is added after the last time period in the model. **Note.** You cannot insert a time period between existing time periods in a model.
Includes a bucketList element with the same start date and end date as an existing time period in the model object. The bucketList element includes an element with a new data value.	The value from the Supply Chain Planning XML document replaces the value in the time period in the model object.

Supply Chain Planning XML Document Contents	Result
Includes a bucketList element with the same start date and end date as an existing time period in the model object. The bucketList element includes an empty element (for example, <quantity></quantity>).	The value in the time period in the model object remains unchanged.
Includes a bucketList element with the same start date and end date as an existing time period in the model object. The bucketList element does not include a particular element.	The value in the time period in the model remains unchanged.

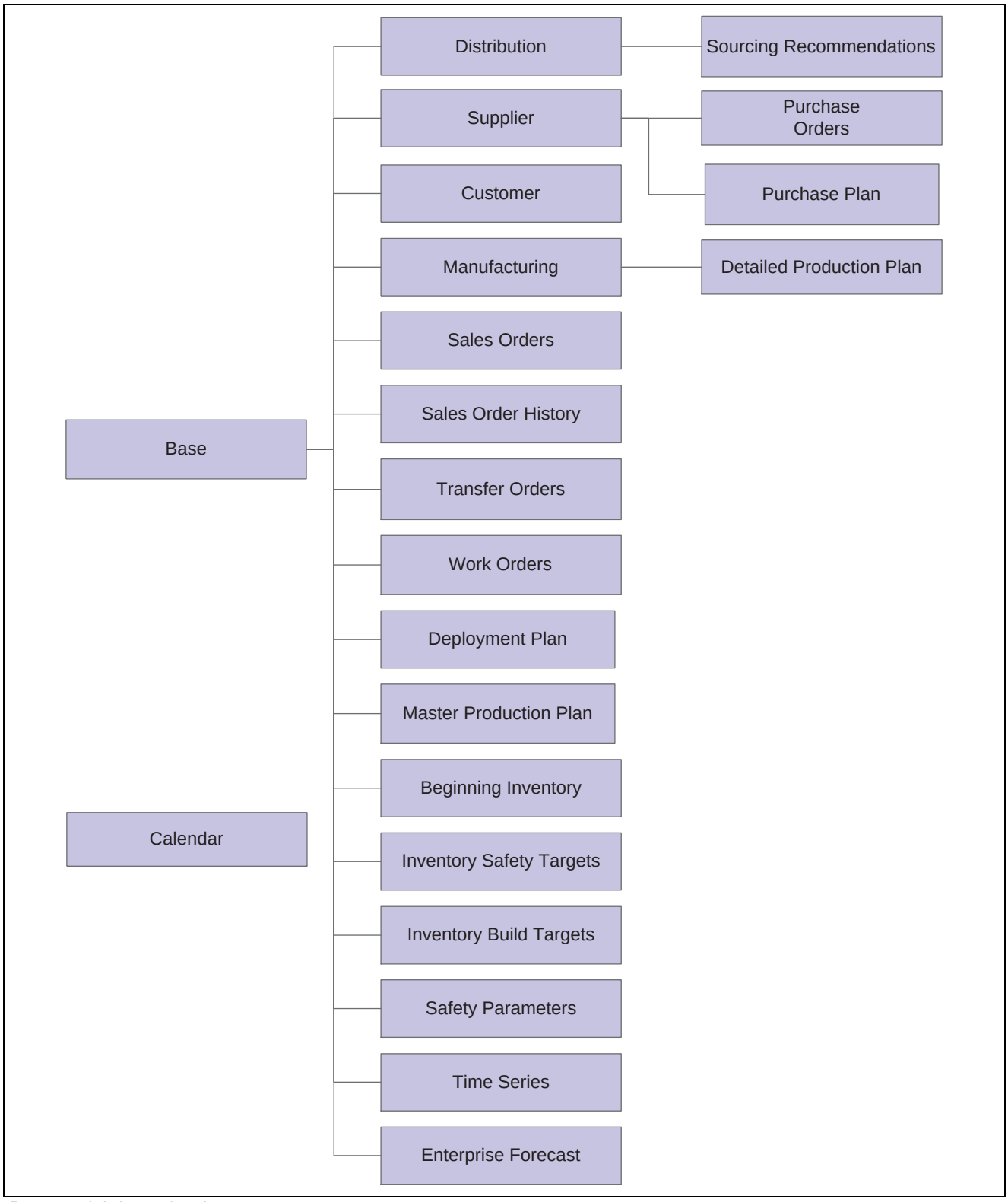
Understanding Importing Data in Supply Chain Planning XML 2.0 Format in Full Import Scenarios

In full import scenarios, Supply Chain Business Modeler usually imports only enterprise and transactional data that is different in the imported XML packages than in the destination model. However, when you run a full import scenario with data in Supply Chain Planning XML 2.0 format, Supply Chain Business Modeler does not determine which data has changed. Instead, Supply Chain Business Modeler clears data from the destination model packages and dependent packages, and then imports Supply Chain Planning XML 2.0 data without determining which data has changed.

When Supply Chain Business Modeler clears data from a destination model package, it also clears data from any dependent packages. Dependent packages are data packages that are linked to other packages in a model. For example, the Sourcing Recommendations package in the Base model is dependent on the Distribution package because it refers to and must remain consistent with transport mode information in the Distribution package. If you run a full import scenario to import data in Supply Chain Planning XML 2.0 format into the Distribution package in the Base model, data from the Sourcing Recommendations package in the Base model is also cleared to prevent data inconsistencies.

To determine which related data is cleared if you run a full import of data in Supply Chain Planning XML 2.0 into a model package, you can review the following dependent package information, look in the schema viewer, or use the model dependentPackages command in the Business Modeler Shell.

In the Base model, all packages except the Calendar package are dependent on the Base package. The Calendar and Base packages are not dependent on any packages. The following diagram shows the remaining package dependencies in the Base model:



Base model dependencies

Each package in the Base model is represented by a rectangle in the diagram. If a package has dependent packages, the dependent packages appear to the right of the package and are connected by a line. Dependent packages can also be connected through lines to other packages. For example, in the Base model, the Sourcing Recommendations package is dependent on both the Distribution package and the Base package.

In the Operation model, all packages except the Calendar package are dependent on the Base package. The Calendar and Base packages are not dependent on any packages.

In addition, the Operation model has the following package dependencies:

- The SourcingRecommendations package is dependent on the Distribution package.
- The PurchaseOrderRecommendations and PurchaseOrders packages are dependent on the Supplier package.
- The ProductionSchedule package is dependent on the Manufacturing package.

In the Tactical model, all packages except the Calendar package are dependent on the Base package. The Calendar and Base packages are not dependent on any packages.

In addition, the Tactical model has the following package dependencies:

- The SourcingRecommendations package is dependent on the Distribution package.
- The PurchaseOrderRecommendations and PurchaseOrders packages are dependent on the Supplier package.

In the Strategic models, all packages except the Calendar package are dependent on the Base package. The Calendar and Base packages are not dependent on any packages.

In addition, the Strategic model has the following package dependencies:

- The SourcingRecommendations package is dependent on the Distribution package.
- The PurchaseOrders packages are dependent on the Supplier package.

In the Demand model, all packages are dependent on the Base package. The Base package is not dependent on any packages.

Setting Operation and Routing Display Names

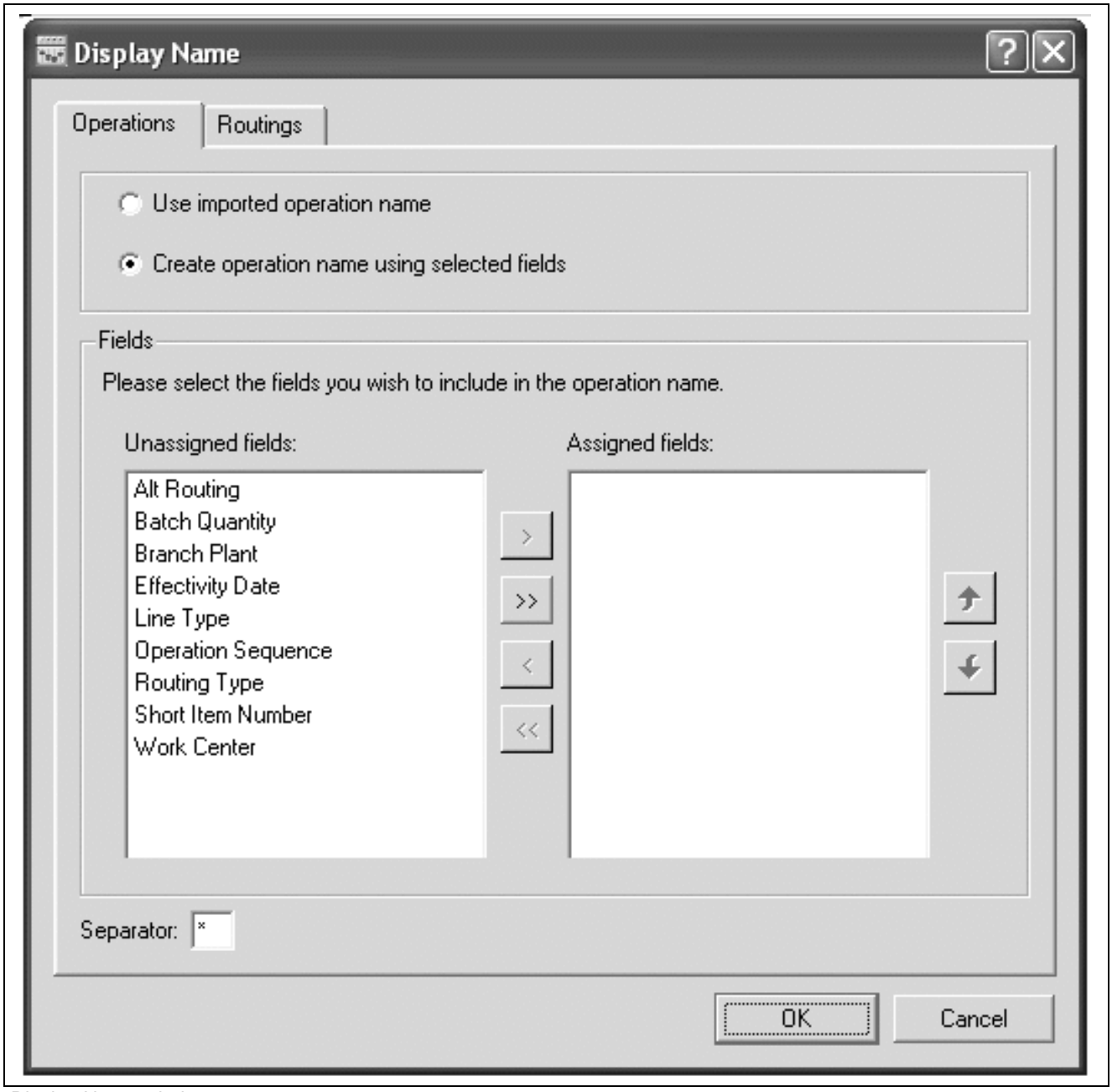
This section contains information about setting operation and routing display names from within the Supply Chain Business Modeler program.

Workspaces Used to Set Operation and Routing Display Names

Window Name	Navigation	Usage
Display Name	• From the Edit menu, select Configure Display Names. • Click the Operations tab.	Set operation display names upon import.
Display Name	• From the Edit menu, select Configure Display Names. • Click the Routings tab.	Set routing display names upon import.

Setting Operation and Routing Display Names

Access the Display Name window.



Display Name window

To set operation and routing display names:

1. Select the source of operation names when the extracts are imported from EnterpriseOne.

   Use imported operation name — Upon import, use the operation name specified in the XML Manufacturing package. If an operationName field is blank, the content from the operationCode field is used. If this option is chosen, the bottom portion of the Display Names window can no longer be accessed.

Create operation name using selected fields Format the operation name at the time of import, choosing the specific fields you want displayed, and their order.

2. If you have chosen to create the operation name using selected fields, choose the fields you want to include in the display and click the > button. The field names are moved to the right side of the window.
3. Specify the order of the fields using the up and down buttons.
4. Specify the separator between the fields.
5. Click the Routings tab.
6. Select the source of routing names when the extracts are imported from EnterpriseOne.

Use imported routing name Upon import, use the routing name specified in the XML Manufacturing package. If a routingName field is blank, the content from the routingCode field is used. If this option is chosen, the bottom portion of the Display Names window can no longer be accessed.

Create routing name using selected fields Format the routing name at the time of import, choosing the specific fields you want displayed and their order.

7. If you have chosen to create the routing name using selected fields, choose the fields you want to include in the display and click the > button. The field names are moved to the right side of the window.
8. Select the order of the fields by clicking on a field you want to move, and then use the up and down buttons to move the field to the correct position.
9. Specify the separator between the fields.
10. Click OK.

See “displayName configure”, Business Modeler Commands.

Creating, Modifying, and Deleting Import Scenarios

This section lists prerequisites and common elements, and discusses how to:

- Create import scenarios.
- Modify import scenarios.

Prerequisites

Before you complete the tasks in this section:

- Create an extract area and at least one data folder for saving data files that you want to import into Supply Chain Business Modeler.
- Save any Tcl scripts that you want to run as pre-import or post-import scripts in the Scripts folder of the extract area.

Common Elements Used in this Section

Data Folder Select the name of the data folder where the data package for importing is saved. Data packages that you import in an import scenario must be saved in one or more data folders in a single extract area.

Enterprise Data Select this option to display Enterprise Data package names in the data selection list. Packages in the Enterprise Data category include static data, such as supplier, customer, and distribution information.

Extract Area Enter the name of the extract area in which the data packages for importing are saved. If you have not created an extract area, the Extract Area field appears in red. You cannot create an import scenario until you create an extract area.

Full Import Select this option to create an import scenario in which Supply Chain Business Modeler imports full data packages in Supply Chain Planning XML format and assigns default values to some missing data values.

Incremental Import Select this option to create an import scenario in which Supply Chain Business Modeler updates existing data in a model with data from incremental data packages in Supply Chain Planning XML format.

Model Enter the name of the model that you want to populate with the imported data. You can import data from a supply chain management system into the Base model and from EnterpriseOne Supply Chain Planning applications into Operation, Tactical, Strategic, and Demand models.

If you want to import data into a planning model, the model must include the branch or branches that are included in the data; you cannot import data for branches that are not assigned to the planning model. For example, if you assign only Branch A to a planning model, you cannot import optimized plans for Branch B into the model.

Name Enter a name for the import scenario.

Optimized Plans Select this option to display Optimized Plans package names in the data selection list. Packages in the Optimized Plans category include supply chain plans and forecasts created by EnterpriseOne supply chain planning applications such as Production Scheduling, Production and Distribution Planning, Strategic Network Optimization, Demand Forecasting, and Demand Consensus.

Package Select this option to import all data packages in the data selection list in the export scenario.

Post-script Enter the name of a Tool Command Language (Tcl) script that you want to run after importing data. Post-import Tcl scripts enable you to call external processes that you would like to link to the data import, such as planning application solves and data extracts.

The Tcl script must be saved with the .tcl extension in the Scripts directory of the extract area used in the scenario, and cannot include Business Modeler commands.

Pre-script Enter the name of a Tcl script that you want to run before importing data. Pre-import Tcl scripts enable you to call external processes that you want to link to the data import, such as planning application solves and data extracts.

The Tcl script must be saved with the .tcl extension in the Scripts directory of the extract area used in the scenario, and cannot include Business Modeler commands.

Set all data folders to this default folder To import all data packages from the same data folder, select a data folder from this field. Data packages that you want to import in an import scenario must be saved in one or more data folders in a single extract area.

Transactional Data Select this option to display Transactional Data package names in the data selection list. Packages in the Transactional Data category include data that changes frequently, such as sales orders, transfer orders, and work orders.

Workspaces Used to Create, Modify, and Delete Import Scenarios

Window Name	Navigation	Usage
Add Import Scenario	• Click the Import Scenarios button in the Shortcuts bar. Select File, Add Import Scenario. • Click the Import Scenarios button in the Shortcuts bar. Right-click the Import Scenario Navigator and select Add Import Scenario. • In the Standard toolbar, click the Add Import Scenario button.	Create import scenarios.
Modify Scenario	Click the Import Scenarios button in the Shortcuts bar. Right-click the import scenario in the Import Scenario Navigator and select Modify Scenario.	Modify import scenarios.
Import Scenarios - Renaming	• Click the Import Scenarios button in the Shortcuts bar. Select the import scenario and select Edit, Rename. • Click the Import Scenarios button in the Shortcuts bar. Right-click the import scenario and select Rename.	Rename import scenarios.
Import Scenarios - Deleting	• Click the Import Scenarios button in the Shortcuts bar. Select the import scenario and select Edit, Delete. • Click the Import Scenarios button in the Shortcuts bar. Right-click the import scenario and select Delete.	Delete import scenarios.

Creating Import Scenarios

Access the Add Import Scenario workspace.

To create an import scenario:

1. In Introduction, click Next.
2. In Scenario Definition, complete the Name, Model, and Extract Area fields.
3. Select one of the following options:
   - Full Import
   - Incremental Import
4. Click Next.
5. (Optional) In Pre and Post Scripts, complete the Pre-script and Post-script fields.
6. Click Next.
7. In Data Selection, select one or more of the following options:
   - Enterprise Data
   - Transactional Data
   - Optimized Plans
8. Select either Package or the individual data packages that you want to import.
9. Complete the Set all data folders to this default folder field or complete the Data Folder field for each package.
10. Click Next.
11. In Summary, click Finish.

Modifying Import Scenarios

Access the Modify Scenario workspace.

To modify an import scenario:

1. In Scenario Definition, complete the Model and Extract area fields.
2. Select one of the following options:
   - Full Import
   - Incremental Import
3. Click Next.
4. (Optional) In Pre and Post Scripts, complete the Pre-script and Post-script fields.
5. Click Next.
6. In Data Selection, select one or more of the following options:
   - Enterprise Data
   - Transactional Data
   - Optimized Plans
7. Select either the Package option or the individual data packages that you want to import.
8. Complete the Set all data folders to this default folder field or complete the Data Folder field for each data package.

9. Click Next.
10. In Summary, click Finish.

Running Import Scenarios

This section provides overviews of running import scenarios and snapshot dates, lists prerequisites, and lists workspaces used to run import scenarios.

Understanding Running Import Scenarios

To import data into Supply Chain Business Modeler, you must run an import scenario. When you run an import scenario, Supply Chain Business Modeler imports data packages into a model, as specified in the import scenario.

When you run an import scenario, Supply Chain Business Modeler automatically detects whether the data files are compressed in gzip format or are uncompressed. If a data folder includes the same data package in both compressed and uncompressed formats, Supply Chain Business Modeler imports the uncompressed version of the package. For example, if a data folder includes both a Base.xml and a Base.xml.gz file, Supply Chain Business Modeler imports the Base.xml package.

When you run a full import scenario, the system imports full Supply Chain Planning XML data packages into the destination model. If you import full data packages in Supply Chain Planning XML 3.3, 3.2, 3.1 or 3.0 format, Supply Chain Business Modeler determines which enterprise and transactional data is different in the data packages than in the destination models. Supply Chain Business Modeler then imports only the new or changed enterprise and transactional data, deletes from the model any data that is not in the data packages, and assigns default values for required data that is missing from the data packages. If you import full data packages in Supply Chain Planning XML 2.0 format, Supply Chain Business Modeler does not determine which data is new or changed.

When you run an incremental import scenario, the system imports incremental Supply Chain Planning XML packages and merges the data with existing model data.

Supply Chain Business Modeler also runs any pre- or post-scripts that are specified in the import scenario.

The model, extract area, data folder, and data files specified in an import scenario must all exist before the import scenario can run successfully. If an import scenario does not run, and errors occur during the data import, you can view error information in the error log.

Each data package that you import must have the exact file name and structure required by Supply Chain Business Modeler. You must ensure that each Supply Chain Planning XML package has the correct structure and syntax and the appropriate level of granularity for the model into which you are importing data. For example, if you want to import data into the Strategic model, you must ensure that the data is in the correct Supply Chain Planning XML format for the Strategic level.

To ensure that a document conforms to the Supply Chain Planning XML format, you can validate XML documents using XSDs that are shipped with Supply Chain Business Modeler. Supply Chain Business Modeler is shipped with XSDs for documents in Supply Chain Planning XML format for each model level.

Note. The names of some Base model optimized plan data packages are different in Supply Chain Planning XML 3.3, 3.2, 3.1, and 3.0 format than in Supply Chain Planning 2.0 format because of changes associated with plan marshalling. As a result, you cannot import Supply Chain Business Modeler optimized plan packages in Supply Chain Planning XML 2.0 format into the Base model. Instead, you can import optimized plan packages in Supply Chain Planning XML 2.0 format into the Operation model.

After you import data into a model, you can view the model to determine how many data objects are populated in the model. You can also convert the data to the representation that is required for another Supply Chain Management or Supply Chain Planning system by populating the appropriate Supply Chain Business Modeler model with the data.

Note. When you import date-dependent data into Supply Chain Business Modeler, times in the data are interpreted as being in the time zone of the system where Supply Chain Business Modeler is running, even if the data was created on a system in a different locale.

See Also

"Setting Up Models," Obtaining Model Information

"Exporting Data from Supply Chain Business Modeler"

Understanding Snapshot Dates

When you run an import scenario, you must enter a snapshot date. The snapshot date is the date and time as of which all of the transactional data that you are importing is consistent and provides a synchronized view of the enterprise. For example, if a work order is partially completed as of the snapshot date, the appropriate inventory adjustments must also be made to reflect the consumption on the work order by the snapshot date.

The snapshot date that you specify is saved in the Model Parameters package in the model, and is populated to new models when the data is refreshed in or published to other models. The snapshot date is used in calculating the export horizon and planning horizon end dates when you export data from models.

Prerequisites

Before you complete the tasks in this section:

- Ensure that the model, extract area, data folder, and any Tcl scripts that are specified in the scenario exist.
- If you have renamed an extract area, folder, file, or script, ensure that the new name is specified in the scenario.
- Ensure that the Supply Chain Planning XML documents that you want to import into Supply Chain Business Modeler are saved in the extract area data folder that is specified in the import scenario.

Common Element in this Section

Snapshot Date Select the time and date as of which all transactional data is assumed to be consistent and for which a synchronized view of the enterprise exists.

Workspaces for Running Import Scenarios

Window Name	Navigation	Usage
Import Scenarios	• Click the Import Scenarios button in the Shortcuts bar. Select the import scenario that you want to run. Select File, Import. • Click the Models button in the Shortcuts bar. Right-click the import scenario that you want to run, and then select Import.	Run import scenarios.
Models	• Click the Models button in the Shortcuts bar. • Right-click the model that you want to import data into and select Import.	Run import scenarios from models.

CHAPTER 7

Setting Up Models

This chapter provides overviews of reference and planning models, model properties, and data packages, and discusses how to:

- Create, modify, and delete planning models.
- Set model export and plan marshalling horizons.
- Obtain model information.

Understanding Reference and Planning Models

Data is saved in models in the Supply Chain Business Modeler object database. Models are subsets of predefined data objects that you can populate with data from supply chain management and planning systems. Each model in Supply Chain Business Modeler represents a view of a supply chain with the granularity that is required for a specific supply chain management or planning purpose. To convert data to the representation that is required by a particular Supply Chain Management or Supply Chain Planning system, you can populate the appropriate model with data.

Supply Chain Business Modeler saves supply chain data in models in an object-oriented database. Each model in the Supply Chain Business Modeler database consists of data packages that provide a structure for supply chain data. Each data package contains related objects and attributes for saving supply chain data. You can populate each model structure with data from Supply Chain Management and Supply Chain Planning systems.

When you first start Supply Chain Business Modeler, five models appear in the Models workspace. These models are called reference models. Each reference model contains data for all branches of the supply chain at the level of granularity required for a particular supply chain management or planning purpose. Reference models toward the top of the Models workspace contain data for creating long-range plans, such as demand forecasts. Models toward the bottom of the workspace contain data for creating short-term, detailed plans, such as production schedules.

This table describes the reference models:

Reference Model	Description
Base	The Base model represents a supply chain at the finest level of granularity. You can import data from a supply chain management system into the Base model, and then populate other models with this data for use in supply chain planning. You can also export optimized plans from the Base model for further refinement or implementation using a Supply Chain Management system.

Reference Model	Description
Operation	The Operation model contains event-based data with a fine or high level of granularity. The data can be used to generate production sequencing, sequence-dependent changeovers, and work center synchronization using EnterpriseOne Supply Chain Planning products such as Production Scheduling and Production Scheduling - Process. Data in the Operation model includes information about specific transactions and orders, and is not summarized in time periods.
Tactical	The Tactical model contains data required for creating net production requirements, purchase order recommendations, and deployment plans using Production and Distribution Planning. Data in the Tactical model is summarized in daily time periods.
Strategic	The Strategic model contains data for creating sourcing guidelines, capacity requirements, and inventory build targets using EnterpriseOne Strategic Network Optimization. Data in the Strategic model is summarized in time periods. You can specify whether the data is saved in day, week, month, 4–week fiscal months, 4–4–5 week fiscal months, quarter, or year time periods.
Demand	The Demand model contains data for creating demand forecasts using EnterpriseOne Supply Chain Planning products such as Demand Forecasting and Demand Consensus. Data in the Demand model is summarized in time periods. You can specify whether the data is saved in day, week, month, 4–week fiscal months, 4–4–5 week fiscal months, or year time periods.

Data in a reference model usually represents the entire enterprise, and often includes information for multiple business units with different constraints and planning cycles. To plan separately for different branches and business units, you can divide the data from the Operation and Tactical models into planning models in Supply Chain Business Modeler. Each planning model can contain data for one or more branches in an organization, enabling you to import and export data for specific branches and create separate optimized plans for different parts of the organization.

Each planning model usually includes data required by a specific EnterpriseOne Supply Chain Planning system model. For example, at the Tactical model level, you could create a planning model that includes data for all branches in an apparel division and a planning model that includes data for all branches in a sporting goods division. You could then export the data from the planning models for use in separate Production and Distribution Planning repositories. At the Operation model level, you could create a planning model for each branch and export the data from each model for detailed scheduling in Production Scheduling.

Before you create a planning model, you must import branch data into the Base model so that Supply Chain Business Modeler can determine the available branch names. Data for a specific branch can appear in only one planning model at each reference model level. For example, you cannot assign the Toronto branch to more than one planning model at the Operation model level.

In the Models workspace, planning models appear to the right of the associated reference models.

To obtain information about individual models in Supply Chain Business Modeler , you can view data package, object, and attribute information in an HTML-based schema viewer that is provided with Supply Chain Business Modeler. You can also determine how many objects are populated in each model by viewing object counts in the Models workspace.

See Also

"Setting up Models," Obtaining Model Information

Understanding Model Properties

This section discusses:

- Time period patterns in the Tactical, Strategic, and Demand models.
- Calendar month and calendar quarter time periods.
- Export and planning horizons.
- Plan marshalling properties in the Base model.
- Demand History in the Demand model.
- Customer aggregation in the Demand model.

Time Period Patterns in the Tactical, Strategic, and Demand Models

Data is summarized in time periods in the Tactical, Strategic, and Demand models. The period start date and time are used as the identifier or name for each time period.

You must create time period patterns in the Tactical, Strategic, and Demand models before populating the models with data. To create a time period pattern, you must set the following model properties:

- Period type.

  The period type is the length of the time periods in which data is aggregated in a model. In the Tactical model, Day is the only possible period type. In the Strategic and Demand models, Day, Week, Month, Fiscal Month 4 Weeks, Fiscal Month 445, and Year time periods are available. In the Strategic model, Quarter time periods are also available.

- Alignment date.

  The alignment date is the date and time that fixes the starts of time periods in a model. The alignment date fixes all time period starts, except for with the Fiscal Month 445 period type, where it sets the start of 13–week 445 patterns.

Based on the period type and alignment date in a model, Supply Chain Business Modeler determines the pattern of time periods in the model. For example, if the period type in a model is Month and the alignment date is 2005–01–01T00:00:00, Supply Chain Business Modeler fixes month time periods to begin on the first day of each month at 00:00:00, such as 2005–02–01T00:00:00, 2005–03–01T00:00:00, and 2005–04–01T00:00:00.

When you refresh data in the Tactical, Strategic, or Demand model, data is aggregated into the time periods set in the model. If the snapshot date of the data in the source model lies within a time period, Supply Chain Business Modeler divides that time period into two periods: one starting at that period's start and ending at the snapshot date, and the other starting at the snapshot date and ending at that time period's end. For example, if day-long time periods in a model begin at midnight each day, and the model is refreshed with data that has a snapshot date of 2005–06–01T18:00:00, Supply Chain Business Modeler divides that day-long time period into two time periods: one beginning at 2005–06–01T00:00:00 and ending at 2005–06–01T18:00:00, and the other beginning at 2005–06–01T18:00:00 and ending at 2005–06–02T00:00:00. The remaining time periods in the model begin at midnight each day.

If you change the period pattern in a Tactical, Strategic, or Demand model after populating the model with data, data values are not automatically aggregated in the new period type. You must refresh data in a model to aggregate data in the model using the new time period pattern.

Tactical planning models use the same time period pattern that is set in the Tactical reference model. If you change the Tactical reference model period pattern and refresh data in the Tactical reference model, data in Tactical planning models remains aggregated in the original time period pattern until you refresh the planning model data.

In previous versions of Supply Chain Business Modeler, the period in the Strategic and Demand models was also used as the horizon period: the unit of measure for the export horizon and planning horizon. The period type is now set separately from the export horizon period unit, and can differ.

Period patterns and export horizons in models are also independent. The export horizon end date and time is determined by the snapshot date of data in a model and the export horizon duration set in the model.

When you export data from a Strategic or Demand model with data saved in Fiscal Month 445 periods, the time period pattern can change, depending on the start of the export horizon. For example, if you choose 445 periods in the Demand model and the alignment date is 2005–01–01T00:00:00, the first six time periods are:

- Four-week period starting 2005–01–01T00:00:00.
- Four-week period starting 2005–01–29T00:00:00
- Five-week period starting 2005–02–26T00:00:00
- Four-week period starting 2005–04–02T00:00:00
- Four-week period starting 2005–04–30T00:00:00
- Five-week period starting 2005–05–28T00:00:00

Because Supply Chain Business Modeler exports data for any time periods that start before the export horizon end, if the snapshot date is 2005–02–01T00:00:00, Supply Chain Business Modeler exports data starting with the four–week period starting 2005–01–29T00:00:00. The pattern of time periods in the exported data is then 4–5–4

Note. In the ModelParameters.xml package, the alignment date is the start date of the period that contains the snapshot date. If the period pattern in the model is Fiscal Month 445, the alignment date is the start of the first four-week fiscal month in a 445 series that contains the snapshot date. The alignment date in the ModelParameters package can differ from the alignment date in the Supply Chain Business Modeler GUI, but they generate the same time periods.

If you change the time in the alignment date in the Tactical reference model and then refresh data in the model, Supply Chain Business Modeler must refresh all enterprise and transactional data to aggregate it into the new time periods rather than refreshing only data that has actually changed. To ensure the best possible performance, do not change the alignment date time in the Tactical model frequently.

Calendar Month and Calendar Quarter Time Periods

Because not all calendar months have the same number of days, Supply Chain Business Modeler uses a special algorithm to create calendar month and calendar quarter time periods.

When you refresh the Strategic or Demand model with data, if the alignment date in the model is before the 28th day of a month, calendar month or calendar quarter time periods in the model begin on that day in each month. For example, if the alignment date is 2005-01-01T12:00:00, calendar month time periods in the Strategic model begin at 12:00:00 on the first day of each month, such as January 1, 2005, February 1, 2005, and March 1, 2005. Similarly, if the alignment date is 2005-01-27T06:00:00, calendar quarter time periods in the Strategic model begin at 6:00:00 on the 27th at the start of each quarter, such as January 27, 2005, April 27, 2005, and July 27, 2005.

However, if the alignment date in the model is after the 27th day of the month, Supply Chain Business Modeler determines how many days the alignment date is from the end of the month. Calendar month time periods in the model begin the same number of days before the end of each month. For example, the alignment date in the Demand model could be 2005-03-28T06:00:00. Because there are 31 days in March and the 28th is three days before the end of March, all other calendar month time periods in the model would begin three days before the end of each month, such as on April 27, 2005 and May 28, 2005 at 6 a.m.

Export and Planning Horizons

Before you can export data from a reference or planning model, you must specify an export horizon duration for the model. An export horizon duration is the length of time following the snapshot date for which data is exported from the model.

You must also specify a planning horizon duration for each model. The planning horizon duration is the length of time for which you want a Supply Chain Planning system to create optimized plans after you export data from the model. The planning horizon duration must be less than or equal to the export horizon duration.

Export and planning horizons for a model are measured in a single horizon duration unit.

See Also

"Exporting Data from Supply Chain Business Modeler," Understanding Running Export Scenarios

Plan Marshalling Properties in the Base Model

In the Base model, you must specify the lengths of time for which you want to include production schedules and net production requirements in the Base model marshalled plans. Plan marshalling durations ensure that you do not duplicate or double-count production requirements when you export data to a supply chain management system.

See Also

"Setting up Models," Understanding Model Properties

Demand Point History and Sales Order History Aggregation in the Demand Model

In the Demand model, you can specify the number of time periods prior to the snapshot date for which you want to populate demand history data in the Demand model with Sales Order history data when you refresh the Demand model.

Customer data from the Base model is aggregated into channels in the Demand model. Channels represent distribution methods for items from branches. If you do not select a Customer aggregation level, each channel in the Demand model includes data for a single customer. If you select a different hierarchy level, data is aggregated into channels at the level that you select.

Customer Aggregation in the Demand Model

In the Demand model, you can specify the Customer Aggregation level for aggregating customer data.

When you refresh data in the Demand model, customer data from the Base model is aggregated into channels and time periods in the Demand model. Channels represent distribution methods for items from branches. If you do not select a Customer aggregation level, each channel in the Demand model includes data for a single customer only. If you select a different hierarchy level, data is aggregated into channels at the level that you select.

Understanding Data Packages

This section discusses:

- Units of measure in the Base data package.
- Manufacturing data package.
- Optimized plans in the Base model.
- Customer, Item, and Branch Hierarchies.

Units of Measure in the Base Data Package

Unit of measure information is saved in the Base data package in all models. This table lists Base data package objects and the unit of measure information that they provide:

Object	Unit of Measure Information
Item	The Item object in the Base package of a model specifies the following unit of measure information for each item in a supply chain: • Planning unit of measure. For example, the Item object could indicate that the planning unit of measure for bicycle frames is each, or individual bicycle frames. • Weight of one planning unit. For example, the Item object could indicate that each bicycle frame weighs 10 pounds. • Volume of one planning unit. For example, the Item object could indicate that each bicycle frame has a volume of 10 cubic feet.

Object	Unit of Measure Information
Standard UOM	The Standard UOM object in the Base package of a model provides factors for converting measurements from one unit of measure to another unit that measures the same characteristic or unit type. For example, you could use factors from the Standard UOM object to convert item weights from pounds to kilograms or from litres to gallons. To calculate the amount of an item in the desired unit, you can multiply the amount in the original unit by the conversion factor. Factors in the Standard UOM table are not item-specific. For example, regardless of the item that you are weighing, a kilogram equals 2.2 pounds.
Item UOM	The Item UOM object in the Base package of a model provides factors for converting measurements between units that measure different characteristics or dimensions. For example, you could use factors from the Item UOM object to determine how many items are contained in 10 pounds of the item. To calculate the amount of an item in the desired unit, you can multiply the amount in the original unit by the conversion factor. The Item UOM object also specifies the characteristic or dimension that is measured by the target or desired unit of measure. Factors in the Item UOM table are item-specific. For example, the weight of one bicycle chain is different than the weight of one bicycle.

See Also

"Setting Up Models," Obtaining Model Information

Understanding the Manufacturing Data Package

The Manufacturing package in the Supply Chain Business Modeler data model provides information about routings, operations, and resources in a supply chain. You can use this information in EnterpriseOne Supply Chain Planning products such as Strategic Network Optimization, Production and Distribution Planning, and Production Scheduling to determine which routings, operations, resources, and items to use when you manufacture products.

To define routings for manufacturing items, Supply Chain Business Modeler uses the following three data objects:

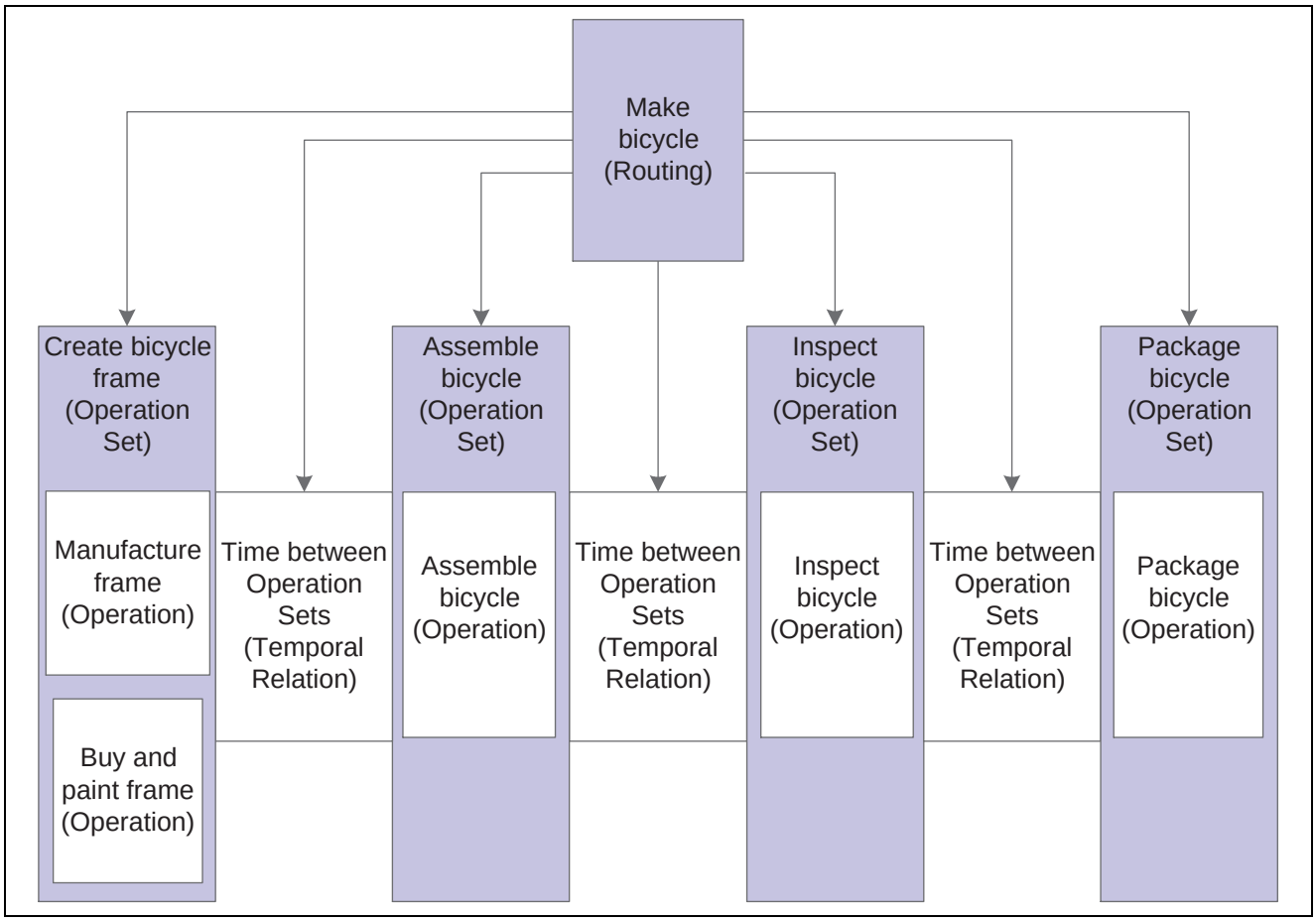
- Routing
- Operation Set
- Temporal Relation

However, these objects do not specify where the routings occur or which resources they use. Like a cake recipe, which provides steps for making a cake but allows you to choose where to make the cake and which bowl to use for mixing, these objects describe high-level manufacturing processes. You can use other EnterpriseOne Supply Chain Planning products to recommend which routings and resources to use to produce the required items.

The Routing object in Supply Chain Business Modeler specifies routings in a supply chain and refers to the Operation Set object to define the high-level processes in each routing. For example, the Routing object could define a routing called the Make Bicycle routing. The Operation set object could define the following high-level operation sets in the routing:

- Create bicycle frame
- Assemble bicycle
- Inspect bicycle
- Package bicycle

The following diagram provides a representation of this routing:



Routing representation

Because the Routing object refers to operation sets instead of specific operations, you can specify alternate operations that can be used to complete a manufacturing step in a routing. For example, the Create Bicycle Frame process can be completed using either the Manufacture Frame operation or the Buy and Paint Frame operation, whereas the Assemble Bicycle process can be completed only by using the Assemble Bicycle operation.

Note. The Manufacturing package also includes an object called Operation Group. While the Operation Set object is used to specify alternative operations for completing manufacturing steps, the Operation Group object is used to represent operation groups that you can use to create changeover patterns.

The Routing object also refers to the Temporal Relation object. The Temporal Relation object specifies the order of the operation sets in the routing, and the minimum and maximum amount of time between the operations. The Temporal Relation object can specify the time between operations as described in the following table.

Temporal Relation	Description
StartsAfterStart	The time between the operation sets is measured from the start of the first operation to the start of the second operation.

Temporal Relation	Description
StartsAfterEnd	The time between the operation sets is measured from the start of the first operation to the end of the second operation.
EndsAfterEnd	The time between the operation sets is measured from the end of the first operation to the end of the second operation.

Note. If a routing includes only one operation set, the secondOperationSetCode element in the Temporal Relation object should not be populated.

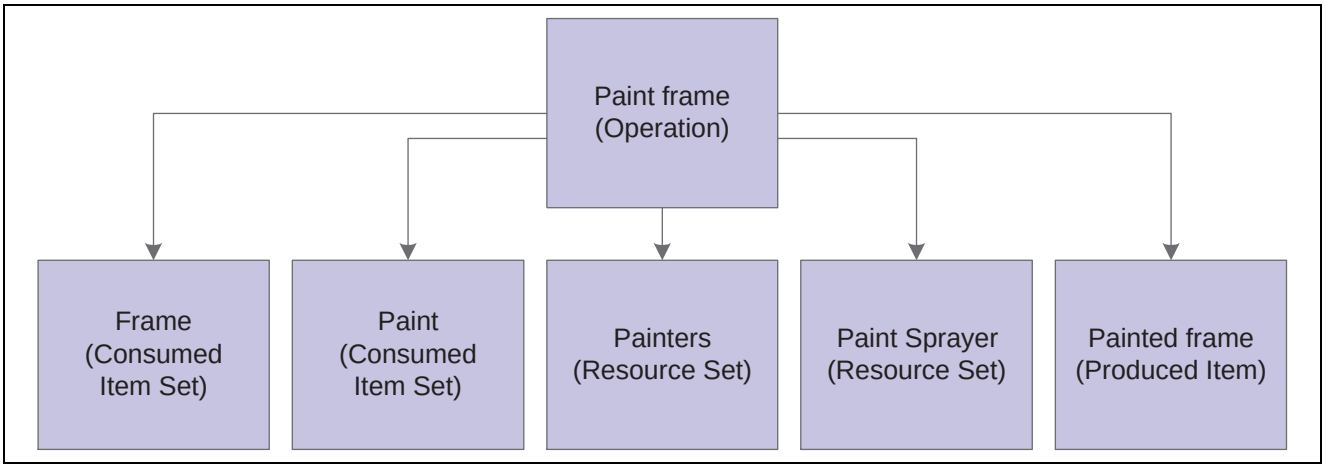
To define operations for manufacturing products, Supply Chain Business Modeler uses the following four data objects:

- Operation
- Consumed Item Set
- Resource Set
- Produced Item

These objects do not specify where the operations occur or which items and resources they use.

The Operation object in Supply Chain Business Modeler defines operations in a supply chain. Operations are not specific to a particular branch or resources, but they must produce the same item or items regardless of where the operation occurs and the resources and the items consumed in the operation.

For example, the following diagram shows an Operation object that defines the Paint Frame operation:



Operation representation

To define types of items and resources that are associated with an operation, the Operation object refers to the following three objects:

- Consumed Item Set.

  The Consumed Item Set object specifies the types of items that are consumed in specific operations. For example, the Consumed Item Set object indicates that the Paint Frame operation requires paint. It does not specify the brand of paint that must be used or whether paint is available at particular branches. It indicates only that something must be used as paint in the operation.

- Resource Set.

The Resource Set object specifies the types of resources that are required in the operation. For example, the Resource Set object indicates that the operation requires a paint sprayer. This object does not specify the exact paint sprayer that must be used or whether paint sprayers are available at particular branches. It indicates only that something must be used as a paint sprayer in the operation.

Note. The Manufacturing package also includes an object called Resource Group. Whereas the Resource Set object is used to specify types of resources that are required in operations, the Resource Group object is used to represent resource groups that you can use to create changeover patterns.

- Produced Item.

  The Produced Item object specifies the item or items produced by an operation. Produced items are listed in this object rather than grouped in a set so that Supply Chain Planning products can schedule the production of specific items. For example, the Produced Item object indicates that the produced item from the Paint Frame operation is a painted frame. This object does not provide any branch-specific information.

Because routings and operations are specified without reference to branches and specific resources and items, other items in the manufacturing package provide information about the availability of routings, operations, resources, and items at branches in a supply chain. Manufacturing package objects provide branch-specific information about the following elements of manufacturing processes:

Routings	The Branch Routing object specifies the times when routings are available at particular branches. It also specifies the overhead cost associated with routings at particular branches.
Operations	The Branch Operation object specifies the times when operations are available at particular branches. It also specifies the overhead cost associated with operations at particular branches. The Branch Operation Set object specifies the preferred operation in operation sets in routings at particular branches.
Consumed Items	The Consumed Item BOM object specifies items that can be used to fulfill a consumed item set role in an operation at a branch. For example, the Consumed Item BOM object could indicate that item 9797753 could be used as paint at one branch, while item 9797750 or item 9797754 could be used as paint at another branch. The Consumed Item BOM object also indicates when the item is available at a branch, and it provides information about how much of the item is consumed during the operation at the branch.

Resources	The Operation BOR object specifies the resources that can be used to fulfill a resource set role in an operation at a branch. For example, the Operation BOR object could indicate that two paint sprayers are available at the Cranston branch and three paint sprayers are available at the Gunnison branch. The Operation BOR object also indicates whether this is the preferred resource for an operation at this branch. The Crew object specifies the availability and costs associated with crew resources at specific branches. The Machine object specifies the availability and costs associated with machine resources at specific branches. The Tool object specifies the availability and costs associated with tool resources at specific branches.
Produced Items	The Produced Item BOM object provides production cost and yield information for items produced by operations at specific branches. Although an operation produces the same item or items at each branch, the production costs and yield of the item can vary from branch to branch.

To represent changeovers, Supply Chain Business Modeler uses the Changeover Pattern object. This object specifies the time and cost associated with changing a resource from one operation to another. When you publish data from Supply Chain Business Modeler workbench, the system populates the Changeover Pattern object with changeover patterns from the changeover editor.

To specify groups of resources or operations for defining a changeover, Supply Chain Business Modeler uses the Operation Group and Resource Group objects. When you publish data to the Base model from the Supply Chain Business Modeler changeover editor, Supply Chain Business Modeler populates the Operation Group and Resource Group objects with groups that you have created in the group editor.

See Also

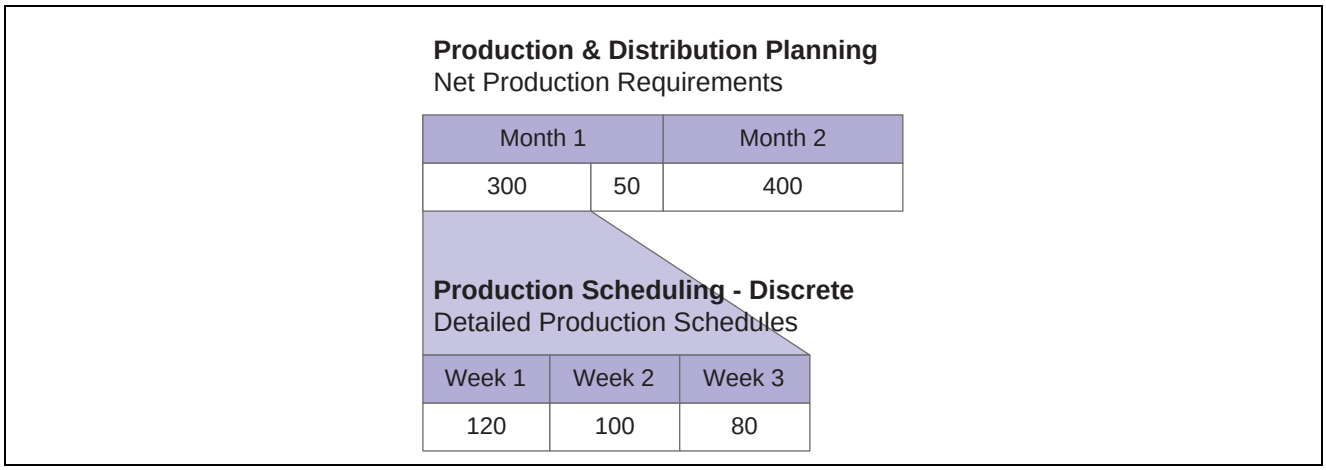
"Setting Up Models," Obtaining Model Information

Optimized Plans in the Base Model

Using Supply Chain Business Modeler and Supply Chain Planning products, you can create high-level production requirements, and then refine the plans to create detailed, short-term production schedules. The short-term production schedules specify how to meet demand that is required according to the net production requirements. If you send both plans to a supply chain management system, you will duplicate or double-count some production requirements in the supply chain management system.

For example, you could use Production and Distribution Planning to determine that you must produce 750 units of an item at a branch in the next two months, with 300 units to be produced during the first three weeks. You could then use Production Scheduling to determine how to produce the required 300 units at the branch during the first three weeks. If you send both the net production requirements and the detailed production schedules to a supply chain management system, the plans will indicate that you should produce 1050 units of the item at the branch in the next two months when you require only 750 units of the item. This problem occurs because the detailed production schedules duplicate 300 units of production that are also specified by the net production requirements.

The following diagram illustrates this example:



Production schedules duplicating net production requirements

To ensure that you do not duplicate or double-count production requirements in a supply chain management system, Supply Chain Business Modeler combines plans from various planning levels into a set of coherent or marshalled plans. When you publish data from the Operation model to the Base model, Supply Chain Business Modeler populates marshalled plan objects in the Base model with optimized plan data from the Operation model.

Before Supply Chain Business Modeler can create marshalled plans, you must specify the following:

- Short horizon duration.

  The short horizon duration is the amount of time following the snapshot date for which you want to include production schedules in the Base model marshalled plans. If you do not want to include production schedules in the Base model marshalled plans, you can enter 0 as the short horizon duration.

- Medium horizon duration.

  The medium horizon duration is the amount of time following the end of the short plan marshalling horizon for which you want to include net production requirements in the Base model marshalled plans.

When you publish optimized plan data from the Operation model, Supply Chain Business Modeler uses the plan marshalling horizons and optimized plan data from the Operation model to populate the following Base model packages:

- Detailed Production Plan.

  Supply Chain Business Modeler populates this package with routing instances and operation instances from the Production Schedule object in the Operation model. Routing instances and all their operations are included in the Detailed Production Plan if they have at least one operation instance that starts before the end of the short plan marshalling horizon. Pegging Demand Item and Pegging Detail are not included in the Detailed Production Plan package.

- Master Production Plan.

  Supply Chain Business Modeler populates this package with net production requirements from the Operation model that are not scheduled to be produced by operation instances in the Detailed Production package and that have requested dates before the end of the medium plan marshalling horizon.

  Supply Chain Business Modeler computes the item quantities that are scheduled to be produced by operation instances in the Detailed Production Plan package. Supply Chain Business Modeler then subtracts the scheduled item amounts from the net production requirements in the Operation model that have requested dates before the end of the medium plan marshalling horizon. The remaining net production requirements are populated in the Master Production Plan object.

- Purchase Plan.

  The Purchase Plan package contains all plans from the Purchase Order Recommendations package in the Operation model.

- Deployment Plan.

  The Deployment Plan package contains plans from the Net Deployment Requirements package in the Operation model.

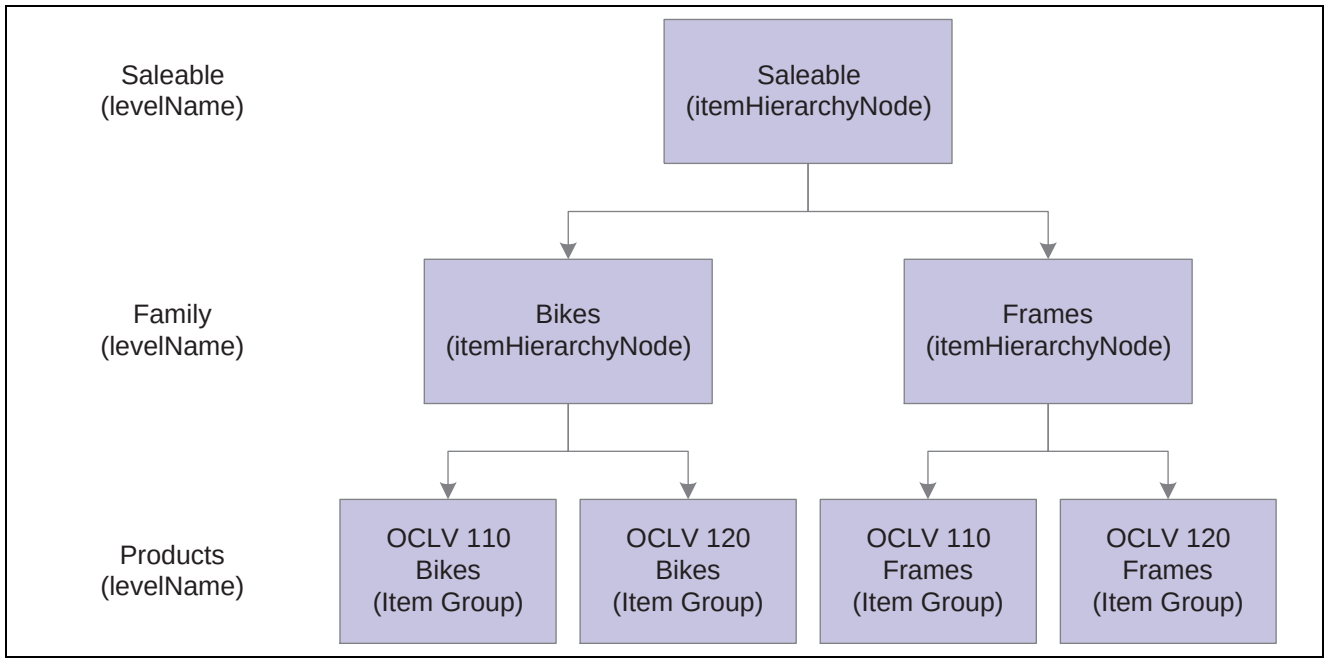
See Also

Setting Plan Marshalling Horizons

Customer, Item, and Branch Hierarchies

Supply Chain Business Modeler includes Customer, Item, and Branch Hierarchy Level objects. These objects are used to group related customers, items, and branches in nested subsets.

Item, Branch, and Customer Hierarchy Level objects include levelName elements. These elements list levels in the hierarchy. For example, in this diagram, levelName attribute values from the top to the bottom of the item hierarchy are Saleable, Family, and Products:



Sample item hierarchy

Customer, Item, and Branch Hierarchy Level objects also include refinement elements, which specify the level that is immediately below each level in a hierarchy. include refinement elements. For example, in the diagram shown, the refinement of the Saleable level is Family, and the refinement of the Family level is Products. There is no refinement element for the Products level, because there are no lower levels in the hierarchy.

The lowest level in a Customer, Item, or Branch hierarchy includes customer, item, or branch groups. For example, in the diagram shown, the Products level is the lowest level in the hierarchy, and includes four item groups: OCLV 110 Bikes, OCLV 120 Bikes, OCLV 110 Frames, and OCLV 120 Frames. The members of customer, item, and branch groups are specified in the Customer Group, Item Group, and Branch Group objects.

Higher levels in a Customer, Item, or Branch hierarchy include hierarchy nodes. For example, in the item hierarchy shown in the previous diagram, the Family level includes two item hierarchy nodes: Bikes and Frames. Each node includes a set of nodes or groups in the level below them in the hierarchy. For example, in the previous diagram, the Bikes itemHierarchyNode element includes OCLV 110 Bikes and OCLV 120 Bikes, while the Frames itemHierarchyNode element includes OCLV 110 Frames and OCLV 120 Frames. Similarly, the Bikes node at the Saleable level includes the Bikes and Frames nodes.

Customer hierarchy information supports aggregation in Demand Consensus and Demand Forecasting. Using customer hierarchy information, customer data is aggregated into channels when you refresh data in the Demand model. In the Demand model, itemPrice is moved from the Customer package to the Base package.

If you do not aggregate customer data to a higher level in the Demand model, Supply Chain Business Modeler populates location information in the Channel object in the Demand model when you refresh the model with data. If you aggregate the data, Supply Chain Business Modeler does not populate location information in the Channel object in the Demand model.

Note. The Customer package in the Demand model is deprecated and will be removed in future releases. Do not use the Customer package in new implementations.

See Also

Setting the Customer Aggregation Level in the Demand Model

Creating, Modifying, and Deleting Planning Models

This section lists prerequisites, common elements, and workspaces used to create, modify, and delete planning models.

Prerequisite

Create and run an import scenario to import branch data into the Base model in Supply Chain Business Modeler.

Common Elements Used in this Section

Assigned Branches	Lists branches that are assigned to the planning model. To remove a branch from the planning model, click the branch name and then click the left arrow button.
Assigned to	Specifies the planning model that the branch is assigned to.
Model name	Every model in Supply Chain Business Modeler must have a unique name.
Unassigned Branches	Lists branches that are not assigned to a planning model. To assign a branch to the planning model, click a branch and then click the right arrow button.

Workspaces Used to Create, Modify, and Delete Planning Models

Window Name	Navigation	Usage
New Planning Model	• Click the Models button in the Shortcuts bar. Click the Tactical or Operation model, and then select File, Add Planning Model. • Click the Models button in the Shortcuts bar. Right-click the Tactical or Operation model, and select Add Planning Model.	Create planning models.
Planning Model - Properties	Click the Models button in the Shortcuts bar. Right-click the planning model that you want to modify, and select Properties.	Modify planning models.
Model Partition	Click the Models button in the Shortcuts bar. Right-click the Operation or Tactical model and select Partition Branches.	Change branches assigned to multiple planning models.
Models - Deleting	• Click the Models button in the Shortcuts bar. Select the planning model and select Edit, Delete. • Click the Models button in the Shortcuts bar. Right-click the planning model and select Delete.	Delete planning models.

Setting Model Properties

This section lists prerequisites and windows used to set model properties, and discusses how to:

- Set export and planning horizons.
- Set plan marshalling horizons.
- Set the demand history in the Demand model.
- Set the customer aggregation level in the Demand model.

Prerequisite

Import the Customer package into the Base model before setting the customer aggregation level in the Demand model.

Windows Used to Set Model Properties

Window Name	Navigation	Usage
Horizons	Click the Models button in the Shortcuts bar. Right-click the model for which you want to specify an export or planning horizon and select Properties.	Set export and planning horizons.
Period Pattern	Click the Models button in the Shortcuts bar. Right-click the Tactical, Strategic, or Demand model and select Properties.	Set the period pattern in the Tactical, Strategic, or Demand model.
Plan Marshalling Properties	Click the Models button in the Shortcuts bar. Right-click the Base model, and select Plan Marshalling.	Set plan marshalling horizons.
Demand Properties - Horizon	Click the Models button in the Shortcuts bar. Right-click the Demand model and select Properties. In the Properties dialog box, click the Horizon tab.	Set demand history duration in the Demand model
Demand Properties - Aggregation	Click the Models button in the Shortcuts bar. Right-click the Demand model and select Properties. In the Properties dialog box, click the Aggregation tab.	Set customer aggregation level in the Demand model.

Setting Export and Planning Horizons

Access the Model - Properties window.

Planning Horizon Duration	Specifies the length of time following the snapshot date for which you want a Supply Chain Planning system to create optimized plans using the data after you export data from the model. The export horizon is measured in the Duration Unit.
Export Horizon Duration	Specifies the length of time following the snapshot date for which data will be exported from the model when you export the data. The export horizon is measured in the Duration Unit. If you export data from a Tactical, Strategic, or Demand model, Supply Chain Business Modeler exports data for all time periods that overlap the export horizon.
Duration Unit	Specifies the time unit of measure for the export horizon and planning horizon.

See Also

"Setting up Models"

"Exporting Data from Supply Chain Business Modeler"

Setting Plan Marshalling Horizons

Access the Plan Marshalling Properties window.

Short Duration and duration unit	Specifies the amount of time following the snapshot date for which you want to include routing instances and operation instances in the Detailed Production Plan.
Medium Duration and duration unit	Specifies the amount of time following the end of the short plan marshalling horizon for which you want to include net production requirements that have not been scheduled in the Master Production Plan.

Setting the Demand History in the Demand Model

Access the Demand Model - Horizon window.

Demand History Number of Periods	Specifies the number of periods prior to the snapshot date for which you want to populate demand history data in the Demand model.

Setting the Customer Aggregation Level in the Demand Model

Access the Demand Model - Aggregation window.

Customer Aggregation Level	Specifies the level at which you want to aggregate customer data in channels in the Demand model.

See Also

Understanding Customer, Item, and Branch Hierarchies

"Refreshing Data in Models"

Obtaining Model Information

This section provides an overview of obtaining model information and lists the workspaces used to obtain model information.

Understanding Obtaining Model Information

To obtain information about models in Supply Chain Business Modeler, you can use the schema viewer that is provided with Supply Chain Business Modeler or view model structures in the models workspace.

The schema viewer is a series of HTML pages that displays data package information for reference and planning models in Supply Chain Business Modeler. The schema viewer provides information about all data objects and attributes in each model package, including data types and default values. The schema viewer also lists the dependent packages for each data package.

In addition, you can use the Supply Chain Business Modeler Models workspace to determine how many objects are populated in each model.

Note. For performance reasons, some objects with invalid references are not deleted until they are exported or refreshed in a connected reference model. As a result, the number of objects listed for a model in the Models workspace can differ from the number of objects that would be exported from the model or refreshed in a connected reference model.

Workspaces Used to Obtain Model Information

Window Name	Navigation	Usage
Schema viewer in Windows	Open the following file, where *path* is the directory where Supply Chain Business Modeler is installed, in Internet Explorer: *path*\scp\8.12.1\scbm\docs\schema\main.html	View data package information.
Schema viewer in UNIX	Open the following file, where *path* is the directory where Supply Chain Business Modeler is installed, in Netscape Navigator: *path*/scp/8.12.1/scbm/docs/schema/main.html	View data package information.
Models	Click the Models button in the Shortcuts bar. Click a model.	View object counts.

Viewing Data Package Information

Access the Schema Viewer workspace in Windows or UNIX.

Click the links for the data objects and attributes for which you want information.

The following table explains symbols used in the Schema Viewer workspace:

Schema Viewer Symbol	Description
Pk	Primary key. This attribute is used to uniquely identify the object.
Fx	Fixed attribute. The attribute value does not vary with the effective date.
Tv	Time-varying attribute. The value of the data varies with an effective date.
Bk	A value for the attribute was calculated for the time period.
Object Reference	Pointer to another object. Click the link to review the reference object.
Primitive	Attribute that originates in this object.
R	Required attribute. A value must exist for the attribute.

Schema Viewer Symbol	Description
*	Repeated value. Multiple values are allowed for this attribute.
indented object name	Objects that are indented in the overview page are nested in data files in Supply Chain Planning XML 3.3, 3.2, 3.1, and 3.0 format.

CHAPTER 8

Creating Calendars That Model Resource Availability

This chapter provides overviews of calendars and the default calendar and discusses how to:

- Create and manage calendars and components.
- Create and manage events and event patterns.
- Manage components in calendars.
- Publish calendars to the Base model.

Understanding Calendars

Supply Chain Business Modeler includes a calendar editor that enables you to model the availability of resources in a supply chain by:

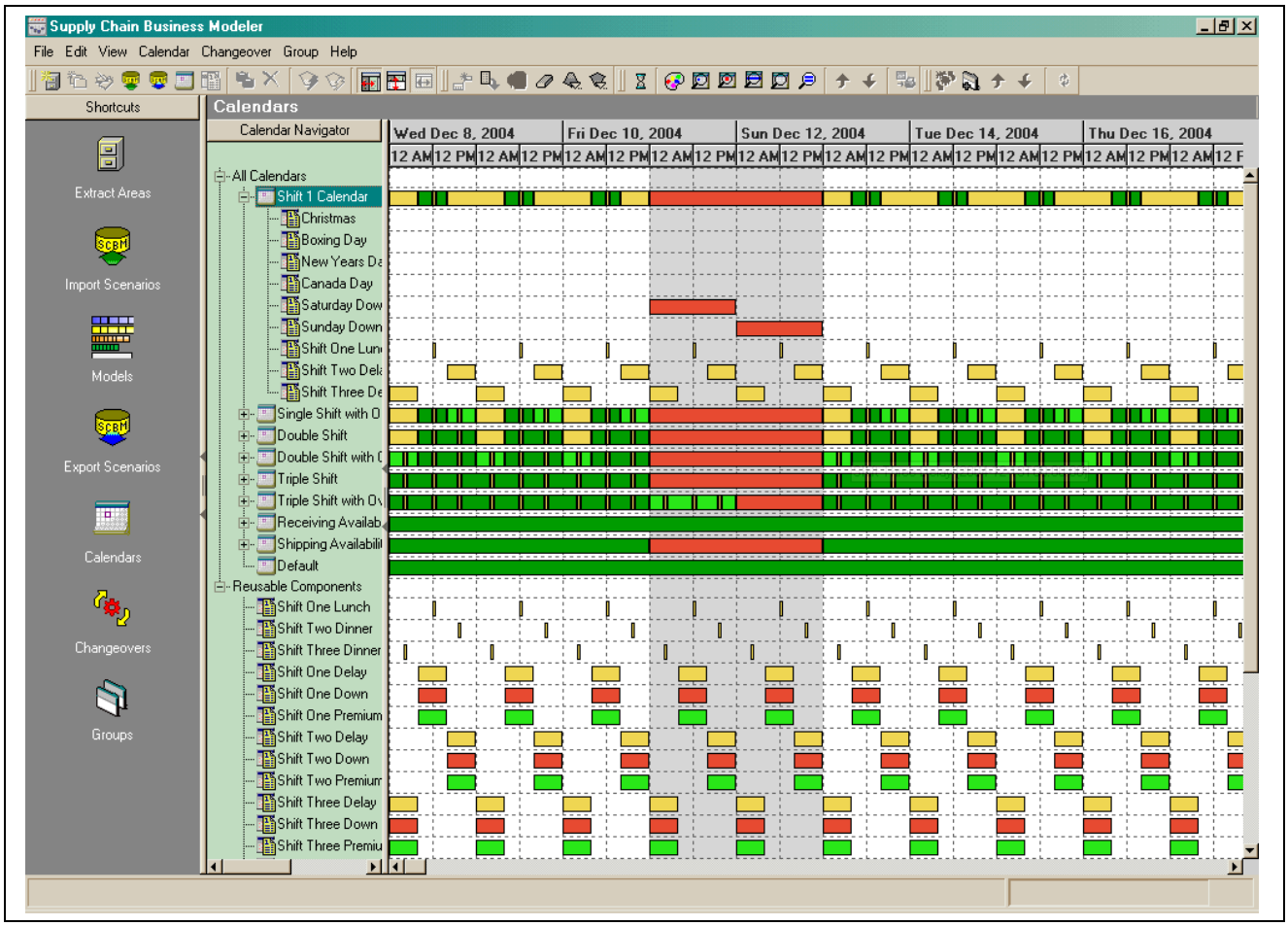
- Creating calendars.
- Adding components, events, and event patterns that represent capacity and availability patterns for resources in the supply chain.
- Publishing calendars to the Base model.

When you create a new calendar in Supply Chain Business Modeler, the calendar includes only uptime. The term uptime refers to time when resources are available. You can add components, events, and event patterns that represent downtime, delay time, and premium time.

Components are timelines that can contain an event pattern or one or more events. After you create a component, you can add one or more events or an event pattern to indicate the availability of resources in a supply chain or to indicate whether they cost extra to use. Events represent nonrecurring times when resources in a supply chain are available, unavailable, or cost extra to use. Event patterns represent recurring times when resources are available, unavailable, or cost extra to use. You cannot add events or event patterns directly to calendars. Instead, you must add events and event patterns to components that can be used in Supply Chain Business Modeler calendars.

When you add components to a calendar, the calendar timeline shows the pattern of resource availability and unavailability that results from the components and events in the calendar. Different colors in the calendar timelines represent events when resources are available, are unavailable, or cost extra to use.

The following example shows calendars, components, events, and event patterns in the calendar editor:



Calendar editor

The calendar editor includes the Calendar Navigator pane and an area that shows the timeline of events for each calendar and component. The calendars and components that you create are listed in the Calendar Navigator. In this example, there are nine calendars, including the Default calendar.

If a calendar includes events in separate components that are scheduled for the same time, events in components that are highest in the calendar list have priority over events in components that are lower in the calendar list. For example, in the Shift 1 Calendar list, the Saturday Down component appears before the Shift 1 Lunch component. As a result, the downtime events in the Saturday Down component appear in the Shift 1 calendar timeline even though Shift 1 Lunch component events are scheduled for the same time. To change which events appear in the calendar timeline, you can change the order of the components in the calendar list.

When you first create a calendar in Supply Chain Business Modeler, it is not saved in the Supply Chain Business Modeler database and it is not associated with any objects in the Supply Chain Business Modeler database. After you create a calendar, you can publish the calendar to the Base model and transform the data for use in supply chain planning and management systems. You can use the associated calendars in Supply Chain Business Modeler or in calculations outside of Supply Chain Business Modeler to determine the availability of the resources. For external functions, you can export the resulting calendar into Production Scheduling, or use the calendar when abstracting resource availability for data models where data is summarized in time periods.

In addition to adding components to calendars in Supply Chain Business Modeler, you can create reusable components that you can add to one or more calendars.

You can create a component in Supply Chain Business Modeler by adding a new component or by duplicating an existing component. If you create a new component, the component does not include any events and you must add one recurring or one or more one-time events to indicate when resources are unavailable, are available, or cost extra to use. If you duplicate a component, the new component has the same characteristics and events as the original component.

Note. When you use the Supply Chain Business Modeler graphical user interface, you must create a component before you can add an event pattern to the component. However, using a single Business Modeler Shell command, you can create components and their event patterns simultaneously.

When you use the Supply Chain Business Modeler graphical user interface, you cannot copy a component to another calendar if it is already attached to a calendar. However, by using Business Modeler Shell commands, you can copy calendar components to other calendars.

Understanding the Default Calendar

To ensure that EnterpriseOne Supply Chain Planning products create accurate production schedules, each crew, machine, and tool in a Supply Chain Business Modeler model must have an associated calendar that indicates when the resource is available, unavailable, or costs extra to use. To ensure that a calendar is always available to associate with resources, Supply Chain Business Modeler includes a calendar called Default. When you import resource data in Supply Chain Planning XML format into Supply Chain Business Modeler, the system assigns Default as the availabilityCalendar attribute value for any crew, machine, or tool that has no availabilityCalendar value specified in the XML data.

Initially, the Default calendar appears in the calendar editor and represents constant up time. You cannot delete the Default calendar from the calendar editor, rename the Default calendar, or create another calendar in the calendar editor with the name Default. However, you can change the Default calendar in the editor by adding down, delay, premium, and up time events. For example, if resources in a supply chain are unavailable on weekends, you can add downtime events on all weekends in the Default calendar.

When you first import data into the Base model, the system ensures that a Default calendar is populated in the Base model. If the data that you are importing includes a Calendar.xml file with a calendar named Default, the Default calendar from the XML data is populated in the Base model. However, if you do not import a Default calendar, the Default calendar from the calendar editor is published to the Base model.

After the Default calendar is populated in the Base model, it cannot be deleted. Even if you clear the Base model, the Default calendar remains. However, you can change events in the Default calendar in the Base model by importing a Calendar.xml file with different Default calendar events or by revising the Default calendar in the calendar editor and publishing it to the Base model.

To populate other models with the Default calendar from the Base model, you can refresh the data in connected reference and planning models.

If you clear the Base model, the Default calendar remains. If you clear any other model that is populated with the Default calendar, the Default calendar is cleared from the model.

Note. Although you cannot create another calendar called “Default”, you can create a calendar with the name “default” with different capitalization. For example, you can create calendars named “DEFAULT”, “default”, or “DefaulT”.

Creating and Managing Calendars and Components

This section lists the workspaces used to create and manage calendars and components.

Workspaces Used to Create and Manage Calendars and Components

Window Name	Navigation	Usage
Calendar Navigator - Add Calendar	• Click the Calendars button in the Shortcuts bar. Select File, Calendar, Add Calendar. • Click the Calendars button in the Shortcuts bar. Right-click All Calendars and select Add Calendar.	Create calendars.
Calendar Navigator - Add Calendar Component	• Click the Calendars button in the Shortcuts bar. Select a calendar and select File, Calendar, Add Component. • Click the Calendars button in the Shortcuts bar. Right-click the calendar name and select Add Component.	Create calendar components.
Calendar Navigator - Add Reusable Component	• Click the Calendars button in the Shortcuts bar. Select Reusable Components and select File, Calendar, Add Component. • Click the Calendars button in the Shortcuts bar. Right-click Reusable Components and select Add Component.	Create reusable components.
Calendar Navigator - Duplicate	• Click the Calendars button in the Shortcuts bar. Select a calendar name and select Edit, Duplicate. • Click the Calendars button in the Shortcuts bar. Right-click a calendar name and select Duplicate.	Duplicate calendars.
Calendar Navigator - Rename	• Click the Calendars button in the Shortcuts bar. Select a calendar name and select Edit, Rename. • Click the Calendars button in the Shortcuts bar. Right-click a calendar name and select Rename.	Rename calendars.

Page Name	Navigation	Usage
Calendar Navigator - Delete	• Click the Calendars button in the Shortcuts bar. Click a calendar name, and select Edit, Delete. • Click the Calendars button in the Shortcuts bar. Right-click a calendar name, and select Delete.	Delete calendars.

Creating and Managing Events and Event Patterns

This section provides an overview of events and event patterns, lists common elements, and lists the workspaces used to create and manage events and event patterns.

Understanding Events and Event Patterns

Events and event patterns can represent the following four time types:

Downtime	Downtime events represent times when a resource is unavailable. Manufacturing operations can occur before or after a downtime event, but cannot span a downtime event. An example of a downtime event is a plant closure where manufacturing operations cannot start before and end after the weekend.
Delay time	Delay time events represent times when a resource is unavailable. Manufacturing operations can begin before a delay time event and finish after the delay time event. That is, operations can begin before a delay time event and finish after the event. An example of a delay time event is a crew's lunch hour where manufacturing operations can start before and end after the hour.
Uptime	Uptime events represent times when resources are available. An example of an uptime event is the standard work hours in the regular day-to-day operation of a plant.
Premium time	Premium-time events represent time for which added costs apply. An example of a premium-time event is an overtime shift for which workers receive higher wages.

Note. When you use Supply Chain Business Modeler graphical user interface, you must create a component before you can add an event pattern to the component. However, using a single Business Modeler Shell command, you can create components and their event patterns simultaneously.

An event pattern represents recurring times when resources are available or unavailable, or cost extra to use. An event represents a nonrecurring time when resources are available or unavailable, or cost extra to use.

You cannot add events or event patterns directly to calendars. Instead, you must add events and event patterns to components that can be used in calendar or reusable components.

You can add one or more events to any component that does not contain an event pattern. You cannot add an event or event pattern to a component that already includes an event pattern or one or more events. Each event pattern must appear in its own component. You can add an event pattern only to an empty component.

Note. You cannot add an event to a component if the time of the event overlaps the time of an event that already appears in the component. To create an event that occurs at the same time as another event, you must add the event to another component.

You cannot change the properties of a single event occurrence in an event pattern; you must change the properties of the entire event pattern.

When you change the properties of an event pattern that is in a reusable component, the changes do not appear in calendars where you have already added the component. If you want the revised reusable component to appear in a calendar, you must delete the component in the calendar and copy the revised component to the calendar.

You can delete a one-time event from a component. When you delete a one-time event from a component, you do not delete the component or any other one-time events within the component.

Note. You cannot delete an event pattern from a component. To delete an event pattern, you must delete the component in which the event pattern appears.

Common Elements Used in this Section

Daily When creating an event pattern, select to indicate that the event occurs each day.

Delay Select to indicate that the event represents delay time. Delay time events represent times when a resource is unavailable. Manufacturing operations can begin before a delay time event and finish after the event. An example of delay time is a crew's lunch time.

Down Select to indicate that the event represents downtime. Downtime events represent times when a resource is unavailable. An example of downtime is a plant closure over a weekend. Manufacturing operations can occur before or after a downtime event, but they cannot span a downtime event.

Duration Specify the length of the event. The system automatically calculates the end date and time based on the start time and duration.

Duration unit of measure Enter the time unit for measuring the duration.

End after When creating an event pattern, select this option to indicate that the event occurs a specified number of times and then stops. After selecting this option, type the number of occurrences in the Occurrences field.

End by When creating an event pattern, select this option to indicate that the event stops recurring after a specified date. After selecting this option, specify an end date and end time.

Monthly Select to indicate that the event occurs each month.

No end date Select to indicate that the event recurs indefinitely.

Premium Select to indicate that the event represents premium time. Premium-time events represent time for which added costs apply. An example of a premium-time event is an overtime shift for which workers receive higher wages.

Start Date Enter the start date of the event.

Start Date time Enter the start time of the event.

Uptime Select to indicate that the event represents uptime. Uptime events represent normal operational times when resources are available. An example of an uptime event is the standard work hours in the regular day-to-day operation of a plant.

Weekly Select to indicate that the event occurs each week.

Yearly Select to indicate that the event occurs each year.

Workspaces Used to Create and Manage Events and Event Patterns

Window Name	Navigation	Usage
Add Event Pattern	Click the Calendars button in the Shortcuts bar. Right-click a component and select Add Pattern.	Create event patterns in components.
Add Event	Click the Calendars button in the Shortcuts bar. Right-click a component and select Add Event.	Add events to components.
Component Properties	Click the Calendars button in the Shortcuts bar. Right-click a component and select Properties.	Modify event pattern properties.
Event Properties	Click the Calendars button in the Shortcuts bar. Right-click a component and select Properties.	Modify event properties.

Managing Components in Calendars

This section provides an overview of managing components in calendars and lists workspaces used to manage components in calendars.

Understanding Managing Components in Calendars

When you add a component to a calendar, the component appears at the bottom of the list of components in the calendar. The effect of events in the component appears in the calendar timeline. After you add a component, you can change the priority of components in the calendar.

In Supply Chain Business Modeler, calendars appear in a tree structure in the Calendar Navigator. The Calendar workspace shows the timeline and pattern of events in each calendar and component in the list. When you add components to a calendar, the calendar timeline shows the pattern of resource availability and unavailability that results from the components and events in the calendar. Different colors in the calendar timelines represent times when resources are available, are unavailable, or cost extra to use.

After you create a reusable component and add events or an event pattern, you can add the component to one or more calendars. When you add a reusable component to a calendar, a copy of the component is created in the calendar. This copy is no longer associated with the original component. If you make changes to a reusable component, the changes do not appear in calendars to which you have previously added the component. To have the changed component appear in a calendar, you must add the reusable component to the calendar. If a copy of the component already appears in the calendar, you must delete the component copy before you can add the revised component to the calendar. If you rename a reusable component, it retains its old name in any calendar to which you previously added the component. Similarly, if you delete a reusable component, it is not removed from any other calendars in which it resides.

The events that appear in a calendar result from the events that are defined in its components. Events in components that appear higher in the list of components in a calendar in the Calendar Navigator have priority over events in components that are lower in the list. If two components include events at the same time, the event in the higher component in the list in the Calendar Navigator takes priority and appears in the calendar timeline. For example, if a downtime event appears in the first component in a calendar list at the same time as an uptime event in the last component in the calendar list, the downtime event appears in the calendar timeline.

To change which events appear in a calendar, you can change the order of the components in the calendar list within the Calendar Navigator.

If multiple events are scheduled for the same time in separate components in a calendar, events in components that are highest in the calendar list have priority over events in components that are lower in the calendar list.

Note. If events from a component do not appear in the rolled up calendar timeline, ensure that you have correctly prioritized the components in the calendar.

See Also

Understanding Calendars

Workspaces Used to Manage Components in Calendars

Window Name	Navigation	Usage
Calendar Navigator	Click the Calendars button in the Shortcuts bar. Drag the component name to the calendar name.	Add reusable components to calendars.
Calendar Navigator - Raise Priority	• Click the Calendars button in the Shortcuts bar. Right-click the calendar component and select Raise Priority. • Click the Calendars button in the Shortcuts bar. Select the calendar component and select Calendar, Raise Priority.	Assign higher priorities to calendar components.
Calendar Navigator - Lower Priority	• Click the Calendars button in the Shortcuts bar. Right-click the calendar component and select Lower Priority. • Click the Calendars button in the Shortcuts bar. Select the calendar component and select Calendar, Lower Priority.	Assign lower priorities to calendar components.

Publishing Calendars to the Base Model

This section provides an overview of publishing calendars to the Base model and discusses how to publish a calendar to the Base model.

Understanding Publishing Calendars to the Base Model

When you first create a calendar that represents the availability of resources in a supply chain, it is not saved in a model. To save calendar data in a model, you must publish the calendar.

When you publish a calendar, the list of events in the calendar is copied to the Base model. The calendar object created in the Base model has the same name as the calendar in the Calendar Navigator, and it has the same list of events that appears in the calendar timeline. If a calendar with the same name already exists in the Base model, the system replaces the existing calendar with the calendar that you are publishing to the model.

Workspace Used to Publish Calendars

Window Name	Navigation	Usage
Publish Calendar	Click the Calendars button in the Shortcuts bar. Right-click the calendar that you want to publish and select Publish Calendar.	Publish a calendar to the Base model.

Publishing Calendars

Access the Publish Calendar workspace.

Start date time	Specify the start date and time of events in the calendar that you want to publish to the Base model.
Calendar duration	Specify the length of time in the calendar for which you want to publish events to the Base model.

CHAPTER 9

Creating Changeover Patterns and Groups

This chapter provides overviews of changeovers and groups, populating the changeover and group editors with data, creating and managing resource and operation groups, creating and managing changeover patterns, changeover pattern priorities, and publishing changeover patterns and groups to the Base model, and discusses how to:

- Populate the changeover and group editors with data.
- Create and manage groups and changeover patterns.
- Change the priority of changeover patterns.
- Publish changeover patterns and groups to the Base model.

Understanding Changeovers and Groups

Supply Chain Business Modeler includes the changeover editor, which you use to create changeover patterns. Changeover patterns specify the amount of time and money that is required to change a resource from performing one operation to performing another operation. For example, a changeover pattern can specify the time and cost involved in changing a machine die to a larger bore or changing a printing press to a different color of ink.

Changeover patterns can apply to individual resources and operations or to groups of resources and operations. For example, instead of specifying the time and cost involved in changing a particular paint sprayer from spraying red paint to spraying green paint, a changeover pattern might specify the time and cost involved in changing any paint sprayer at a branch from spraying dark shades of paint to spraying lighter shades of paint.

Supply Chain Business Modeler also includes the group editor, which enables you to create groups of resources or operations to use in changeover patterns. You can create resource and operation groups in the group editor, and then create changeover patterns that apply to those groups of operations or resources. You can also create changeover patterns that apply to all resources and operations.

To model the time and cost associated with changeovers, you can:

- Populate the changeover editor and group editor with branch, resource, operation, and calendar data from the Base model.

  If you have not imported or cleared data in the Base model since the last time you refreshed the changeover editor and group editor, the Refresh option is disabled in the Changeover and Group menus. You must import data or clear in the Base model before you can refresh data in the changeover editor and the group editor.
- Create resource and operation groups for use in the changeover patterns.
- Enter changeover patterns that represent the time and cost required for changing a resource from one operation to another.
- Publish changeover patterns from the changeover editor to the Base model.

Understanding Populating the Changeover and Group Editors with Data

Before you define changeover patterns in the changeover editor, you must populate the editor with branch, resource, operation, and calendar data from the Base model. Similarly, before you can create groups of resources and operations in the group editor, you must populate the group editor with resource and operation codes from the Base model.

To populate the changeover and group editors with data from the Base model, you must refresh the editors. You cannot refresh data separately in the changeover and group editors. If you refresh data in the changeover editor, Supply Chain Business Modeler also refreshes data in the group editor. Similarly, if you refresh data in the group editor, Supply Chain Business Modeler also refreshes data in the changeover editor.

When you refresh data in the editors, the system populates the changeover editor with branch codes that represent plants or plant/distribution centers in the supply chain. The system does not populate the changeover editor with branch codes that represent distribution centers because changeovers cannot occur at branches where no manufacturing occurs.

Understanding Creating and Managing Resource and Operation Groups

Changeover patterns can apply to groups of resources and operations. Supply Chain Business Modeler includes the group editor, which enables you to create groups of resources or operations to use in changeover patterns. You can create resource and operation groups in the group editor, and then create changeover patterns that apply to the groups rather than to individual resources or operations.

After you create a resource or operation group in the group editor and add a new changeover pattern in the changeover editor, you can select the new resource and operation groups from drop-down menus in changeover editor fields.

In the Supply Chain Business Modeler graphical user interface, you can add resources and operations to groups only if the resources and operations are populated in the group editor. Using a Business Modeler Shell command, you can specify resource and operation group members that are not populated in the group editor. These invalid group members appear with a red circle and X when you view the group in the group editor. Groups with invalid members cannot be specified in changeover patterns in the changeover editor.

If unneeded resource and operation groups exist, you can delete or clear the groups from the group editor. When you clear groups from the group editor, all groups that you have defined are deleted, but resource and operation data is not cleared. If you have published the groups to the Base model, the groups are still populated in the Resource Group and Operation Group objects until you clear the Base model or publish new groups to the Base model.

After you create a resource or operation group, you can add members to or remove members from the group.

Understanding Creating and Managing Changeover Patterns

In the changeover editor, you can create changeover patterns that represent the time and cost required for changing a resource from one operation to another.

You can create changeover patterns that apply to individual resources and operations and to groups of resources and operations. You can also create changeover patterns that apply to all resources and operations at a branch by entering wildcard characters in the changeover patterns.

The changeover editor does not allow you to create changeover patterns with branch, resource, and operation value combinations that are not populated in the changeover editor. After you specify a branch for a changeover pattern, you can select only resources or resource groups that are available at that branch. For example, in a changeover pattern for the Cranston branch, the changeover editor only allows you to select a resource group that includes at least one Cranston resource. Similarly, you can select an operation for a changeover pattern only if it is available for the specified branch and resource.

Changeover patterns with invalid branch, resource, and operation values can only exist in the changeover editor if you:

- Refresh the changeover and group editors with Base model data that no longer includes branch, resource, and operation values that were previously populated in the changeover editor.
- Use the bm::changeover addPattern command to create changeover patterns with branches, resources, or operations that are not populated in the changeover editor.

You cannot create two changeover patterns that have the same values in the Branch, Resource, From Operation, and To Operation fields.

If unneeded changeover patterns exist, you can delete or clear the patterns from the changeover editor. When you clear changeover patterns from the changeover editor, all patterns that you have defined are deleted, but branch, resource, operation, and calendar data is not cleared from the changeover editor. If you have published the changeover patterns to the Base model, these changeover patterns are still populated in the Changeover Pattern object until you clear the Base model or publish new changeover patterns to the Base model.

The order of changeover patterns in the changeover editor determines which information is applied to a particular changeover. You must ensure that the information that you want to apply to a changeover is specified in the first pattern in the changeover editor that applies to the changeover.

Before you can create changeover patterns, you must import data from a supply chain management system into the Base model and populate the changeover editor with data by refreshing it with Base model data.

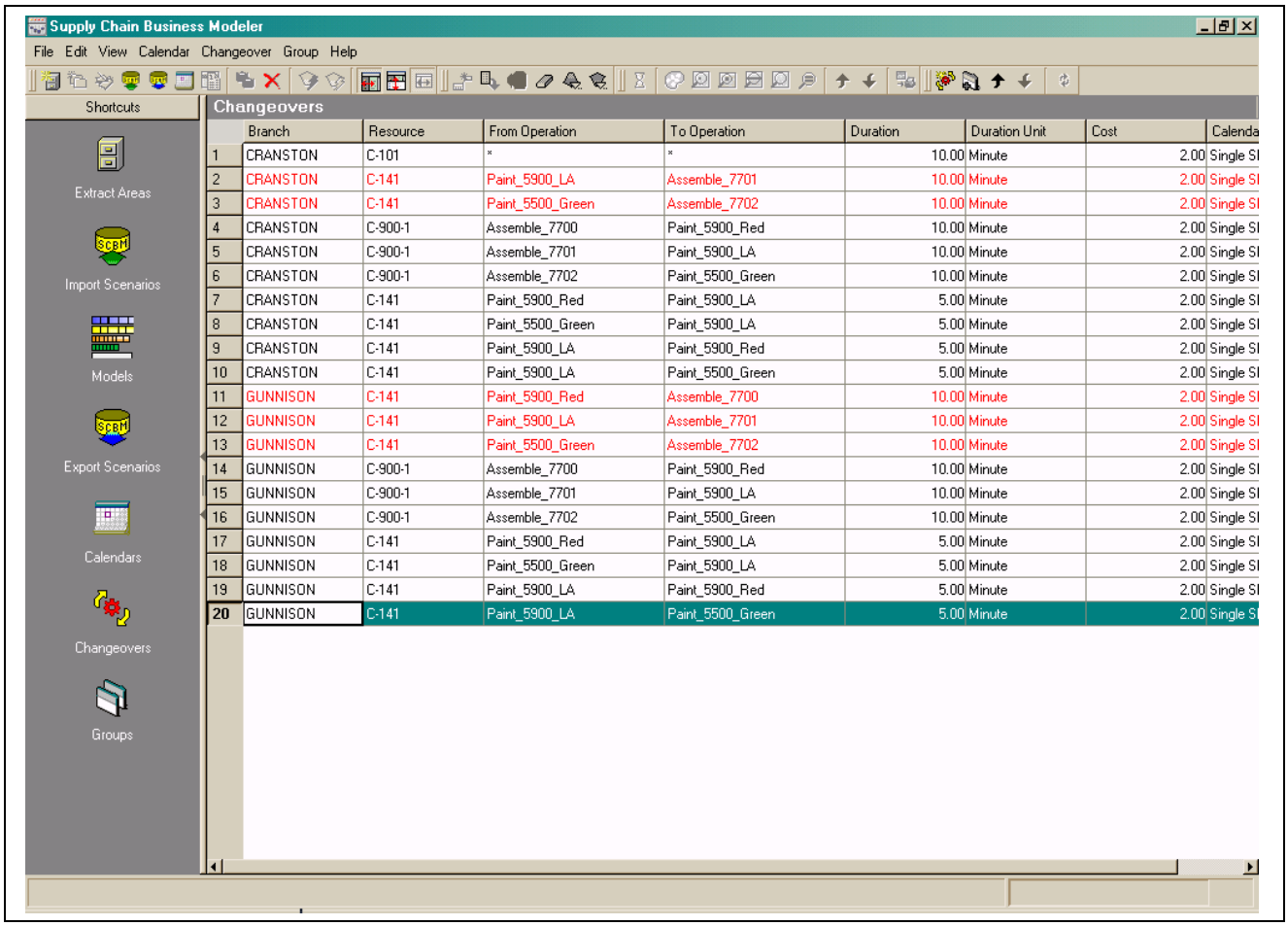
See Also

Changing the Priority of Changeover Patterns

Understanding Changeover Pattern Priorities

The order of patterns in the changeover editor determines the time and cost information that is applied to each changeover. If two patterns apply to the same changeover, information from the pattern that is highest in the changeover editor is applied to the changeover.

The following example shows changeover patterns in the Changeover editor:



Changeover editor

In this example:

- The first changeover pattern indicates that changing the C-101 resource at the Cranston branch between any two operations takes two hours and has a cost of $200.00.
- The second changeover pattern indicates that changing the C-101 resource at the Cranston branch from the Assemble_7700 operation to the Assemble_7701 operation takes three hours and has a cost of $300.00.
- The third changeover pattern indicates that changing any resource at any branch between any two operations takes four hours and has a cost of $100.00.

Although all three patterns provide information that could apply to the changeover of the C-101 machine at the Cranston branch from the Assemble_7700 operation to the Assemble_7701 operation, Supply Chain Business Modeler applies the first changeover pattern in the list to the changeover. Therefore, the duration of this changeover is two hours and the cost is $200.00.

You must be careful to ensure that the information that you want to apply to a changeover is specified in the first pattern in the changeover editor that applies.

Understanding Publishing Changeover Patterns and Groups to the Base Model

After you create changeover patterns in the changeover editor and groups in the group editor, you can publish the patterns and groups to the Base model.

When you publish patterns from the changeover editor, the system clears existing patterns from the Changeover Pattern package in the Base model and populates the package with valid patterns from the changeover editor. If some patterns in the changeover editor are associated with branches, resources, operations, or calendars that do not exist in the Base model, the system highlights these invalid changeover patterns in the changeover editor and does not publish them to the Base model.

When you publish operation and resource groups from the group editor, the system clears existing groups from the Operation Group and Resource Group packages in the Base model and populates these packages with groups from the group editor.

When you publish changeover patterns from the changeover editor to the Base model, these values populate the primary key fields in the Changeover Pattern object.

Invalid patterns appear in red in the changeover editor and are not published to the Base model when you publish changeover editor data. In addition, any new pattern that still appears in red in the changeover editor is not published to the Base model when you publish changeover editor data.

When you publish changeover patterns, the position of each changeover pattern in the changeover editor is populated in the Base model as the pattern's sequence or priority.

Populating the Changeover and Group Editors with Data

This section lists the workspaces used to populate the Changeover and Group editors with supply chain data.

Workspaces Used to Populate the Changeover and Group Editors with Data

Window Name	Navigation	Usage
Changeovers	Click the Changeovers button in the Shortcuts bar. Select Changeover, Refresh.	Populate the changeover editor with data.
Groups	Click the Groups button in the Shortcuts bar. Select Group, Refresh.	Populate the group editor with data.

Creating and Managing Groups and Changeover Patterns

This section lists the workspaces used to create and manage groups and changeover patterns, and discusses how to:

- Create changeover patterns.
- Modify changeover patterns.

Common Elements Used in this Section

Group name	Enter a name for the group.
Description	(Optional) Enter a description of the group.
Assigned resources or operations	Lists resources or operations that are not assigned to the group. To assign a resource or operation to the planning model, click a branch and then click the right-pointing arrow button.
Unassigned resources or operations	Lists resources or operations that are not assigned to the group. To remove a resource or operation from the planning model, click a branch and then click the left-pointing arrow button.
Branch	Select the branch where the changeover occurs. The branch that you select in a row determines the values that appear in the Resource, From Operation , and To Operation fields. You must select a branch in a new changeover row before you can resize columns or select a row in the changeover editor.
Resource	Select the resource. The Resource drop-down list includes only resources and resource groups that are available at the selected branch in the changeover pattern. The branch and resource that you select in a row determine the values in the From Operation and To Operation fields. To create a changeover pattern that applies to all resources at the specified branch, select the asterisk from the drop-down list box.
From Operation	Click the From Operation field in the changeover row and, from the drop-down list box, select the first operation or operation group in the changeover pattern. The From Operation drop-down list includes only operations and operation groups that are associated with the branch and resource that are selected in the changeover pattern. To create a changeover pattern that applies to all operations for the specified resource or resource group at the specified branch, select the asterisk from the drop-down list box.
To Operation	Click the To Operation field in the changeover row and, from the drop-down list box, select the second operation or operation group in the changeover pattern. The To Operation drop-down list includes only operations and operation groups that are associated with the branch and resource that are selected in the changeover pattern. To create a changeover pattern that applies to all operations for the specified resource or resource group at the specified branch, select the asterisk from the drop-down list box.
Duration	Specify the amount of time required for changing the resource from the From Operation to the To Operation.

Cost	Specify the cost associated with changing the resource from the From Operation value to the To Operation value.
Calendar	Displays the calendar in Supply Chain Business Modeler that indicates when the changeover can occur.

Workspaces Used to Create and Manage Groups and Changeover Patterns

Window Name	Navigation	Usage
Groups Properties - Add resource group	Click the Groups button in the Shortcuts bar. Select Resource Groups. Select Group, Add Group.	Create resource groups.
Groups Properties - Add operation group	Click the Groups button in the Shortcuts bar. Select Operation Groups. Select Group, Add Group	Create operation groups.
Group Properties - Modify	Click the Groups button in the Shortcuts bar. Right-click the group that you want to change and select Properties.	Modify groups.
Group Properties - Delete group	Click the Groups button in the Shortcuts bar. Select the group that you want to delete. Select Edit, Delete.	Delete groups.
Group Properties - Clear groups	Click the Groups button in the Shortcuts bar. Select Group, Clear Groups.	Delete all groups.
Changeovers	Click the Changeovers button in the Shortcuts bar.	Add changeover patterns.
Changeovers	Click the Changeovers button in the Shortcuts bar.	Modify changeover patterns.
Changeovers	Click the Changeovers button in the Shortcuts bar. Click the number to the left of the changeover pattern that you want to delete. Select Edit, Delete.	Delete changeover patterns.
Changeovers	Click the Changeovers button in the Shortcuts bar. Select Changeover, Clear Changeovers.	Clear changeover patterns.

Creating Changeover Patterns

Access the Changeovers workspace.

To modify a changeover pattern:

1. From the Changeover menu, select Add Changeover.
2. In the new changeover row, complete the Branch, Resource, From Operation, and To Operation fields.

   If you press the Esc key before you click outside the Branch, Resource, From Operation, and To Operation fields in the new changeover row, the system deletes the new changeover row.
3. In the new changeover row, complete the Cost and Calendar fields.

Modifying Changeover Patterns

Access the Changeovers workspace.

To modify a changeover pattern:

1. In the changeover row, complete theBranch, Resource, From Operation, and To Operation fields.

   If you press the Esc key before you click outside the Branch, Resource, From Operation, and To Operation fields in the new changeover row, the system deletes the new changeover row.
2. In the changeover row, complete the Cost and Calendar fields.

Changing the Priority of Changeover Patterns

This section lists the workspaces used to change the priority of changeover patterns.

Workspaces Used to Change the Priority of Changeover Patterns

Window Name	Navigation	Usage
Changeovers	Click the Changeovers button in the Shortcuts bar. Click the number to the left of the changeover pattern for which you want to establish a new priority. Select Changeover, Move Changeover Up.	Assign a higher priority to a changeover pattern.
Changeovers	Click the Changeovers button in the Shortcuts bar. Click the number to the left of the changeover pattern for which you want to establish a new priority. Select Changeover, Move Changeover Down.	Assign a lower priority to a changeover pattern.

Publishing Changeover Patterns and Groups to the Base Model

This section lists the workspaces used to publish changeover patterns and groups to the Base model.

Workspaces Used to Publish Changeover Patterns and Groups to the Base Model

Window Name	Navigation	Usage
Changeovers	Click the Changeovers button in the Shortcuts bar. Select Changeover, Publish Changeovers.	Publish changeovers.
Groups	Click the Groups button in the Shortcuts bar. Select Group, Publish Groups.	Publish groups.

CHAPTER 10

Refreshing Data in Models

This chapter provides overviews of refreshing data in models, refreshing data in the Tactical, Strategic, and Demand models, data values in time periods, data values in Tactical model time periods, data values in Strategic model time periods, data values in the Demand model, time-related values in time periods, and discusses how to:

- Refresh data in models.
- Clear data from models.

Understanding Refreshing Data in Models

After you import data into Supply Chain Business Modeler and publish data from the calendar, changeover, and group editors, you can populate the appropriate models with the data.

To populate a model other than the Base model with data, you can refresh data in the model. When you refresh data in a model, Supply Chain Business Modeler populates the model with enterprise, transactional, and optimized plan data from a connected model. The data includes the snapshot date from the source model.

When you refresh data in a *reference* model, the system populates the model with enterprise, transactional, and optimized plan data from the connected reference model that is lower in the model hierarchy. The system aggregates the data to the level of granularity required by the destination model.

When you refresh data in a *planning* model, the system populates the model with data from the reference model at the same level in the model hierarchy. Only data that is relevant to the branches assigned to the planning model is populated in the planning model. For example, if a planning model at the Tactical level includes data for Branch A, the planning model is populated with Branch A data from the Tactical model. Only data that is within the planning model horizon is populated in the planning model.

The following table lists the data sources that the system uses when you refresh data in a model:

Model in Which Data Is Refreshed	Data Source
Demand model	Base model
Strategic model	Tactical model
Tactical model	Operation model
Operation model	Base model
Planning model at the Tactical level	Tactical model
Planning model at the Operation level	Operation model

Note. You cannot refresh data in the Base model. To populate the Base model with enterprise, transactional, and optimized plan data, you must import data in Supply Chain Planning XML format. You can also publish optimized plans from a connected reference model.

When you refresh data in a planning or reference model at the Operation or Tactical level, Supply Chain Business Modeler determines whether data was imported into the source or destination model since data was last refreshed in the destination model. If data has been imported, Supply Chain Business Modeler determines what enterprise and transactional data is different in the source model than in the destination model, populates the destination model with any new or changed enterprise and transactional data from the source model, and deletes any data that is not in the source model from the destination model.

If you import data directly into an Operation or Tactical model and then refresh data in the model, Supply Chain Business Modeler does not preserve the imported data. Instead, Supply Chain Business Modeler determines what enterprise and transactional data is different in the source model than in the destination model, populates the destination model with any new or changed enterprise and transactional data from the source model, and deletes any data that is not in the source model from the destination model.

Supply Chain Business Modeler clears enterprise and transactional data from destination model objects before refreshing data in an Operation or Tactical model when:

- A large number of data changes have occurred since data was last cleared in the destination model. If there have been many data changes since data was last cleared in the destination model, it can be quicker and use less memory for Supply Chain Business Modeler to clear data in the destination model before refreshing the data than to populate the model with data changes only.

  Note. If you change the period pattern in the Tactical reference model and then refresh data in the model, Supply Chain Business Modeler must refresh all enterprise and transactional data to aggregate it in the new time periods rather than refreshing only data that has actually changed. To ensure the best possible performance, do not change the period pattern in a model frequently.

- Refreshing optimized plan data. Supply Chain Business Modeler does not detect changes in optimized plan objects.

When you refresh data in the Strategic or Demand model, the system clears existing data from the model before populating the model with data.

When Supply Chain Business Modeler clears data before importing data in the model, it records this action in its progress log.

When you refresh data in a model, the system populates only the destination model with data; the data does not move automatically to any other models. For example, if you refresh data in the Tactical model, only the Tactical model is populated with data from the Operation model. To transfer data between models that are not directly connected, you must populate with the data each connected model between the two models. For example, to populate a planning model at the Tactical level with data from the Base model, you must do the following:

- Refresh data in the Operation model.

  The system populates the Operation model with enterprise, transactional, and optimized plan data from the Base model.

- Refresh data in the Tactical model.

  The system populates the Tactical model with enterprise, transactional, and optimized plan data from the Operation model.

- Refresh data in the planning model at the Tactical level.

The system populates the planning model with data from the Tactical model.

Understanding Refreshing Data in the Tactical, Strategic, and Demand Models

Based on the period type and alignment date in the Tactical, Strategic, and Demand models, Supply Chain Business Modeler determines the pattern of time periods in these models. For example, if the period type in a model is Month and the alignment date is 2005–01–01T00:00:00, Supply Chain Business Modeler creates month time periods that start on the first of each month at 00:00:00, such as 2005–02–01T00:00:00, 2005–03–01T00:00:00, and 2005–04–01T00:00:00.

When you refresh data in the Tactical, Strategic, or Demand model, data is aggregated into the time periods set in the model. If the snapshot date of the data in the source model lies within a time period, Supply Chain Business Modeler divides that time period into two periods: one starting at that period's start and ending at the snapshot date, and the other starting at the snapshot date and ending at that time period's end. For example, if day-long time periods in a model begin at midnight each day, and the model is refreshed with data that has a snapshot date of 2005–06–01T18:00:00, Supply Chain Business Modeler divides that day-long time period into time periods: one beginning at 2005–06–01T00:00:00 and ending at 2005–06–01T18:00:00, and the other beginning at 2005–06–01T18:00:00 and ending at 2005–06–02T00:00:00. The remaining time periods in the model begin at midnight each day.

If you change the period pattern in a Tactical, Strategic, or Demand model after populating the model with data, data values are not automatically aggregated in the new period pattern. You must refresh data in a model to aggregate data in the model using a new time period pattern.

Tactical planning models use the same time period pattern that is set in the Tactical reference model. If you change the Tactical reference model period pattern and refresh data in the Tactical reference model, data in Tactical planning models remains aggregated in the original time period pattern until you refresh the planning model data.

Note. If you change the time in the alignment date in the Tactical reference model and then refresh data in the model, Supply Chain Business Modeler must refresh all enterprise and transactional data to aggregate it in the new time periods rather than refreshing only data that has actually changed. To ensure the best possible performance, do not change the alignment date time in the Tactical model frequently.

See Also

"Setting up Models," Time Period Patterns in the Tactical, Strategic, and Demand Models

Understanding Data Values in Time Periods

This section provides overviews of data values in time periods, rules for determining data values in time periods, rules for determining data values in Tactical model time periods, and rules for determining data values in Strategic model time periods.

Data Values in Time Periods

When you refresh data in the Tactical, Strategic, or Demand model, Supply Chain Business Modeler must determine the appropriate value for each data attribute in each time period. This table lists the types of data values that change when you when you refresh data in a model where data is summarized in time periods:

- Event dates and times.

  When you refresh data in a model where data is summarized in time periods, the system changes some event dates and times to the start date and time or end date and time of the periods that they are in. For example, if a product request date on a sales order is 2005-01-01 15:30:00, and a day-long time period begins at 2005-01-01 00:00:00, the value of the product request date in the Tactical model is 2005-01-01 00:00:00. The start date and time of a period are also used as the period name or ID.

  Some event dates and times that are not used in high level planning— such as sales order creation dates— are not changed when they are refreshed in a model where data is summarized in time periods.

- Calendar objects.

  Calendar objects in Tactical, Strategic, and Demand models include Uptime, Down, Delay, and Premium attribute values for each time period. When you refresh data in a model where data is summarized in time periods, Supply Chain Business Modeler calculates the total amount of each time type in each month period. For example, if there are five daily time periods in a week in the Tactical model that have three hours of premium time, the premium time attribute value for the week time period in the Strategic model is 15.

- Time-varying attributes.

  Supply Chain Business Modeler must determine a value for each time-varying attribute in a time period.

- Time values and units of measure in the Strategic and Demand models.

  When you refresh data in the Strategic or Demand model, Supply Chain Business Modeler converts time-related values to be measured in the time period type for the model. For example, in the Tactical model, the shelf life for an item at a branch could be 14 and the shelf life unit of measure could be days. If the period unit in the Strategic model is weeks, when you refresh data in the Strategic model, the shelf life value in the Strategic model is converted to two and the shelf life unit of measure is weeks.

Note. When you refresh data in the Demand model, Supply Chain Business Modeler does not convert time series data to the time period type for the model. Instead, Supply Chain Business Modeler simply copies time series data from the Base model without aggregating it into the Demand model time periods. In addition, Supply Chain Business Modeler does not aggregate customer data into the channel field in the Time Series Detail object.

Rules For Determining Data Values in Time Periods

When you refresh data in the Strategic, Tactical, and Demand models, Supply Chain Business Modeler uses rules to determine time-varying data values.

The following table describes the rules that are used to determine values for time-varying data when you:

- Refresh data in the Tactical model with data from the Operation model.
- Refresh data in the Strategic model with data from the Tactical model.
- Refresh data in the Demand model with data from the Base model.

Rule	Description	Example
alignDate	The system changes the date and time from the source model to the start date and time of the time period where it falls in the destination model.	If a product request date on a sales order in the Operation model is 2005-01-15T12:00:00 and a day-long time period in the Tactical model begins at 2005-01-15T00:00:00, the product request date in the Tactical model is 2005-01-15T00:00:00.
average	The value is the average value of the attribute values in a time period in a group.	If attribute values are 20, 30, and 10 in a time period, the value is: (20+30+10)/3 = 20
calendarWeightedAverage	The value is the weighted average value of the attribute during the relevant time type in the time period: uptime or premium time. The average is weighted by the amount of time in the time period for which the attribute value is effective.	If an attribute value is 8 for 10 hours of uptime and 4 for 6 hours of uptime in a day-long time period, the value is: ((8*10)+(4*6))/16=6.5
endAlignDate	The system changes the date and time from the source model to the end date and time of the time period where it falls in the destination model.	If an expiry date for some beginning inventory in the Tactical model is 2005-01-15T00:00:00 and a month time period in the Strategic model ends at 2005-02-01T00:00:00, the expiry date in the Strategic model is 2005-02-01T00:00:00.
lastActive	The value is the last value of the attribute during the time period. If no value is specified for the attribute at the end of the time period, the last value of the attribute during the time period is used as the value. If no value is specified for the attribute during the time period, the default attribute value is used as the value.	If an attribute value is 150 at the end of the time period, the value is 150 regardless of whether the attribute had another value during the time period.
longestActive	The value is the value that is used for the longest period of time during a time period.	If an attribute value is 50 for 10 hours of a time period and 100 for 14 hours, the value is 100.
maximum	The value is the maximum value of the attribute during the time period.	If an attribute value is 50, 100, and 150 at different times during the time period, the value is 150.
minimum	The value is the minimum value of the attribute during the time period.	If an attribute value is 50, 100, and 150 at different times during the time period, the value is 50.

Rule	Description	Example
Or	This algorithm is used for boolean attributes that indicate when a resource or operation is available. If the value of this attribute is one at any time during the time period, the value of the attribute is one. If the value of this attribute is zero throughout the time period, the value of the attribute is zero.	**Note.** This algorithm is used with the rangeWeightedAverage and rangeSum algorithms.
rangeSum	The value is the total of the attribute values in the period. If the start or end of a time period in the source model is not aligned with the start or end of a time period in the destination model, the system calculates the proportion of the value that is in the period, and adds that to the total.	If an enterprise forecast for an item at a branch is 100, 200, and 300 for three calendar months, the item forecast in a calendar quarter in the Strategic model is: 100+200+300=600 If the enterprise forecast in the Tactical model is 25, 25, 25, 50, and 49 in weekly periods, and the Strategic model is summarized in calendar month time periods, the end of the calendar month is not usually aligned with the end of a week. The system then calculates the proportion of the forecast amount in the time period. If 2 of the 7 days of the week time period are contained in the month time period in the Strategic model, the item forecast for the month in the Strategic model is: 25+25+25+50+(49*(2/7))=139

Rule	Description	Example
rangeWeightedAverage	The value is the average value of the attribute during the time period, weighted by the amount of time in the time period for which each value is effective. If no attribute value is specified for part of a time period, the time is still included in the calculation of the average. If an object includes a Boolean availability attribute, only the time when the object is available is included in the weightedAverage calculation. If the availability attribute indicates that the attribute is unavailable for the entire time period, the default attribute value is used as the value.	If an attribute value is 100 for 16 hours and 200 for the remaining 8 hours in a day-long time period, the value is: ((16*100)+(8*200))/24=133.33 If an attribute value is 100 for 16 hours and no value is specified for the remaining 8 hours in a day-long time period, the value of the attribute is: 16*100/24=66.67 In a day-long time period, if an object is: • Available for 16 hours with an attribute value of 100 • Available for 3 hours with an attribute value of 200 • Unavailable for 5 hours, and includes a Boolean availability attribute the value of the attribute is: ((16*100)+(3*200))/(24-5)=115.79
rateCumulation	The value is the resource capacity during uptime or premium time in a time period multiplied by the amount of relevant time in the associated calendar.	If a resource's capacity is 10 during premium time in a time period, and there are four hours of premium time in the time period in the associated calendar, the value of the attribute is: 10*4=40
sum	The value is the sum of the attribute values during the time period.	If attribute values are 20, 30, and 10 in a time period, the value is: 20+30+10=60
weightedAverage	The value is the weighted average of the attribute based on quantities in a time period.	If an attribute value is 20 for 100 items and 15 for 1000 items in a time period, the value is: ((20*100)+(15*1000))/(1100)=15.45

Note. Optimized plans that you publish data from the Tactical model to the Operation model are created for specific dates and times. These dates and times do not change when they are published from a model.

See Also

"Setting Up Models"

Understanding Data Values in Tactical Model Time Periods

When you refresh data in the Tactical model, Supply Chain Business Modeler copies some values from the Operation model without changing the values and changes the values of some dates and time-varying data.

The following table lists the objects and attribute values that are changed when you refresh data in the Tactical model, and lists the rules that are used to calculate the value in each time period:

Tactical Model Object	Tactical Model Attribute	Rule Used to Determine Value in Each Time Period
ItemStorage	storageCost	RangeWeightedAverage
	preference	LastActive
Storage	storageWeightCapacity	LastActive
	storageVolumeCapacity	LastActive
ItemBranch	itemCost	RangeWeightAverage
InventoryPolicy	maximumLevel	LastActive
	minimumLevel	LastActive
	stockoutCost	RangeWeightedAverage
	maximumCoverage	LastActive
	useCoverForSafety	LastActive
	useCoverForMaximum	LastActive
BranchProductSubstitution	cost	RangeWeightedAverage
	preference	LastActive
TransportMode	numberofVehicles	RangeWeightedAverage
	flatRatePerTrip	RangeWeightedAverage
	costPerWeightUnit	RangeWeightedAverage
	costPerVolumeUnit	RangeWeightedAverage
SupplierItem	standardCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	minimumPeriodOrder	minimum
	maximumPeriodOrder	maximum
	fixedOrderQuantity	maximum

Tactical Model Object	Tactical Model Attribute	Rule Used to Determine Value in Each Time Period
	demandCoverageOrder	maximum
CustomerSourcing	preference	lastActive
ItemPrice	price	RangeWeightedAverage
CustomerProductSubstitution	cost	RangeWeightedAverage
	preference	lastActive
ProducedItemBOM	productionOverheadCost	RangeWeightedAverage
ConsumedItemBOM	available	Or
	quantity	RangeWeightedAverage
	scrap	RangeWeightedAverage
	consumptionOverheadCost	RangeWeightedAverage
	preference	LastActive
OperationBOR	available	Or
	duration	RangeWeightedAverage
	capacityRequired	RangeWeightedAverage
	lotMultiple	RangeWeightedAverage
	overheadCost	RangeWeightedAverage
	preference	LastActive
BranchOperation	available	Or
	overheadCost	RangeWeightedAverage
BranchRouting	available	Or
	overheadCost	RangeWeightedAverage
Crew	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	CalendarWeightedAverage

Tactical Model Object	Tactical Model Attribute	Rule Used to Determine Value in Each Time Period
	premiumCost	CalendarWeightedAverage
	fixedCost	RangeWeightedAverage
Machine	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	fixedCost	RangeWeightedAverage
Tool	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	fixedCost	RangeWeightedAverage
BeginningInventory	statusDate	AlignDate
	manufacturingDate	AlignDate
	expiryDate	EndAlignDate
SalesOrderDetail	requestedDate	AlignDate
	shipDate	AlignDate
	arriveDate	AlignDate
	pickDate	AlignDate
PurchaseOrderDetail	shipDate	AlignDate
	arriveDate	AlignDate
TransferOrderDetail	shipDate	AlignDate

Tactical Model Object	Tactical Model Attribute	Rule Used to Determine Value in Each Time Period
	arriveDate	AlignDate
WorkOrder	requestedDate	AlignDate
	startDate	AlignDate
	completionDate	AlignDate
WorkOrderOperation	requestedDate	AlignDate
	startDate	AlignDate
	completionDate	AlignDate
	earliestStart	AlignDate
	latestEnd	AlignDate
NetDeploymentRequirementDetail	shipDate	AlignDate
	earliestShipDate	AlignDate
	latestShipDate	AlignDate
	arriveDate	AlignDate
	earliestArriveDate	AlignDate
	latestArriveDate	AlignDate
NetProductionRequirement	createdDate	AlignDate
NetProductionRequirementDetail	earliestProductionDate	AlignDate
	latestProductionDate	AlignDate
InventorySafetyTargetDetail	safetyLevel	LastActive
PurchaseOrderRecommendationDetail	arriveDate	AlignDate
	shipDate	AlignDate
SourcingRecommendationDetail	available	Or

Understanding Data Values in Strategic Model Time Periods

When you refresh data in the Strategic model, Supply Chain Business Modeler aggregates data in the following objects.

- Sales object.

  When you refresh data in the Strategic model, Supply Chain Business Modeler populates the Sales object in the Strategic model with data from the SalesOrder object in the Tactical model. For each item and shipping branch combination in each time period, Supply Chain Business Modeler calculates the sum of the quantity and the weightedAverage of the unitPrice.

- PurchaseOrder object.

  When you refresh data in the Strategic model, Supply Chain Business Modeler populates the PurchaseOrder object with data from the PurchaseOrder object in the Tactical model. For each item, destination branch, supplier, transport mode, and carrier combination in each time period, Supply Chain Business Modeler calculates the sum of the quantity and the weightedAverage of the unitCost.

- TransferOrder object.

  When you refresh data in the Strategic model, Supply Chain Business Modeler populates the TransferOrder object with data from the TransferOrder object in the Tactical model. For each item, origin branch, destination branch, transport mode, and carrier name combination in each time period, Supply Chain Business Modeler calculates the sum of the quantity.

- Forecast object.

  When you refresh data in the Strategic model, Supply Chain Business Modeler populates the Forecast object with data from the EnterpriseForecastDetail object in the Tactical model. For each item and branch combination in each time period, Supply Chain Business Modeler calculates the rangeSum of the quantity.

The following table lists the rules that Supply Chain Business Modeler uses to populate other time-varying attributes in the Strategic model with data from the Tactical model when you refresh data in the Strategic model:

Note. Some attributes are also converted to different time units when they are refreshed in the Strategic model. Time units are converted to the period duration units.

See Time Values and Units of Measure in the Strategic and Demand Models.

Strategic Model Object	Strategic Model Attribute	Rule Used to Determine Value in Each Time Period
ItemStorage	storageCost	RangeWeightedAverage
	preference	LastActive
Storage	storageWeightCapacity	LastActive
	storageVolumeCapacity	LastActive
ItemBranch	itemCost	RangeWeightedAverage
InventoryPolicy	maximumLevel	LastActive
	minimumLevel	LastActive
	stockoutCost	RangeWeightedAverage
	maximumCoverage	LastActive
	useCoverForSafety	LastActive

Strategic Model Object	Strategic Model Attribute	Rule Used to Determine Value in Each Time Period
	useCoverForMaximum	LastActive
TransportMode	numberOfVehicles	RangeWeightedAverage
	flatRatePerTrip	RangeWeightedAverage
	costPerWeightUnit	RangeWeightedAverage
	costPerVolumeUnit	RangeWeightedAverage
SupplierItem	standardCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	minimumPeriodOrder	Minimum
	maximumPeriodOrder	Maximum
	fixedOrderQuantity	Maximum
	demandCoverageOrder	Maximum
CustomerSourcing	preference	LastActive
ItemPrice	price	RangeWeightedAverage
ProducedItemBOM	productionOverheadCost	RangeWeightedAverage
ConsumedItemBOM	available	Or
	quantity	RangeWeightedAverage
	scrap	RangeWeightedAverage
	consumptionOverheadCost	RangeWeightedAverage
	preference	LastActive
OperationBOR	available	Or
	duration	RangeWeightedAverage
	capacityRequired	RangeWeightedAverage
	lotMultiple	RangeWeightedAverage
	overheadCost	RangeWeightedAverage
	preference	LastActive
BranchOperation	available	Or

Strategic Model Object	Strategic Model Attribute	Rule Used to Determine Value in Each Time Period
	overheadCost	RangeWeightedAverage
BranchRouting	available	Or
	overheadCost	RangeWeightedAverage
Crew	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	fixedCost	RangeWeightedAverage
Machine	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	fixedCost	RangeWeightedAverage
Tool	availableCapacity	CalendarWeightedAverage, RateCumulation
	premiumCapacity	CalendarWeightedAverage, RateCumulation
	setupCost	RangeWeightedAverage
	operatingCost	RangeWeightedAverage
	premiumCost	RangeWeightedAverage
	fixedCost	RangeWeightedAverage
BeginningInventory	statusDate	AlignDate

Strategic Model Object	Strategic Model Attribute	Rule Used to Determine Value in Each Time Period
	manufacturingDate	AlignDate
	expiryDate	EndAlignDate

See Also

"Refreshing Data in Models"

Understanding Data Values in the Demand Model

This section provides overviews of customer aggregation in the Demand model and determining data values in the Demand model.

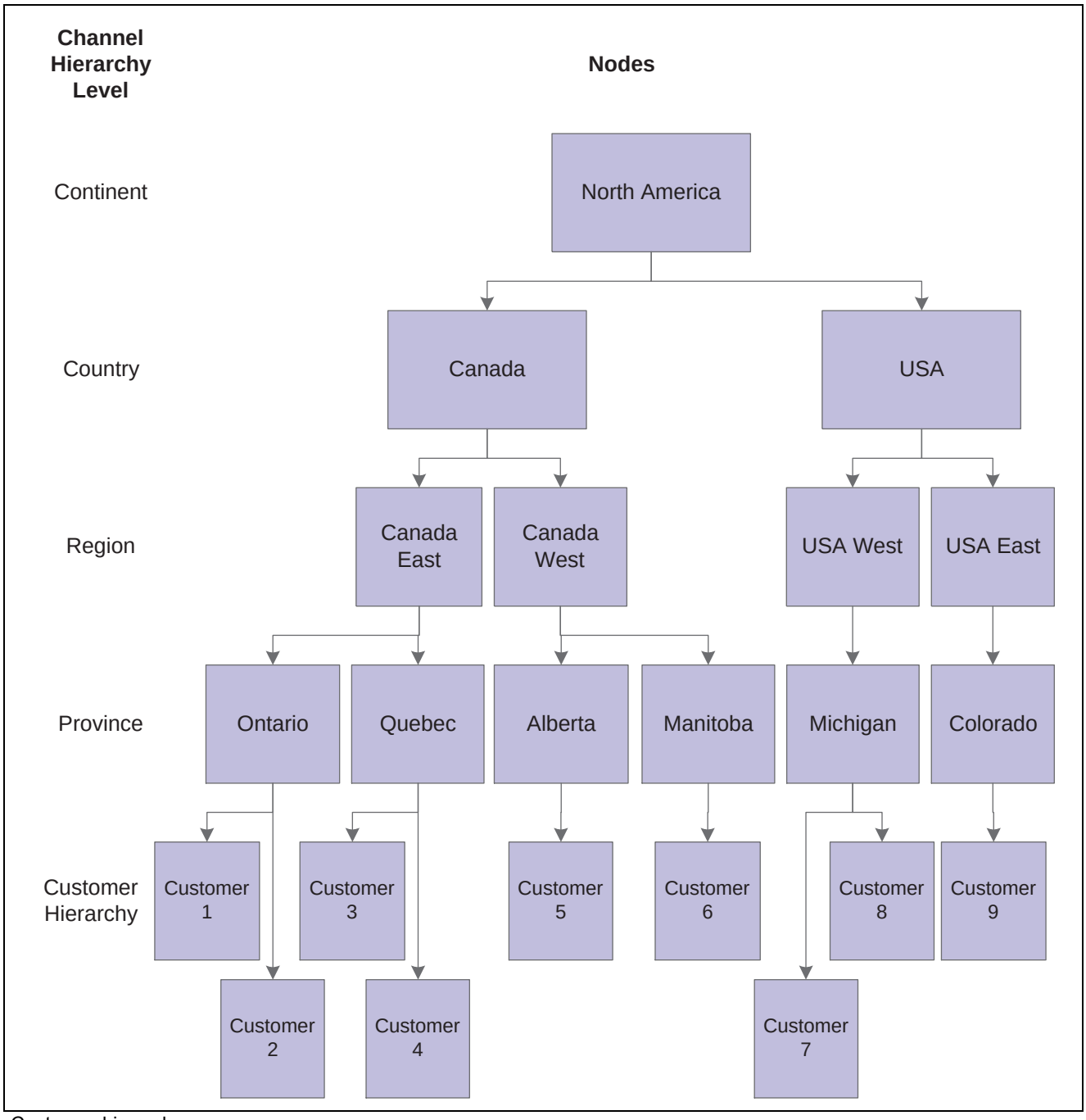
Customer Aggregation in the Demand Model

When you refresh data in the Demand model, customer data from the Base model is populated in the Channel object in the Demand model. The level of customer aggregation that you set in the Demand model determines how data is populated in the Channel object and in related objects in the Demand model.

If you select no customer aggregation in the Demand model, each channel in the Demand model represents an individual customer in the Base model. When you refresh data in the Demand model, Supply Chain Business Modeler directly copies data from the Customer, Customer Group, and Customer Hierarchy Level objects in the Base model to the Channel, Channel Group, and Channel Hierarchy Level objects in the Demand model.

If you select a customer group aggregation level, each channel in the Demand model represents a group of customers from the Base model.

For example, the following diagram shows a customer hierarchy in the Base model:



Customer hierarchy

In this example, if you set the customer aggregation level in the Demand model to Region, Supply Chain Business Modeler aggregates customer data at the Region level, and does not save data for provinces or individual customers in the Demand model channel object. As a result, in the Demand model:

- Canada East, Canada West, USA West, and USA East are channels.
- Continent and Country are channel hierarchy levels.
- Canada and USA are channel groups.
- North America is a channel hierarchy node.

However, if you select No aggregation, each customer in the Base model becomes a channel in the Demand model when you refresh the Demand model data.

The following table shows the channels and channel hierarchy levels that are populated in the Demand model, depending on the customer aggregation level that you specify in the Demand model:

Customer Aggregation Level in Demand Model	Channels	Channel Hierarchy Levels
Continent	North America	none
Country	Canada USA	Continent
Region	Canada East Canada West USA West USA East	Continent Country
No aggregation	Customer 1 Customer 2 Customer 3 Customer 4 Customer 5 Customer 6 Customer 7 Customer 8	Continent Region Country

When you refresh the Demand model, channel values are also used to populate the LeafDemandPoint object in the Demand model. Leaf demand points are demand points at the lowest level in the model and are combinations of items, branches from which the items can be sold, and channels into which the item can be sold. Demand points are points in the business model where potential exists for a product to be sold from a location into a channel. For example, a leaf demand point could represent a particular retailer (Customer 1) that distributes mountain bikes (item) that it received from a particular company location (branch).

Determining Data Values in the Demand Model

When you refresh data in the Demand model, Supply Chain Business Modeler determines data values in the following objects:

- Sales object.

  When you refresh data in the Demand model, Supply Chain Business Modeler populates the Sales object in the Demand model with data from the SalesObject object in the Base model. For each item, branch, and channel combination in each time period, Supply Chain Business Modeler calculates the sum of the quantity attribute and the weightedAverage of the unitPrice attribute.

- SafetyParameters object.

When you refresh data in the Demand model, Supply Chain Business Modeler first uses the lastActive rule to determine the replenishment lead time value for each time period in the Demand model, and converts the resulting value to be measured in the time period unit in the Demand model. Then, for each combination of item, branch, channel, and service level, Supply Chain Business Modeler calculates the rangeWeightedAverage of the replenishment lead time values in each time period.

- Demand Point History Detail object.

  When you refresh data in the Demand model, Supply Chain Business Modeler populates the Demand Point History Detail object with data from the SalesOrderHistory object in the Base model. Demand Point History data is populated for the number of periods prior to the snapshot date specified by the Demand History number of periods. In the Demand model, for each combination of item, branch, channel, and demand point history name, Supply Chain Business Modeler calculates the sum of the quantity and the weightedAverage of the unitPrice in each time period.

  If one or more time periods at the start of the Demand Point History periods have no quantity for an item, branch, channel, and demand point history name combination, Supply Chain Business Modeler does not specify a value for the quantity in these periods. However, if a quantity value in one or more time periods is followed by one or more time periods with no quantity, Supply Chain Business Modeler assigns 0 as the quantity value for the time periods with no quantity. For example, if the quantity values in six periods of demand point history are 0, 0, 10, 10, 0 and 10, Supply Chain Business Modeler does not populate quantity values in the first two periods but assigns values of 10, 10, 0, and 10 for the remaining periods.

  Note. Supply Chain Business Modeler sets the historyName value to SCBM Historical.

- ItemPrice object.

  When you refresh data in the Demand model, Supply Chain Business Modeler populates the ItemPrice object with data from the Customer object in the Base model. First, Supply Chain Business Modeler uses the rangeWeightedAverage rule to calculate a price value for each customer in each time period. Then, for each combination of item, branch, and channel, Supply Chain Business Modeler uses the rangeWeightedAverage rule to determine the price in each time period.

See Also

"Refreshing Data in Models"

Understanding Time-Related Values in Time Periods

This section provides overviews of time values and units of measure in the Strategic and Demand models and costs associated with time periods in the Strategic model.

Time Values and Units of Measure in the Strategic and Demand Models

When you refresh data in the Strategic or Demand model, Supply Chain Business Modeler converts time values such as inventory policies, lead times, and durations to be measured in the time period unit in the destination model.

Time values and units of measures are usually saved in separate object attributes in models. For example, in ItemBranch objects, the shelf life of an item at a branch is saved in a separate attribute from the shelf life unit of measure. In these cases, when you refresh data in the Strategic or Demand model, Supply Chain Business Modeler changes the time unit of measure attribute to the period unit and changes the time value attribute to be measured in the period unit. For example, in the Tactical model, the shelfLife value could be 14 and the shelfLifeUom value could be days. If you refresh the Strategic model with this data when the Strategic model time period value is weeks, the shelfLifeUom value in the Strategic model becomes weeks, and the shelf life value changed to two.

For calculations where time periods in the Strategic or Demand model are calendar months or quarters, Supply Chain Business Modeler considers a month to be 30 days, and a calendar quarter to be 90 days. For example, if the shelf life for an item at a branch is 90 days in the Tactical model, and the period unit is calendar months in the Strategic model, the shelf life in the Strategic model is calculated as 90/30=3 months. If the time period unit in the Strategic model is calendar quarters, the shelf life is calculated as 90/90=1 calendar quarter when you refresh data in the Strategic model.

When you refresh the Demand model with data from the Base model, the only object where a time value and unit of measure is changed is in the SafetyParameters object. In this object, the replenishmentLeadTime value is converted to be measured in the period unit and the timeUnit attribute value is changed to the Demand model period unit.

The following table lists the time-related objects and attributes that are converted when you refresh data in the Strategic model:

Note. Some of the attributes values listed in the following table are also changed using other formulas when you refresh data in the Strategic model.

See “Refreshing Data in Models,” Understanding Determining Data Values in Strategic Model Time Periods.

Costs Associated with Time Periods in the Strategic Model

When you refresh the Strategic model with data, Supply Chain Business Modeler converts some costs that are associated with particular periods of time to represent the costs associated with a single time period in the Strategic model.

For example, in the ItemStorage object in the Tactical model, the storageCost for an item at a branch could be $10, the storageCostTime could be 60, and the storageCostTimeUnit could be days. These values indicate that it costs $10 to store the item for 60 days. When you refresh data in the Strategic model, the cost is changed to reflect the cost of storing the item for one calendar quarter. If the period unit is calendar quarters in the Strategic model, the ItemStorage object in the Strategic model would have a storageCost of $15, a storageCostTime of 1, and a storageCostTimeUnit of calendar quarters.

For calculations where time periods are calendar months or quarters, Supply Chain Business Modeler considers a month to be 30 days, and a calendar quarter to be 90 days.

In the Item Storage object, the storageCost attribute and its related time attributes (storageCost and storageCostTimeUnit) are changed. In the Tool, Machine, and Crew objects, the operatingCost and premiumCost attributes and the related timeUnitForCost attributes are changed.

Some of these attribute values are also changed using other formulas when you refresh data in the Strategic model.

See “Refreshing Data in Models,” Understanding Determining Data Values in Strategic Model Time Periods.

Refreshing Data in Models

This section lists the workspace used to refresh data in models.

Workspace Used to Refresh Data in Models

Window Name	Navigation	Usage
Models	Click the Models button in the Shortcuts bar. Right-click the model that you want to populate with data, and select Refresh Model.	Refresh data in models.

Clearing Data From Models

This section lists the workspace used to clear data from models.

Workspace Used to Clear Data from Models

Window Name	Navigation	Usage
Models	Click the Models button in the Shortcuts bar. Right-click the model that you want to clear, and select Clear Model.	Clear all data from models. **Note.** When you clear data from the Base model, the Default calendar remains.

CHAPTER 11

Publishing Data from Models

This section provides an overview of publishing data from models and lists the workspace used to publish data from models.

Publishing Data from Models

To populate a model with optimized plan data, you can publish optimized plans from a connecting model. You can publish plans from any Supply Chain Business Modeler model except the Base model.

When you publish plans from a reference model, the system populates the connected reference model that is lower in the model hierarchy with the optimized plan data. Any existing optimized plan data is cleared from the destination model.

When you publish plans from a planning model, the system populates the optimized plan data from the model in the reference model at the same model level. The system clears any optimized plan data in the destination reference model for branches that are assigned to the planning model. Data for other branches is not cleared from the reference model.

When you publish optimized plans from a model, Supply Chain Business Modeler checks whether the snapshot date specified for the published plan matches the snapshot date of the enterprise and transactional data that is saved in the destination model. If the snapshot dates are the same, Supply Chain Business Modeler publishes the data to the destination model. If the snapshot dates are not the same, a warning message appears and Supply Chain Business Modeler does not publish the data to the destination model.

The following table lists the data destinations when you publish data from a model:

Model from Which Data Is Published	Data Destination
Demand model	Base model
Strategic model	Tactical model
Tactical model	Operation model
Operation model	Base model
Planning model at the Tactical level	Tactical model
Planning model at the Operation level	Operation model

You cannot publish data from the Base model. To transfer data from the Base model to the Operation model or Demand model, you must refresh data in the destination model.

Workspace Used to Publish Data from Models

Window Name	Navigation	Usage
Models	Click the Models button in the Shortcuts bar. Right-click the model from which you want to publish optimized plans, and select Publish Model.	Publish data from models.

CHAPTER 12

Exporting Data from Supply Chain Business Modeler

This chapter provides overviews of export scenarios and using data exported from Supply Chain Business Modeler and discusses how to:

- Create, modify, and delete export scenarios.
- Run export scenarios.

Understanding Export Scenarios

After importing data into Supply Chain Business Modeler and populating the appropriate model with the data, you can export the data for use in a Supply Chain Management or Supply Chain Planning system. To export data from Supply Chain Business Modeler, you must create and run export scenarios. In an export scenario, you specify:

- The data that you want to export from Supply Chain Business Modeler.
- The version of Supply Chain Planning XML that you want to export. You can export data in Supply Chain Planning XML 3.3, 3.2, Supply Chain Planning XML 3.1, Supply Chain Planning XML 3.0, or Supply Chain Planning XML 2.0 format.
- The model from which you want to export data.
- Tool Command Language (Tcl) scripts that you want to run before or after exporting the data.

  Tcl scripts enable you to link external processes such as planning application solves or data extracts to the export scenario. Pre- and post-export Tcl scripts must be saved with a .tcl extension in the Scripts directory of the extract area specified in the scenario. Pre- and post-scripts cannot include Business Modeler commands.

You can create multiple export scenarios to support different business processes, and run each scenario as often as it is required. For example, you could create the following scenarios:

- An export scenario that exports dynamic data such as sales orders and work orders from a planning model at the Tactical level and invokes a solve in Production and Distribution Planning.
- An export scenario that exports optimized plans from the Base model in Supply Chain Business Modeler, and initiates a data import in a supply chain management system.

After you export data from Supply Chain Business Modeler, you can import the data into a Supply Chain Management or Supply Chain Planning system. You must export data from the correct model level in Supply Chain Business Modeler before you can import it into a supply chain system. For example, data that you import into Production and Distribution Planning must be exported from the Tactical model. Data that you import into a supply chain management system must be exported from the Base model.

See Also

"Exporting Data from Supply Chain Business Modeler"

Using Data Exported from Supply Chain Business Modeler

When you import data from Supply Chain Business Modeler into another supply chain system, you must use the exact name for each data package.

The following table lists the correct names for the data packages:

Data Package Category	Data Package Names
Enterprise data	• Base.xml • Customer.xml • Supplier.xml • Distribution.xml • Manufacturing.xml • Calendar.xml

Data Package Category	Data Package Names
Transactional data	Data packages in this category include: • BeginningInventory.xml • SalesOrders.xml (Sales.xml in the Demand model) • SalesOrderHistory.xml (DemandPointHistory.xml in the Demand model) • WorkOrders.xml • TimeSeries.xml • PurchaseOrders.xml • TransferOrders.xml • SafetyParameters.xml • Configuration.xml
Optimized plans	Data packages in this category include: • EnterpriseForecast.xml • InventoryBuildTargets.xml • InventorySafetyTargets.xml • SourcingRecommendations.xml • NetProductionRequirements.xml (MasterProductionPlan.xml package in the Base model) • NetDeploymentRequirements.xml (DeploymentPlan.xml package in the Base model) • ProductionSchedule.xml (DetailedProductionPlan.xml in the Base model) • PurchaseOrderRecommendations.xml (PurchasePlan.xml in the Base model)

Creating, Modifying, and Deleting Export Scenarios

This section lists prerequisites and common elements, and discusses how to:

- Create export scenarios.
- Modify export scenarios.

Prerequisites

Before you complete the tasks in this section:

- Create an extract area and at least one data folder for saving data files that you export from Supply Chain Business Modeler.

  See “Creating Extract Areas and Data Folders,” Setting Up Extract Areas and Data Folders.

- Save any Tcl scripts that you want to run as pre-export or post-export scripts in the Scripts folder of the extract area.

  See “Scripts Folders,” Setting Up Extract Areas and Data Folders.

Common Elements Used in this Section

Data Folder	Select the name of the data folder for saving the exported data package. Data packages that you want to export in an export scenario must be saved in one or more data folders in a single extract area.
Enterprise Data	Select this option to display Enterprise Data package names in the data selection list. Packages in the Enterprise Data category include static data, such as supplier, customer, and distribution information.
Export compressed	Select this option to export the data in compressed gzip format. Supply Chain Planning XML files in gzip format have the .xml.gz extension.
Extract area	Enter the name of the extract area where exported data packages are saved. You cannot create an export scenario until you create an extract area.
Model	Enter the name of the model from which you want to export data. You can export data from the Base model for use in a Supply Chain Management system and from Operation, Tactical, Strategic, and Demand models for use in EnterpriseOne Supply Chain Planning applications.
Name	Enter a name for the export scenario.
Optimized Plans	Select this option to display Optimized Plans package names in the data selection list. Packages in the Optimized Plans category include supply chain plans and forecasts created by EnterpriseOne supply chain planning applications such as Production Scheduling, Production and Distribution Planning, Strategic Network Optimization, Demand Forecasting, and Demand Consensus.
Package	Select this option to export all data packages in the data selection list in the export scenario.
Post-script	Enter the name of a Tool Command Language(Tcl) script that you want to run after exporting data. Post-import Tcl scripts enable you to call external processes that you would like to link to the data import, such as planning application solves and data extracts. The Tcl script must be saved with the .tcl extension in the Scripts directory of the extract area used in the scenario, and cannot include Business Modeler commands.
Pre-script	Enter the name of a Tcl script that you want to run before exporting data. Pre-import Tcl scripts enable you to call external processes that you want to link to the data export, such as planning application solves and data extracts. The Tcl script must be saved with the .tcl extension in the Scripts directory of the extract area used in the scenario, and cannot include Business Modeler commands.
Set all data folders to this default folder	To export all data packages to the same data folder, select a data folder from this field.
Supply Chain Planning v2.0	Select this option to export data in Supply Chain Planning XML 2.0 format, in which detailed information is separated from its related header data.

Supply Chain Planning v3.0	Select this option to export data in Supply Chain Planning XML 3.0 format, in which detailed information is nested within its related header data.
Supply Chain Planning v3.1	Select this option to export data in Supply Chain Planning XML 3.1 format.
Supply Chain Planning v3.2	Select this option to export data in Supply Chain Planning XML 3.2 format.
Supply Chain Planning v3.3	Select this option to export data in Supply Chain Planning XML 3.3 format.
Transactional Data	Select this option to display Transactional Data package names in the data selection list. Packages in the Transactional Data category include data that changes frequently, such as sales orders, transfer orders, and work orders.

Workspaces Used to Create, Modify, and Delete Export Scenarios

Window Name	Navigation	Usage
Add Export Scenario	• Click the Export Scenarios button in the Shortcuts bar. Select File, Add Export Scenario. • Click the Export Scenarios button in the Shortcuts bar. Right-click the Export Scenario Navigator and select Add Export Scenario. • In the Standard toolbar, click the Add Export Scenario button.	Create export scenarios.
Modify Scenario	Click the Export Scenarios button in the Shortcuts bar. In the Export Scenario Navigator, right-click the export scenario and select Modify Scenario.	Modify export scenarios.
Export Scenarios - Renaming	• Click the Export Scenarios button in the Shortcuts bar. Select the export scenario and select Edit, Rename. • Click the Export Scenarios button in the Shortcuts bar. Right-click the export scenario and select Rename.	Rename export scenarios.
Export Scenarios - Deleting	• Click the Export Scenarios button in the Shortcuts bar. Select the export scenario and select Edit, Delete. • Click the Export Scenarios button in the Shortcuts bar. Right-click the export scenario and select Delete.	Delete export scenarios.

Creating Export Scenarios

Access the Add Export Scenario workspace.

To create an export scenario:

1. In Introduction, click Next.
2. In Scenario Definition, complete the Name, Model, and Extract area fields.
3. In Export Format, select the Supply Chain Planning v3.3, Supply Chain Planning v3.2, Supply Chain Planning v3.1, Supply Chain Planning v3.0, or Supply Chain Planning v2.0 option.

4. (Optional) Select the Export compressed option.
5. Click Next.
6. (Optional) In Pre and Post Scripts, complete the Pre-script and Post-script fields.
7. Click Next.
8. In Data Selection, select one or more of the following options:
   - Enterprise Data
   - Transactional Data
   - Optimized Plans
9. Select either the Package option or the individual data packages that you want to export.
10. Complete the Set all data folders to this default folder field or complete the Data Folder field for each package.
11. Click Next.
12. In Summary, click Finish.

Modifying Export Scenarios

Access the Modify Scenario workspace.

To modify an export scenario:

1. In Scenario Definition, complete the Model and Extract area fields.
2. Click Next.
3. In Export Format, select the Supply Chain Planning v3.3, Supply Chain Planning v3.2, Supply Chain Planning v3.1, Supply Chain Planning v3.0, or Supply Chain Planning v2.0 option.
4. (Optional) Select the Export compressed option.
5. Click Next.
6. (Optional) In Pre and Post Scripts, complete the Pre-script and Post-script fields.
7. Click Next.
8. In Data Selection, select one or more of the following options:
   - Enterprise Data
   - Transactional Data
   - Optimized Plans
9. Select either the Package option or the individual data packages that you want to export.
10. Complete the Set all data folders to this default folder field or complete the Data Folder field for each data package.
11. Click Next.
12. In Summary, click Finish.

Running Export Scenarios

This section provides an overview of export scenarios, lists a prerequisite, and lists workspaces used to run export scenarios.

Understanding Running Export Scenarios

When you run an export scenario, Supply Chain Business Modeler exports the data packages that are specified in the scenario to the data folder or folders that are specified in the scenario. The data is exported in the Supply Chain Planning XML version that is specified in the scenario. Supply Chain Business Modeler also runs any Tcl scripts that are specified as pre-scripts or post-scripts in the export scenario. The model, extract area, data folder, and data files that are specified in the export scenario must all exist before the scenario can run successfully.

When you export data from a model, Supply Chain Business Modeler uses the snapshot date that is saved with the model data and the export horizon duration in the model to calculate the end of the export horizon. For example, if the snapshot date is 2004-01-01T00:00:00 and the export horizon of a model is 31 days, the export horizon extends until 2004-02-01T00:00:00.

The export horizon affects which data Supply Chain Business Modeler exports from a model for each of the following data types:

- Effective-dated data.

  Supply Chain Business Modeler only exports time-varying data values that are in effect at some time during the export horizon. In models where data is aggregated in time periods, Supply Chain Business Modeler only exports bucketed data in time periods that overlap the export horizon. For example, Supply Chain Business Modeler only exports inventory policy information that is in effect at some time during the export horizon.

- Event data.

  Supply Chain Business Modeler exports all fixed data values from before the export horizon end date in a model, including fixed data values from before the snapshot date. For example, Supply Chain Business Modeler exports data for all sales orders before the snapshot date because they can provide backorder information that can be useful in supply chain planning.

After exporting data from Supply Chain Business Modeler, you can view the Supply Chain Planning XML documents in the specified extract area and import the data into supply chain systems.

Note. To determine which Supply Chain Business Modeler data is time-varying, bucketed, and fixed, you can use the schema viewer that is provided with Supply Chain Business Modeler.

See Also

"Setting Up Models," Obtaining Model Information

Prerequisites

Before you complete the tasks in this section:

- Ensure that the model, extract area, data folder, and any Tcl scripts that are specified in the scenario exist.
- If you have renamed an extract area, folder, file, or script, ensure that the new name is specified in the scenario.
- Ensure that you have specified the export horizon for the model from which you are exporting data.

Workspaces Used to Run Export Scenarios

Window Name	Navigation	Usage
Export Scenarios	• Click the Export Scenarios button in the Shortcuts bar. Select the export scenario that you want to run. Select File, Export. • Click the Models button in the Shortcuts bar. Right-click the export scenario that you want to run, and select Export.	Run export scenarios.
Models	Click the Models button in the Shortcuts bar. Right-click the model that you want to export data from and select Export.	Run export scenarios from models.

CHAPTER 13

Creating Supply Chain Planning Application Models

This chapter provides an overview of how to create Supply Chain Planning application models using data exported from Supply Chain Business Modeler and discusses how to create, manage, and run model generation scenarios.

Understanding Creating Supply Chain Planning Application Models Using Data Exported from Supply Chain Business Modeler

After exporting data from Supply Chain Business Modeler, you can use the data to create data models in EnterpriseOne Supply Chain Planning applications.

To specify how Supply Chain Planning applications use data exported from Supply Chain Business Modeler, you can create and run model generation scenarios. When you create a model generation scenario, you can select options for displaying and solving the data model in a particular Supply Chain Planning product. You can then run the model generation scenario to create a model generation rules (.mgr) file. The .mgr file is an XML document that specifies how the Supply Chain Planning product should display or use the data.

You can create model generation scenarios after you install one or more of the following EnterpriseOne Supply Chain Planning 8.12.1 products in the same scp directory as Supply Chain Business Modeler 8.12.1:

- Demand Forecasting.
- Production and Distribution Planning.
- Production Scheduling.
- Strategic Network Optimization.

In a model generation scenario, you specify:

- A name for the model generation scenario

  The name that you enter is used as the name of the model generation rules (.mgr) file.
- The Supply Chain Business Modeler model from which the data was exported.

  The data source determines which applications you can specify model generation parameters.
- The application integration profile: the application and data use.
- Model generation rules: information about how you want the Supply Chain Planning application to use the data that was exported from Supply Chain Business Modeler.

You can specify different options for each Supply Chain Planning 8.12.1 application that you install. Depending on the application that you choose in a model generation scenario, you can specify different options for using the data. For example, a Production Scheduling user could specify the day and time that their work week starts in the model, while a Strategic Network Optimization user could specify whether to optimize a model to find the maximum profit or the minimum cost.

You can create one or more model generation scenarios for each set of exported data.

After creating a model generation scenario, you can run the scenario to create an xml document that lists the configuration options that you chose. You can then call this document when running the connector for importing data into a Supply Chain Planning product.

Creating, Managing, and Running Model Generation Scenarios

This section lists prerequisites and common elements, and discusses how to:

- Create model generation scenarios.
- Modify model generation scenarios.

Prerequisites

Before you complete the tasks in this section:

- Install the destination EnterpriseOne Supply Chain Planning application in the same scp directory as Supply Chain Business Modeler.
- Create an extract area and at least one data folder for saving data files that you export from Supply Chain Business Modeler.

  See “Creating Extract Areas and Data Folders,” Setting Up Extract Areas and Data Folders.

Common Elements Used in this Section

Application integration profile Select an application integration profile for the Supply Chain Planning product where the data model will be generated using data from Supply Chain Business Modeler. Application integration profile names specify the name and version of the application where the data will be used, and indicate how the data will be used. The application integration profile that you choose determines which options you can set in the model generation scenario.

Data Folder Select the data folder where the data to use in creating the application model is saved. The model generation rules (.mgr) file will be created in this data folder when you run the model generation scenario.

Extract area Enter the name of the extract area where the exported data packages are saved. You cannot create a model generation scenario until you create an extract area.

Model Enter the name of the Supply Chain Business Modeler model from which the data is exported for creating the model. The model level determines the applications where the data can be used to create application models. For

example, data exported from a Tactical model can be used to create Production and Distribution Planning models.

Name Enter a name for the model generation scenario. The model generation rules (.mgr) file that is created when you run the model generaiton scenario has the same name as the model generation scenario. For example, if the model generation scenario name is scenario1, the resulting rules file is scenario1.mgr.

Workspaces Used to Create, Modify, and Delete Model Generation Scenarios

Window Name	Navigation	Usage
Add Model Generation Scenario	• Click the Model Generation Scenarios button in the Shortcuts bar. Select File, Add Model Generation Scenario. • Click the Model Generation Scenarios button in the Shortcuts bar. Right-click the Model Generation Scenario Navigator and select Add Model Generation Scenario.	Create model generation scenarios.
Model Generation Scenario Navigator	Click the Model Generation Scenarios button in the Shortcuts bar. Right-click the model generation scenario that you want to run, and select Execute.	Run model generation scenarios.
Modify Scenario	Click the Model Generation Scenarios button in the Shortcuts bar. In the Model Generation Scenario Navigator, right-click the export scenario and select Modify Scenario.	Modify model generation scenarios.

Page Name	Navigation	Usage
Model Generation Scenarios - Renaming	• Click the Model Generation Scenarios button in the Shortcuts bar. Select the Model Generation scenario and select Edit, Rename. • Click the Model Generation Scenarios button in the Shortcuts bar. Right-click the Model Generation scenario and select Rename.	Rename model generation scenarios.
Model Generation Scenarios - Deleting	• Click the Model Generation Scenarios button in the Shortcuts bar. Select the Model Generation scenario and select Edit, Delete. • Click the Model Generation Scenarios button in the Shortcuts bar. Right-click the Model Generation scenario and select Rename.	Delete Model Generation scenarios.

Creating Model Generation Scenarios

To create a model generation scenario:

1. In Introduction, click the Next button.
2. In Scenario Definition, complete the Name and Model fields.
3. Click the Next button.
4. In Source Data, select the extract area and data folder.
5. Click the Next button.
6. In Application Integration Profile, select an application integration scenario for the Supply Chain Planning application where the data model will be created.
7. Click the Next button.
8. Select rules for creating a data model in the destination Supply Chain Planning application.
9. In Summary, click the Finish button.

Modifying Model Generation Scenarios

Access the Modify Scenario workspace.

To modify an export scenario:

1. In Scenario Definition, complete the Name and Model fields.
2. Click the Next button.

3. In Source Data, select the extract area and data folder.
4. Click the Next button.
5. In Application Integration Profile, select an application integration profile for the Supply Chain Planning application where the data model will be created.
6. Click the Next button.
7. Select rules for creating a data model in the destination Supply Chain Planning application.
8. In Summary, click the Finish button.

CHAPTER 14

Viewing Supply Chain Business Modeler Logs

This chapter provides an overview of Supply Chain Business Modeler logs and discusses how to:

- View and clear Supply Chain Business Modeler logs.
- Change the directory where logs are saved.

Understanding Supply Chain Business Modeler Logs

When you use or install Supply Chain Business Modeler, Supply Chain Business Modeler records actions and errors that occur in its Error, Progress, and Installation logs. These logs are useful for retracing steps and solving problems that you encounter when you use or install Supply Chain Business Modeler.

To review the actions that you have performed in Supply Chain Business Modeler and determine whether any errors have occurred, you can view the following Supply Chain Business Modeler logs:

- Progress log.

  Progress logs list every action that you perform in Supply Chain Business Modeler through the graphical user interface or through the command shell. Supply Chain Business Modeler creates a separate progress log for each Supply Chain Business Modeler database that you use.

- Error log.

  Error logs list any errors that occur when you use Supply Chain Business Modeler through the graphical user interface or through the command shell. Supply Chain Business Modeler creates a separate progress log for each Supply Chain Business Modeler database that you use.

- Installation log.

  The installation log lists any errors that occur when you install Supply Chain Business Modeler.

Logs are created in text file format. You can change the directory where the progress and error logs are saved. You can view error, progress, and installation logs by opening them in a text editor.

You can display error messages from a Supply Chain Business Modeler session in a log viewer at the bottom of Supply Chain Business Modeler window. You can hide the log viewer when you do not want to view error messages. You can view the Installation and Progress logs only by opening them in a text editor.

You can clear error messages from the log viewer. If you clear the error messages from the log viewer, they remain in the error log file that Supply Chain Business Modeler saves.

By default, Supply Chain Business Modeler saves Progress and Error logs in the *path*\scp\8.12.1\scbm\logs directory in Windows or the *path*/scp/8.12.1/scbm/logs directory in UNIX. where *path* is the directory where Supply Chain Planning applications are installed. If you do not want to save the logs in this directory, you can specify a different location for the log files using the SCBM_LOGDIR environment variable.

The default maximum log file size is 10 MB. When a log file reaches 10 MB in size, Supply Chain Business Modeler backs up the log file to a separate file in the scbm\logs directory. SCBM can save 99 backup files in this directory.

Viewing and Clearing Supply Chain Business Modeler Logs

This section lists the workspaces used to view and clear Supply Chain Business Modeler logs.

Workspaces Used to View and Clear Supply Chain Business Modeler Logs

Window Name	Navigation	Usage
Supply Chain Business Modeler window	Select View, Logs.	Viewing error messages in Supply Chain Business Modeler window.
Error log	In a text editor, open the databaseName_error.log file, where databaseName is the name of a Supply Chain Business Modeler database. Error log files are saved in the path\scbm\logs directory in Windows or the path/scbm/logs directory in UNIX, where path is the directory where Supply Chain Business Modeler 8.12.1 is installed, unless a different location has been specified by the SCBM_LOGDIR environment variable.	Viewing error messages.

Page Name	Navigation	Usage
Progress log	In a text editor, open the database_progress.log file, where database is the name of a Supply Chain Business Modeler database. Progress log files are saved in the path\scbm\logs directory in Windows or the path/scbm/logs directory in UNIX, where path is the directory where Supply Chain Business Modeler 8.12.1 is installed, unless a different location has been specified by the SCBM_LOGDIR environment variable.	Viewing all Supply Chain Business Modeler actions.
Installation log	In a text editor, open the Supply_Chain_ Business_ Modeler_8.12.1_Install Log.log file from the path\scp\8.12.1\scbm directory in Windows or the path/scp/8.12.1/scbm directory in UNIX, where path is the directory where Supply Chain Business Modeler 8.12.1 is installed.	Viewing installation information.
Log viewer	Right-click the log viewer and select Clear.	Clearing messages from the log viewer.

Changing the Directory Where Logs Are Saved

To change the directory where logs are saved:

1. Select Start, Settings, Control Panel.
2. In Control Panel, double-click System.
3. In System Properties, click the Advanced tab, and then click Environment Variables.
4. In Environment Variables, click New in the System variables area.
5. In the Variable Name field, type SCBM_LOGDIR.
6. In the Variable Value field, type the path to the directory where you want to save the log files.
7. Click OK.

8. In Environment Variables, click OK.
9. In System Properties, click OK.

CHAPTER 15

Managing Supply Chain Business Modeler Databases

This chapter provides an overview of Supply Chain Business Modeler databases and discusses how to:

- Back up databases.
- Restore databases from backup files.
- Migrate databases from previous Supply Chain Business Modeler versions.

Understanding Supply Chain Business Modeler Databases

When you import data into Supply Chain Business Modeler, the data is saved in an embedded Supply Chain Business Modeler database. This database contains any data that you import into, refresh in, or publish to a Supply Chain Business Modeler model.

When you install Supply Chain Business Modeler, the system creates a database called scbm8121db in the 8.12.1 domain. When you create a new domain and add Supply Chain Business Modeler to the domain using Supply Chain Planning shell commands, a new Supply Chain Business Modeler database is created in the domain. The database has the same name as the domain. You can then use Supply Chain Planning shell commands to switch between domains with different Supply Chain Business Modeler databases.

You must ensure that each database has enough space. If a Supply Chain Business Modeler database is using 75 percent or more of its allocated space, you should add a volume to the database to prevent errors when you import data. If you do not allocate sufficient space to a database, you receive an error similar to the following when you try importing data into Supply Chain Business Modeler:

```
TerminatingE3003: NET_ENOPN: Connection is not opened (net/nettcp.c, line 1195)
```

See Also

EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for Windows

EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for UNIX

EnterpriseOne Supply Chain Business Modeler 8.12.1 Installation for Linux

Migrating Database Configurations to Supply Chain Business Modeler 8.12.1

This section provides an overview of migrating database configurations to Supply Chain Business Modeler 8.12.1 and discusses how to:

- Migrate database configurations in Windows.
- Migrate database configurations in UNIX.

Understanding Migrating Database Configurations

You can use a single command to migrate database configurations from an existing Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 installation to Supply Chain Business Modeler 8.12.1 on the same system.

Note. You can migrate database configurations from Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 installations on separate systems by using the startDatabaseBackup script from the previous Supply Chain Business Modeler version and restoring the database configurations using the startDatabaseRestore command from Supply Chain Business Modeler 8.12.1. If you back up databases using the startDatabaseBackup script from a previous Supply Chain Business Modeler version, you must manually recreate any database volumes in Supply Chain Business Modeler 8.12.1.

See Restoring Supply Chain Business Modeler Database Configurations.

When you migrate a database, the system recreates the database in Supply Chain Business Modeler 8.12.1 with all database settings and characteristics from the existing installation. The system also creates the configuration from the existing installation in the Supply Chain Business Modeler 8.12 installation, including:

- Extract areas, data folders, scripts folders, and any files within.
- Import and export scenarios.
- Model configurations and planning models.
- Calendars, changeovers, and groups.

After migrating a database to Supply Chain Business Modeler 8.12.1, you can run the import scenarios to populate Supply Chain Business Modeler with data from the data folders.

Note. You must set the period pattern in the Tactical, Strategic, and Demand models before you can refresh data in these models.

Database volumes that you want to migrate from Supply Chain Business Modeler 8.12 must have different names than the Supply Chain Business Modeler 8.12.1 database. If a Supply Chain Business Modeler 8.12 database volume has the same name as the Supply Chain Business Modeler 8.12.1 database, the system does not migrate the volume to Supply Chain Business Modeler 8.12.1.

Migrating Database Configurations in Windows

To migrate a database configuration in Windows:

Navigate to the *path*\scp\8.12.1\scbm\script\backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:


```
startDatabaseMigrate.bat databaseName domainName SCBMHomeLocation
```

Parameter	Description
databaseName	Name of the Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 database that you want to migrate to Supply Chain Business Modeler 8.12.1.
domainName	Name of the domain where you want to restore the database configuration.
SCBMHomeLocation	Location where Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 is installed, including the SCBM\vers_*x.x* subdirectory, where *x.x* is the version of Supply Chain Business Modeler where the database was created.

To migrate a Supply Chain Business Modeler 2.0 database configuration called scbmdb to the 8.12.1 domain, use the following command:

```
startDatabaseMigrate.bat scbmdb 8.12.1 c:\SCBM\vers_2.0
```

Migrating Database Configurations in UNIX

To migrate a database configuration in UNIX:

Navigate to the *path*/scp/8.12.1/scbm/script/backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
./startDatabaseMigrate.sh databaseName domainName SCBMHomeLocation
```

Parameter	Description
databaseName	Name of the Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 database that you want to migrate to Supply Chain Business Modeler 8.12.1.
domainName	Name of the domain where you want to restore the database configuration.
SCBMHomeLocation	Directory where Supply Chain Business Modeler 8.12, 8.11.1, 8.11, 8.10, or 2.0 is installed, including the SCBM/vers_*x.x* subdirectory, where *x.x* is the version of Supply Chain Business Modeler where the database was created.

To migrate a Supply Chain Business Modeler 2.0 database configuration called scbmdb to the 8.12.1 domain, use the following command:

```
./startDatabaseMigrate.sh scbmdb 8.12.1 /opt/SCBM/vers_2.0
```

Backing Up and Restoring Database Configurations

This section provides an overview of backing up and restoring Supply Chain Business Modeler databases and describes how to:

- Back up database configurations in Windows.
- Back up database configurations in UNIX.
- Restore database configurations in Windows.
- Restore database configurations in UNIX.

Understanding Backing up and Restoring Databases

You can back up Supply Chain Business Modeler 8.12.1 database settings and characteristics, along with the following configuration components:

- Extract areas, data folders, scripts folders, and any files within.
- Import, export, and model generation scenarios.
- Model configurations and planning models.
- Calendars, changeovers, and groups.

After backing up Supply Chain Business Modeler 8.12.1 databases and configurations, you can restore the configurations to the same Supply Chain Business Modeler 8.12.1 installation or to Supply Chain Business Modeler 8.12.1 installations on other machines. After restoring a database configuration to Supply Chain Business Modeler 8.12.1, you can run the import scenarios to populate Supply Chain Business Modeler with data from the data folders.

Backing up Database Configurations in Windows

Navigate to the *path*\scp\8.12.1\scbm\script\backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
startDatabaseBackup.bat domainName [-databaseLocation databaseLocation]
```

Parameter	Description
domainName	Name of the domain of the Supply Chain Business Modeler 8.12.1 database configuration that you want to back up.
-databaseLocation *databaseLocation*	(Optional.) To specify a location for saving the backup information, use the –databaseLocation option, where *databaseLocation* is the backup information directory. If you do not use the –databaseLocation option, the system saves the backup information in the *path*\scp\8.12.1\scbm\backup*domainName* directory, where *path* is the directory where Supply Chain Planning applications are installed and *domainName* is the domain name of the database that you are backing up.

For example, to save Supply Chain Business Modeler 8.12.1 database backup information from a Production domain to a c:\SCBMbackup directory, use the following command:

```
startDatabaseBackup.bat Production -databaseLocation c:\SCBMbackup
```

Backing up Database Configurations in UNIX

Navigate to the *path*/scp/8.12.1/scbm/script/backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
./startDatabaseBackup.sh domainName [-databaseLocation databaseLocation]
```

Parameter	Description
domainName	Name of the domain of the Supply Chain Business Modeler 8.12.1 database configuration that you want to back up.
–databaseLocation databaseLocation	(Optional.) To specify a location for saving the backup information, use the –databaseLocation option, where *databaseLocation* is the backup information directory. If you do not use the –databaseLocation option, the system saves the backup information in the *path*/scp/8.12.1/scbm/backup/*domainName* directory, where *path* is the directory where Supply Chain Planning applications are installed and *domainName* is the domain of the database that you are backing up.

Restoring Database Configurations in Windows

From a command prompt, navigate to the *path*\scp\8.12.1\scbm\script\backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
startDatabaseRestore.bat domainName [-overwrite | -relocate extractAreaLocation]
[-databaseLocation databaseLocation]
```

Parameter	Description
domainName	Name of the domain where you want to restore the database configuration. If you do not include the –databaseLocation option, *domainName* must be the same as the directory where backup information is stored.
–overwrite	(Optional.) To restore extract areas in the same directory where they were saved in the original Supply Chain Business Modeler configuration, include the –overwrite option.

Parameter	Description
–relocate *extractAreaLocation*	(Optional.) To specify a new location for the extract areas, include the –relocate option, where *extractAreaLocation* is the new directory for the extract areas.
–databaseLocation *databaseLocation*	(Optional.) To specify the directory of the database configuration information that you are restoring, include the –databaseLocation option, where *databaseLocation* is the backup information directory. If the database backup information is saved in the *path*\scp\8.12.1\scbm\backup*domainName* directory, where *path* is the directory where Supply Chain Planning applications are installed and *domainName* is the domain where you are restoring the database, you do not need to include the –databaseLocation option.

To restore database information from a c:\SCBM\vers_8.10\backup\scbmdb directory to a Production domain, with extract areas in a c:\scp\domains\Production\scbm_ds directory, use the following command:

```
startDatabaseRestore.bat Production relocate c:\scp\domains\Production\scbm_ds
databaseLocation c:\SCBM\vers_8.10\backup\scbmdb
```

Restoring Database Configurations in UNIX

From a command prompt, navigate to the *path*/scp/8.12.1/scbm/script/backup directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
./startDatabaseRestore.sh domainName [-overwrite | -relocate extractAreaLocation]
[-databaseLocation databaseLocation]
```

Parameter	Description
domainName	Name of the domain where you want to restore the database configuration. If you do not include the –databaseLocation option, *domainName* must be the same as the directory where backup information is stored.
–overwrite	(Optional.) To restore extract areas in the same directory where they were saved in the original Supply Chain Business Modeler configuration, include the –overwrite option.

Parameter	Description
–relocate *extractAreaLocation*	(Optional.) To specify a new location for the extract areas, include the –relocate option, where *extractAreaLocation* is the new directory for the extract areas.
–databaseLocation *databaseLocation*	(Optional.) To specify the directory of the database configuration information that you are restoring, include the –databaseLocation option, where *databaseLocation* is the backup information directory. If the database backup information is saved in the *path*/scp/8.12.1/scbm/backup/*domainName* directory, where *path* is the directory where Supply Chain Planning applications are installed and *domainName* is the domain where you are restoring the database, you do not need to include the –databaseLocation option.

CHAPTER 16

Using the Business Modeler Shell

This chapter provides an overview of the Business Modeler shell and discusses how to:

- Start the Business Modeler Shell.
- Automate business processes using scripts.

Understanding the Business Modeler Shell

The Business Modeler Shell is a command shell that enables you to run Supply Chain Business Modeler functions from a command prompt and create Tcl scripts that automate business processes. Business Modeler Shell commands extend the Tool Command Language (Tcl) shell with Supply Chain Business Modeler functions.

For example, using Business Modeler Shell commands, you can create a script that transfers data from a Supply Chain Management system to a Supply Chain Planning system, creates supply chain plans in the Supply Chain Planning system, and transfers supply chain plans to the Supply Chain Management system for implementation. You can then schedule the script to run automatically, as often as required to support business processes.

Similarly, you can create a script that transfers data from a Supply Chain Management system to EnterpriseOne Order Promising, and schedule the script to run as often as required.

Starting the Business Modeler Shell

This section discusses how to:

- Start the Business Modeler Shell in Windows.
- Start the Business Modeler Shell in UNIX.

Note. You can run only one Supply Chain Business Modeler or Business Modeler Shell session with the same database at a time. Supply Chain Business Modeler will not start if another Supply Chain Business Modeler or Business Modeler Shell session is running using the same database.

Starting the Business Modeler Shell in Windows

To start the Business Modeler Shell in Windows, do one of the following:

- Select Start, Programs, EnterpriseOne Supply Chain Planning 8.12.1, Supply Chain Business Modeler, Business Modeler Shell.

- From a command prompt, navigate to the *path*\scp\8.12.1\common\start directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

  ```
  run_scbm_bmsh.bat
  ```

  You can now enter Business Modeler Shell and other Tcl commands at the % prompt.

Starting the Business Modeler Shell in UNIX

To start the Business Modeler Shell in UNIX:

From a command prompt, navigate to the *path*/scp/8.12.1/common/start directory, where *path* is the directory where Supply Chain Planning applications are installed, and enter the following command:

```
./run_scbm_bmsh.sh
```

You can now enter Business Modeler Shell and other Tcl commands at the % prompt.

Automating Business Processes Using Scripts

This section provides an overview of scripts, lists the workspaces used to create and manage scripts, and discusses how to:

- Create scripts.
- Manage scripts.
- Run scripts.

Understanding Scripts

In addition to entering Business Modeler commands interactively in the Business Modeler Shell, you can create and run scripts of Business Modeler and other Tcl commands. Scripts enable you to link multiple functions and scenarios, and automate business processes. For example, you can create a script that does the following:

1. Extracts data from a supply chain management system.
2. Imports the data into Supply Chain Business Modeler.
3. Refreshes the data in Supply Chain Business Modeler models.
4. Exports the data from Supply Chain Business Modeler.
5. Invokes a solve in a Supply Chain Planning application.

When you create a script, you must ensure that processes occur in a logical sequence in the script. For example, you might want to import data into the Base model before you refresh data in the Operation model.

Because Business Modeler commands are an extension of the Tcl programming language, you can include both Business Modeler and other Tcl commands in scripts that you run from the Business Modeler Shell. However, if you want to specify a script as a pre- or post- script in an import or export scenario, the script cannot include Business Modeler commands and must be saved with the .tcl extension in the extract area's script folder.

You can run scripts from the Supply Chain Planning Shell, or use scheduling tools and utilities to run scripts in batch. For example, you can create a script that imports enterprise data into the Base model, and schedule the script to run monthly. You could also create a script that imports transactional data into the Base model, and schedule that script to run weekly.

The following sample script imports data into the Base model, transfers it to a planning model at the Operation model level, and exports the data for use in Production Scheduling.

Note. The # symbol in Tcl scripts indicates a comment. The Business Modeler Shell ignores lines that begin with # so that you can annotate scripts.

```
# Run an import scenario called importIntoBase that imports data
# from a supply chain management system into Supply Chain Business Modeler
bm::importScenario execute importIntoBase 2004-04-01T00:00:00
# Refresh the Operation model with data from the Base # reference model
bm::model refresh Operation
# Refresh a planning model at the Operation level called TorontoOperation
# with data from the Operation model
bm::model refresh TorontoOperation
# Run an export scenario called exportForPS that exports data from
# the TorontoOperation model for use in
# Production Scheduling
bm::exportScenario execute exportForPS
```

Workspaces Used to Create and Manage Scripts

Window Name	Navigation	Usage
Add Tcl Script	Click the Extract Areas button in the Shortcuts bar. Right-click the Scripts folder where you want to add a script and select Add Tcl Script.	Create scripts in Supply Chain Business Modeler.
Script editor	Click the Extract Areas button in the Shortcuts bar. Click the Scripts folder that contains the Tcl script. In the Detail area, right-click the script and select open.	View or edit a script.
Extract Areas	Click the Extract Areas button in the Shortcuts bar. Click the Scripts folder that contains the Tcl script. In the Detail area, right-click the script and select Rename.	Rename scripts.
Extract Areas	Click the Extract Areas button in the Shortcuts bar. Click the Scripts folder that contains the Tcl script. In the Detail area, right-click the script and select Duplicate.	Duplicate scripts.
Extract Areas	Click the Extract Areas button in the Shortcuts bar. Click the Scripts folder that contains the Tcl script. In the Detail area, right-click the script and select Delete.	Delete scripts.

Creating Scripts

Access the Add Tcl Script workspace.

To create a script:

1. In Add Tcl Script, type or paste Business Modeler and other Tcl commands, and click OK.
2. In Save As, type a name in the File Name field, and click OK.

Note. You can also create scripts in external text editors.

CHAPTER 17

Business Modeler Commands

This section lists Business Modeler discusses Business Modeler Shell Command syntax and provides detailed command information.

Business Modeler Command Syntax

The following syntax rules apply to all Business Modeler commands and parameters:

- Command and subcommand names are case-sensitive.

  Each command and subcommand begins with a lower-case character, but can contain both lowercase and uppercase characters, as specified by the command syntax.

- When you specify a path as a parameter value, use forward slashes (/) as path separators.

  The Business Modeler Shell does not recognize single backslashes as path separators. For example, use the following parameter value to specify a directory path: c:/directory1/directory2

- If a directory path, extract area, data folder, scenario, model, or calendar name includes spaces, enclose the name in quotation marks when you use it as a parameter value.

  For example, you must enclose the following path in quotation marks if you use it as a parameter value: "C:/SCBM data backup"

- When you specify a date and time as a parameter value, use the following format: yyyy-mm-ddTHH:MM:SS.

  The T separating the date and time must be uppercase. The time must be specified in 24-hour format. For example, to specify March 31, 2005 at 6:00 p.m. as a parameter value, specify: 2005-03-31T18:00:00

Business Modeler Command Reference

The following sections provide command reference information for each Business Modeler command.

calendar add

Syntax

`bm::calendar add` *`calendarName`*

Description

Use the calendar add command to create a calendar in the calendar editor.

Note. You cannot create a calendar with the name “Default”. However, you can create a calendar with the name “default” with different capitalization. For example, you can create calendars named “DEFAULT”, “default”, or “DefaulT”.

Parameters

Parameter	Description
calendarName	Name of the calendar that you want to create. If the calendar name includes blanks, the parameter value must be enclosed in quotation marks. You cannot create a calendar called Default; Supply Chain Business Modeler automatically creates a calendar named Default.

Example

The following example creates a calendar called Shift 1:

```
bm::calendar add "Shift 1"
```

See Also

“Creating Calendars that Model Resource Availability,” Creating and Managing Calendars and Components
“Creating Calendars that Model Resource Availability,” Understanding Calendars

calendar delete

Syntax

`bm::calendar delete` *`calendarName`*

Description

Use the calendar delete command to delete a calendar from the calendar editor.

Note. You cannot delete the Default calendar. However, you can delete a calendar with the name “default” with different capitalization. For example, you can delete a calendar named “DEFAULT”, “default”, or “DefaulT”.

Parameters

Parameter	Description
calendarName	Name of the calendar that you want to delete. If the calendar name includes blanks, the parameter value must be enclosed in quotation marks.

Example

The following example deletes a calendar called Shift 1:

```
bm::calendar delete "Shift 1"
```

See Also

“Creating Calendars that Model Resource Availability,” Creating and Managing Calendars and Components
“Creating Calendars that Model Resource Availability,” Understanding Calendars

calendar events

Syntax

```
bm::calendar events calendarName startDateTime endDateTime
```

Description

Use the calendar events command to obtain a list of the events that have priority in a date range in a calendar in the calendar editor and would be copied to the Base model if you were to publish the calendar.

Parameters

Parameter	Description
calendarName	Name of the calendar for which you want the list of events.
startDateTime	Start date and time of the date range in *yyyy-mm-dd*T*HH:MM:SS* format.
endDateTime	End date and time of the date range in *yyyy-mm-dd*T*HH:MM:SS* format.

Returns

Returns a list of calendar events and their properties. For each event, this list includes the following information: *{{Calendar where the event appears} {Event start date and time} {Event end date and time} Type of time that the event represents}*. If the calendar name includes blanks, the name is enclosed in curly brackets.

Example

The following example generates a list of events that have priority in a calendar called Shift 1 from January 1, 2005 at midnight to January 2, 2005 at 6:00 p.m.:

```
bm::calendar events "Shift 1" 2005-01-01T00:00:00 2005-01-02T18:00:00
```

Depending on the components, events, and event patterns that have been created in Supply Chain Business Modeler, this command returns a list of events similar to the following:

```
{{Stat Holiday} 2005-01-01T00:00:00 2005-01-02T00:00:00 Down} {{Calendar 1}
2005-01-02T00:00:00 2005-01-02T12:00:00 Uptime}
{Lunch 2005-01-02T12:00:00 2005-01-02T13:00:00 Delay}
```

See Also

"Creating Calendars that Model Resource Availability," Creating and Managing Events and Event Patterns
"Creating Calendars that Model Resource Availability," Understanding Calendars

calendar names

Syntax

```
bm::calendar names
```

Description

Use the calendar names command to obtain the names of all calendars in the calendar editor.

Returns

Returns a list of calendars in the calendar editor. If a calendar name includes spaces, it is enclosed in curly brackets in the list.

Example

If you have one calendar in Supply Chain Business Modeler named Shift 1 and one calendar named Shift2, this command returns:

```
Default {Shift 1} Shift2
```

See Also

"Creating Calendars that Model Resource Availability," Creating and Managing Calendars and Components
"Creating Calendars that Model Resource Availability," Understanding Calendars

calendar publish

Syntax

`bm::calendar publish` *`calendarName startDateTime duration durationUnit`*

Description

Use the calendar publish command to publish events from a calendar in the calendar editor to the Base model. When you publish a calendar, the system copies the list of events that results from the component events in the calendar to the Base model calendar object.

Parameters

Parameter	Description
calendarName	Name of the calendar that you want to publish. If the calendar name includes blanks, it must be enclosed in quotation marks.
startDateTime	Start date and time for publishing calendar events to the Base model. The *startDateTime* must be specified in *yyyy-mm-dd*T*HH:MM:SS* format. The time must be specified in 24-hour format.
duration	Amount of time in the calendar after the *startDateTime* for which you want to publish events to the Base model, measured in the *durationUnit*.
durationUnit	Time unit for measuring the duration. Values for *durationUnit* are Hour, Day, Week, Month, and Year.

Example

To publish 120 days of events starting on January 1, 2004 at midnight from a calendar called Shift 1 to the Base model, use the following command:

```
bm::calendar publish "Shift 1" 2004-01-01T00:00:00 120 Day
```

See Also

"Creating Calendars that Model Resource Availability," Publishing Calendars to the Base Model
"Creating Calendars that Model Resource Availability," Understanding Calendars

calendar rename

Syntax

```
bm::calendar rename oldCalendarName newCalendarName
```

Description

Use the calendar rename command to rename a calendar in Supply Chain Business Modeler calendar editor.

Note. You cannot rename the Default calendar.

Parameters

Parameter	Description
oldCalendarName	Name of the calendar that you want to rename. If the calendar name includes spaces, it must be enclosed in quotation marks.
newCalendarName	New name for the calendar. If the calendar name includes spaces, it must be enclosed in quotation marks.

Example

The following example changes the name of a calendar from Shift2 to Shift 2:

```
bm::calendar rename Shift2 "Shift 2"
```

See Also

"Creating Calendars that Model Resource Availability," Creating and Managing Calendars and Components
"Creating Calendars that Model Resource Availability," Understanding Calendars

changeover addPattern

Syntax

```
bm::changeover addPattern branch resource fromOperation toOperation duration
durationUnit [-cost cost] [-calendar calendar]
```

Description

Use the changeover addPattern command to create a changeover pattern in Supply Chain Business Modeler. When you create a changeover pattern, you must specify the branch, resource, and operations associated with the changeover. You can also specify a calendar that indicates when the changeover pattern can occur.

If you create a changeover pattern with a branch, resource, operation, or calendar value or combination of these values that is not populated in the changeover editor, the pattern is considered invalid. Invalid changeover patterns are not published to the Base model when you publish the changeover patterns. In addition, invalid changeover patterns appear in red text in the changeover editor in Supply Chain Business Modeler graphical user interface.

Invalid changeover patterns can become valid if you refresh the changeover editor with Base model data that includes the branches, resources, operations, and calendars specified in the changeover patterns.

When you create a changeover pattern, it appears at the bottom of the changeover editor list. The order of the changeover patterns in the changeover editor determines which time and cost information is applied in a changeover. If two patterns in the changeover editor apply to the same changeover at a branch, the changeover pattern that is uppermost in the changeover editor is used.

You can change the order of patterns in the changeover editor only from within Supply Chain Business Modeler graphical user interface. You cannot change the order of changeover patterns using Business Modeler commands.

Parameters

Parameter	Description
branch	Branch where the changeover occurs.
resource	Resource that is involved in the changeover. To create a changeover pattern that applies to all resources, use an asterisk as the resource value.
fromOperation	Operation that the resource is being changed from. To create a changeover pattern that applies when you are changing a resource from any operation, use an asterisk as the fromOperation value.
toOperation	Operation that the resource is being changed to. To create a changeover pattern that applies when you are changing a resource to any operation, use an asterisk as the toOperation value.
duration	Amount of time required for changing the resource from the old operation to the new operation. The duration value must be greater than or equal to zero.
durationUnit	Time unit for measuring the duration. Values for durationUnit are Second, Minute, Hour, and Day.
–cost *cost*	An optional parameter that specifies the cost associated with changing the resource from the old operation to the new operation. The cost value must be greater than or equal to zero. If you do not specify a cost, Supply Chain Business Modeler assigns a cost of zero.
–calendar *calendar*	An optional parameter that specifies a calendar in Supply Chain Business Modeler that indicates when the changeover can occur. If you do not specify a calendar, Supply Chain Business Modeler assigns Default as the calendar value.

Each changeover pattern in the changeover editor must have a unique combination of values for branch, resource, fromOperation, and toOperation. If you try to create a changeover pattern that has the same branch, resource, fromOperation, and toOperation values as an existing pattern in the changeover editor, you will receive an error message.

Any parameter value that includes spaces must be enclosed in quotation marks.

Example

To specify that changing the M-5000 machine at the Cranston branch from an operation called Mold_5500_Green to an operation called Mold_5900_LA takes one hour, has a cost of 200, and is associated with a calendar called Double Shift, use the following command:

```
bm::changeover addPattern Cranston M-5000 Mold_5500_Green Mold_5900_LA 1 Hour
-cost 200 -calendar "Double Shift"
```


To specify that a changeover for any resource at the Gunnison, from any operation to any other operation, takes 1 hour and has a cost of 200, use the following command:

```
bm::changeover addPattern Gunnison * * * 1 Hour -cost 200 -calendar "Double Shift"
```

See Also

"Creating Changeover Patterns and Groups," Creating and Managing Groups and Changeovers
"Creating Changeover Patterns and Groups," Understanding Creating and Managing Resource and Operation Groups

changeover branches

Syntax

```
bm::changeover branches
```

Description

Use the changeover branches command to obtain a list of all branches that are populated in the changeover editor. Branch information is populated in the changeover editor when you refresh the changeover editor with Base model data.

Returns

The command returns a list of branches that are populated in the changeover editor. If the branch name includes blanks, the name is enclosed in curly brackets in the list.

Example

The command returns a list of branches such as the following:

```
Cranston Gunnison
```

See Also

"Creating and Managing Changeover Patterns," Understanding Creating and Managing Changeover Patterns

changeover calendars

Syntax

```
bm::changeover calendars
```

Description

Use the changeover calendars command to obtain a list of all calendars that are populated in the changeover editor. Calendar data is populated in the changeover editor when you refresh data in the changeover and group editors.

The command returns a list of calendars that are populated in the changeover editor. If the calendar name includes blanks, the name is enclosed in curly brackets in the list.

Example

The command returns a list of calendars such as the following:

```
Default {Double Shift} {Single Shift} {Triple Shift}
```

See Also
"Creating and Managing Changeover Patterns," Understanding Creating and Managing Changeover Patterns

changeover clearPatterns

Syntax
```
bm::changeover clearPatterns
```

Description
You can use the changeover clearPatterns command to delete all changeover patterns from the changeover editor. When you delete changeover patterns from the changeover editor, branch codes, resource codes, operation codes, and calendar names that are populated in the changeover editor remain in the changeover editor.

See Also
"Creating and Managing Changeover Patterns," Workspaces Used to Create and Manage Groups and Changeover Patterns

changeover deletePattern

Syntax
`bm::changeover deletePattern` *`branch resource fromOperation toOperation`*

Description
Use the changeover deletePattern command to delete a changeover pattern from the changeover editor.

Parameters

Parameter	Description
branch	Branch where the changeover occurs.
resource	Resource that is involved in the changeover. To delete a changeover pattern that was created with an asterisk as the resource value, use an asterisk as the *resource* value.
fromOperation	Operation that the resource is being changed from. To delete a changeover pattern that was created with an asterisk as the *fromOperation* value, use an asterisk as the *fromOperation* value.
toOperation	Operation that the resource is being changed to. To delete a changeover pattern that was created with an asterisk as the *toOperation* value, use an asterisk as the *toOperation* value.

Note. The changeover deletePattern command deletes only one changeover pattern at a time. If you include an asterisk as a parameter value, only a changeover pattern that was created using an asterisk for the parameter value is deleted. For example, the following command deletes a changeover pattern at only the Gunnison branch where asterisks were specified for the resource, fromOperation and toOperation values; it does not delete all changeover patterns from the changeover editor: `bm::changeover deletePattern Gunnison * * *`

Example

To delete a changeover pattern that is defined for the M-5000 machine at the Cranston branch from an operation called Mold_5500_Green to an operation called Mold_5900_LA, use the following command:

```
bm::changeover deletePattern Cranston M-5000 Mold_5500_Green Mold_5900_LA
```

To delete a changeover pattern that is defined for the M-5000 machine at the Gunnison branch between any two operations, use the following command:

```
bm::changeover deletePattern Gunnison M-5000 * *
```

See Also

"Creating and Managing Changeover Patterns," Workspaces Used to Create and Manage Groups and Changeover Patterns

changeover operations

Syntax

`bm::changeover operations` *`branch resource`*

Description

Use the changeover operations command to obtain a list of all operations that are associated with particular resources and branches in the changeover editor. Branch, resource, and operation data is populated in the changeover editor when you refresh the changeover editor with Base model data.

Parameters

Parameter	Description
branch	Branch where the changeover occurs.
resource	Resource that is involved in the changeover. You can include an asterisk as the *resource* value to obtain a list of operations that are associated with any resources.

Example

To obtain a list of all operations that are possible for the M-5000 machine at the Gunnison branch, use the following command:

```
bm::changeover operations Gunnison M-5000
```

To obtain a list of all operations that are possible for any machines at the Toronto branch, use the following command:

```
bm::changeover operations Toronto *
```

See Also

"Creating Changeover Patterns and Groups," Understanding Creating and Managing Resource and Operation Groups

"Creating Changeover Patterns and Groups," Creating and Managing Groups and Changeover Patterns

changeover patterns

Syntax

`bm::changeover patterns [-valid]`

Description

Use the changeover patterns command to obtain a list of changeover patterns that have been created in the changeover editor.

Parameters

Parameter	Description
–valid	An optional parameter that indicates that you want a list of the valid changeover patterns in the changeover editor. Valid changeover patterns have branch, resource, operation, and calendar values, and combinations of these variables that are populated in the changeover editor. When you publish changeover patterns, only valid changeover patterns are populated in Base model. To obtain a list of all changeover patterns that are defined in the changeover editor, do not include the –valid option.

Returns

The command returns a list that includes the following information for each changeover pattern: *{branch resource fromOperation toOperation duration durationUnit cost calendar}*. Information for each changeover pattern is enclosed in curly brackets. Branch, resource, fromOperation, and toOperation names that include blanks are also enclosed in curly brackets.

Example

The command returns a changeover pattern list such as the following:

```
{Cranston M-5000 Mold_5500_Green {Mold 5500 LA} 2 Hours 100 {Double Shift}}
{Gunnison * * * 1 Hours 50 Default}
```

See Also

"Creating Changeover Patterns and Groups," Understanding Creating and Managing Resource and Operation Groups

changeover publish

Syntax

```
bm::changeover publish
```

Description

Use the changeover publish command to publish changeover patterns from the changeover editor to the Changeover Pattern object in the Base model. When you publish changeover patterns from the changeover editor, changeover patterns that were previously saved in the Base model are cleared.

If some patterns in the changeover editor are associated with branches, resources, or operations that no longer exist in the Base model, the system highlights these invalid changeover patterns in the changeover editor in Supply Chain Business Modeler graphical user interface and does not publish these changeover patterns to the Base model.

When you publish changeover patterns, the position of each changeover pattern in the changeover editor is populated as the patternPriority attribute in the Changeover Pattern object in the Base model.

See Also

"Creating Changeover Patterns and Groups," Publishing Changeover Patterns and Groups to the Base Model

changeover resources

Syntax

```
bm::changeover resources branch
```

Description

Use the changeover resources command to obtain a list of resources that are available in the changeover editor. Branch and resource data is populated in the changeover editor when you refresh the changeover editor with Base model data.

Parameters

Parameter	Description
branch	Branch where the changeover occurs.

Returns

Returns a list of resources that are populated in the changeover editor. The list includes the name and type of each resource. The resource type can be crew, tool, or machine. Information for each resource is enclosed in curly brackets. Resource names that include spaces are also enclosed in curly brackets.

Example

To obtain a list of resources associated with the Gunnison branch, you can use the following command:

```
bm::changeover resources Gunnison
```

The command returns a list of resources such as the following:

```
{{Air Ratchet} tool} {C-920 machine} {Machinist crew}
```

See Also

"Creating Changeover Patterns and Groups," Understanding Creating and Managing Resource and Operation Groups

component add

Syntax

```
bm::component add [-calendar calendarName] componentName [-pattern startDateTime
duration durationUnit eventType frequency [-endBy endDateTime]]
```

Description

Use the component add command to create an empty component or a component that includes an event pattern in Supply Chain Business Modeler. You can create a reusable component or add a component to a calendar.

Parameters

Parameter	Description
componentName	Name of the component that you want to create.
–calendar *calendarName*	An optional parameter that indicates that you want to create the component in a calendar. *calendarName* is the name of the calendar where you want to create the component. The calendar name must be enclosed in quotation marks if it includes blanks. To create a reusable component, do not include the –calendar option.
–pattern *startDateTime duration durationUnit eventType frequency* [–endBy *endDateTime*]]	An optional parameter that indicates that you want to create a recurring event pattern in the component. To create a component that does not include a recurring event, do not include the –pattern argument. Instead, use the event command to add one or more one-time events after you create an empty component.
startDateTime	Start date and time of the initial event in *yyyy-mm-dd*T*HH:MM:SS*format.
duration	Duration of the initial event in the component. The duration is measured in the time unit specified by durationUnit. Values for durationUnit are Second, Minute, Hour, and Day.
eventType	Type of time that the event represents.
frequency	Frequency with which the event recurs. Values are Daily, Weekly, Monthly, and Yearly.
–endBy *endDateTime*	To end the recurring pattern after a certain date, include the –endBy option, where *endDateTime* is the date and time when the event stops occurring in *yyyy–mm–dd*T*HH:MM:SS* format.

The following table lists values for *eventType*:

Value	Description
Down	Indicates that the event represents downtime. An example of downtime is a plant closure over a weekend. Manufacturing operations can occur only before or after a downtime event.
Delay	Indicates that the event represents delay time. An example of delay time is a daily lunch schedule for a crew. Manufacturing operations can begin before a delay time event and finish after the event.
Uptime	Indicates that the event represents normal operational time.
Premium	Indicates that the event represents time for which added cost exists, such as cost associated with an overtime shift.

Example

The following example creates a reusable component called Stat Holidays:

```
bm::component add Stat Holidays
```

The following example creates an empty component called Training in a calendar called Shift 1:

```
bm::component add -calendar "Shift 1" Training
```

The following example creates a reusable component called Lunch that includes a one-hour delay event that occurs daily starting on January 1, 2004 at 12:00 p.m.:

```
bm::component add Lunch -pattern 2004-01-01T12:00:00 1 Hour Delay Daily
```

The following example creates a component called Meetings that includes a 30 minute-long delay event that occurs weekly starting on January 1, 2004 at 10:15, and ends by June 30, 2004 at 6 p.m., in a calendar called Shift 1:

```
bm::component add -calendar "Shift 1" Meetings -pattern 2004-01-01T10:15:00 30 Minute
Delay Weekly -endBy 2004-06-01T18:00:00
```

See Also

"Creating Calendars that Model Resource Availability"

component delete

Syntax

`bm::component delete [-calendar` *`calendarName`*`]`
`componentName`

Description

Use the component delete command to delete a component in Supply Chain Business Modeler.

Parameters

Parameter	Description
componentName	Name of the component that you want to delete. If the component name includes blanks, the name must be enclosed in quotation marks.
–calendar *calendarName*	If the component is in a calendar, include the –calendar argument, where *calendarName* is the name of the calendar from which you want to delete the component. The calendar name must be enclosed in quotation marks if it includes blanks. To delete a reusable component, do not include the –calendar argument.

Example

The following example deletes a component called Training from a calendar called Shift 1:

```
bm::component delete -calendar "Shift 1" Training
```

The following example deletes a reusable component called Meetings:

```
bm::component delete Meetings
```

See Also

"Creating Calendars that Model Resource Availability"

component duplicate

Syntax

```
bm::component duplicate [-calendar srcCalendar]
srcComponent destCalendar destComponent
```

Description

You can use the component duplicate command to create a new component in a calendar by copying an existing component. You can use this command to copy a reusable component to a calendar or to copy a component from one calendar to another.

Parameters

Parameter	Description
srcComponent	Name of the component that you want to copy. If the component name includes blanks, the name must be enclosed in quotation marks.
destCalendar	Name of the calendar to which you want to copy the component. If the calendar name includes blanks, the name must be enclosed in quotation marks.
destComponent	Name of the new component. If the calendar name includes blanks, the name must be enclosed in quotation marks.
–calendar *calendarName*	To copy a component that is associated with a calendar, include the –calendar option, where *calendarName* is the name of the calendar that the component is associated with. To copy a reusable component to a calendar, do not include the –calendar option.

Example

The following example copies a reusable component called Lunch to a new component called Lunch in a calendar called Shift2:

```
bm::component duplicate Lunch Shift2 Lunch
```

The following example copies a component called Lunch from a calendar called Shift 1 to a new component called Lunch in a calendar called Shift2:

```
bm::component duplicate -calendar "Shift 1" Lunch Shift2 Lunch
```

See Also

"Creating Calendars that Model Resource Availability"

component events

Syntax

```
bm::component events [-calendar calendarName]
componentName startDateTime endDateTime
```

Description

You can use the component events command to obtain a list of one-time events and their properties in a specified date range in a component.

Parameters

Parameter	Description
componentName	Component for which you want to obtain a list of events. If the component name includes spaces, it must be enclosed in quotation marks.
startDateTime	Start of the date range for which you want a list of events in the component. The *startDateTime* must be specified in *yyyy–mm–dd*T*HH:MM:SS* format.
endDateTime	End of the date range for which you want a list of events in the component. The *startDateTime* must be specified in *yyyy–mm–dd*T*HH:MM:SS* format.
–calendar *calendarName*	If the component is part of a calendar, you must include the –calendar option, where *calendarName* is the name of the calendar that includes the component. If the component is reusable, do not include the –calendar option.

Returns

Returns a list that includes the following information for each event: {{*Component name where the event appears*} {*Event start date and time*} {*Event end date and time*} {*Type of time that the event represents*}}

Example

The following example returns a list of events between January 1, 2005 and February 1, 2005 in a reusable component called Maintenance:

```
bm::component events Maintenance 2005-01-01T00:00:00 2005-02-01T00:00:00
```

The following example returns a list of events between January 1, 2005 and February 1, 2005 in a component called Maintenance in a calendar called Shift 1:

```
bm::component events -calendar "Shift 1" Maintenance 2005-01-01T00:00:00
2005-02-01T00:00:00
```

The command returns a list of events such as the following:

```
{Maintenance 2005-01-05T18:00:00 2005-01-05T22:00:00 Down} {Maintenance
2005-01-19T18:00:00 2005-01-19T22:00:00 Down}
```

See Also

"Creating Calendars that Model Resource Availability"

component names

Syntax

```
bm::component names [-calendar calendarName]
[-recurringOnly ] [-nonRecurringOnly ]
```

Description

Use the component names command to obtain the names of components in Supply Chain Business Modeler and determine whether they include events or event patterns.

Parameters

Parameter	Description
–calendar *calendarName*	To obtain a list of the components in a particular calendar, include the –calendar option, where *calendarName* is the name of the calendar for which you want a list of components. The calendar name must be enclosed in quotation marks if it includes blanks. To obtain a list of reusable components, do not include the –calendar option.
–recurringOnly	To obtain a list of components that contain event patterns, include the –recurringOnly option.
–nonRecurringOnly	To obtain a list of components that contain one-time events, include the –nonRecurringOnly option.

To obtain a list of all components and whether they include event patterns, do not include the –recurringOnly or –nonRecurringOnly option. If you do not include the –recurringOnly or –nonRecurringOnly option, the resulting list specifies whether each component includes an event pattern or one-time events.

Returns

If you include the –recurringOnly or –nonRecurringOnly option, the command returns a component list such as the following:

```
{Stat Holidays} Training
```

If you do not include the –recurringOnly or –nonRecurringOnly option, the command returns a component list such as the following:

```
{Lunch Recurring} {{Stat Holidays} NonRecurring}
```

Example

The following example returns a list of all components in a calendar called Shift 1:

```
bm::component names -calendar "Shift 1"
```

The following example returns a list of reusable components that include event patterns:

```
bm::component names -recurringOnly
```

The following example returns a list of reusable components that include one-time events:

```
bm::component names -nonRecurringOnly
```

The following example returns a list of all components that contain event patterns in a calendar named Shift Patterns:

```
bm::component names -calendar "Shift Patterns" -recurringOnly
```

See Also

“Creating Calendars that Model Resource Availability”

component pattern

Syntax

`bm::component pattern [-calendar` *`calendarName`*`] `*`componentName`*

Description

Use the component pattern command to obtain information about an event pattern in a component.

Parameters

Parameter	Description
componentName	Name of the component with the event pattern for which you want to obtain the properties. If the component name includes spaces, it must be enclosed in quotation marks.
–calendar *calendarName*	If the component is part of a calendar, you must include the –calendar option, where *calendarName* is the name of the calendar that includes the component.

Returns

The command returns the following information for the event pattern in the component:

{Duration of the initial event in the component} {Time unit for the duration of the initial event} {Type of time that the event represents} {Frequency with which the event recurs} {Start date and time of the initial event} {Whether the event pattern stops after a certain date} {If the event pattern stops, the date and time when it stops}

Example

The following example returns information about an event pattern in a component called Lunch in a calendar called Shift 1:

```
bm::component pattern -calendar "Shift 1" Lunch
```

The command returns information such as the following:

```
1 Hour Down Daily 2003-09-01T12:00:00 false
```

Where false indicates that the event pattern continues indefinitely and does not end after a certain number of occurrences or by a certain date.

See Also

"Creating Calendars that Model Resource Availability"

component rename

Syntax

`bm::component rename [-calendar` *`calendarName`*`]`
`oldComponentName newComponentName`

Description

Use the component rename command to rename a component.

Parameters

Parameter	Description
oldComponentName	Component that you want to rename. If the component name includes spaces, it must be enclosed in quotation marks.
newComponentName	New name for the component. If the component name includes spaces, it must be enclosed in quotation marks.
–calendar *calendarName*	If the component that you want to rename is part of a calendar, include the –calendar option, where *calendarName* is the name of the calendar where the component exists. If the component that you want to rename is a reusable component, do not include the –calendar option.

Example

The following example changes the name of a component in a calendar called Shift1 from Component1 to Stat Holidays:

```
bm::component rename -calendar Shift1 Component1 "Stat Holidays"
```

The following example changes the name of a reusable component called Component1 to Stat Holidays:

```
bm::component rename Component1 "Stat Holidays"
```

See Also

"Creating Calendars that Model Resource Availability"

dataFolder add

Syntax

`bm::dataFolder add` *`extractAreaName dataFolderName`*

Description

Use the dataFolder add command to add a data folder to an extract area. Before you create a data folder, the associated extract area must exist.

Parameters

Parameter	Description
extractAreaName	Name of the extract area in which you want to create the data folder.
dataFolderName	Name of the data folder that you want to create. If the extract area or the data folder name includes blanks, it must be enclosed in quotation marks.

Example

To add a data folder called From Cranston PS to an extract area called Inbound to Operation, use the following command:

```
bm::dataFolder add "Inbound to Operation" "From Cranston PS"
```

See Also
"Managing Extract Areas and Data Folders," Creating, Renaming, and Deleting Data Folders

dataFolder delete

Syntax
```
bm::dataFolder delete extractAreaName dataFolderName
```

Description
Use the dataFolder delete command to delete a data folder from an extract area.

Parameters

Parameter	Description
extractAreaName	Name of the extract area from which you want to delete a data folder. If the extract area name includes blanks, it must be enclosed in quotation marks.
dataFolderName	Name of the data folder that you want to delete. If the data folder name includes blanks, it must be enclosed in quotation marks.

Example
The following example deletes a data folder called "To Cranston PS" from an extract area called Outbound:

```
bm::dataFolder delete Outbound "To Cranston PS"
```

See Also
"Managing Extract Areas and Data Folders," Creating, Renaming, and Deleting Data Folders

dataFolder path

Syntax
```
bm::dataFolder path extractAreaName dataFolderName
```

Description
You can use the dataFolder path command to obtain the path of a data folder.

Parameters

Parameter	Description
extractAreaName	Extract area that the data folder is associated with. If the extract area name includes blanks, it must be enclosed in quotation marks.
dataFolderName	Name of the data folder for which you want the path. If the data folder name includes blanks, it must be enclosed in quotation marks.

Returns
Depending on the location of the extract area and data folder, this command returns a path similar to the following:

```
/opt/scp/domains/8.12.1/scbm_ds/EnterpriseOne/data/inboundData
```

Example

The following example returns the path of a data folder called EnterpriseOne in an extract area called EnterpriseOne:

```
bm::dataFolder path EnterpriseOne inboundData
```

See Also

"Managing Extract Areas and Data Folders," Creating, Renaming, and Deleting Data Folders

dataFolder rename

Syntax

`bm::dataFolder rename` *`extractAreaName oldDataFolderName newDataFolderName`*

Description

Use the dataFolder rename command to rename a data folder.

Parameters

Parameter	Description
extractAreaName	Extract area that contains the data folder. If the extract area name includes blanks, it must be enclosed in quotation marks.
oldDataFolderName	Data folder that you want to rename. If the data folder name includes blanks, it must be enclosed in quotation marks.
newDataFolderName	New name for the data folder. If the data folder name includes blanks, it must be enclosed in quotation marks.

Example

To change the name of a data folder in an extract area called EnterpriseOne from outbound plans to outboundPlans, use the following command:

```
bm::dataFolder rename EnterpriseOne "outbound plans" outboundPlans
```

See Also

"Managing Extract Areas and Data Folders," Creating, Renaming, and Deleting Data Folders

displayName assignedFields

Syntax

`bm::displayName assignedFields` *`objectType`*

Description

Use the displayName assignedFields command to return all fields that have been assigned for the operation or routing display name.

Parameters

Parameter	Description
objectType	The name of the object for which you wish to list available fields. Operation and Routing are the available objects.

Example

The following example returns the fields assigned for operation display names:

```
bm::displayName assignedFields Operation
```

See Also

"Understanding Operation and Routing Display Names," Understanding Importing Data into Supply Chain Business Modeler.

displayName availableFields

Syntax

```
bm::displayName availableFields objectType
```

Description

Use the displayName availableFields command to return all available fields for a given object type.

Parameters

Parameter	Description
objectType	The name of the object for which you wish to list available fields. Operation and Routing are the available objects.

Example

The following example returns the available display name fields for Operations:

```
bm::displayName availableFields Operation
```

See Also

"Understanding Operation and Routing Display Names," Understanding Importing Data into Supply Chain Business Modeler.

displayName configure

Syntax

```
bm::displayName configure objectType fieldList [-separator separator]
```

Description

Use the displayName configure command to configure the appearance of the operation or routing display names including the assigned fields and the separator character.

Parameters

Parameter	Description
objectType	The name of the object for which you wish to list available fields. Operation and Routing are the available objects.
fieldList	The list of fields in the order you want them to be displayed in the display name.
separator	The character to be used as a separator between fields names in the display name. This parameter is optional.

Example

The following example sets the fields and separator to be included in the operations display name:

```
bm::displayName configure Operation {{Branch Plant} {Operation Sequence}} -separator ^
```

See Also

"Understanding Operation and Routing Display Names," Understanding Importing Data into Supply Chain Business Modeler.

displayName separator

Syntax

`bm::displayName separator` *`objectType`* `[-set` *`separator`*`]`

Description

Use the displayName separator command to either view or configure the separator used for the operation or routing display name. If the —set parameter is not included, this command displays the current setting.

Parameters

Parameter	Description
objectType	The name of the object for which you wish to list available fields. Operation and Routing are the available objects.
—set *separator*	The character to be used as a separator between fields names in the display name. This parameter is optional.

Example

The following example sets the separator value to be included in the operations display name:

```
bm::displayName separator Operation -set ^
```

See Also

"Understanding Operation and Routing Display Names," Understanding Importing Data into Supply Chain Business Modeler.

displayName useImportedNames

Syntax

```
bm::displayName useImportedNames objectType [-on | -off]
```

Description

Use the displayName useImportedNames command to populate the operation or routing display name from the operationName or routingName fields in the Manufacturing Package. upon import. If set to on, the display names are populated from the values in the Manufacturing package. If set to off, the display names are populated based on the displayNameConfigure command upon import.

Parameters

Parameter	Description
objectType	The name of the object for which you wish to list available fields. Operation and Routing are the available objects.

Example

The following example sets the operation display name to the contents of the operationName field in the Manufacturing package upon import:

```
bm::displayName useImportedNames Operation -on
```

See Also

"Understanding Operation and Routing Display Names," Understanding Importing Data into Supply Chain Business Modeler.

event add

Syntax

```
bm::event add [-calendar calendarName] componentName eventName
startDateTime endDateTime eventType
```

Description

Use the event add command to create a one-time event in an existing calendar component. You can add a one-time event to any component that does not include an event pattern.

You cannot add an event to a component if the time of the new event overlaps the time of an event that has already been created in the component. To add an event to a calendar that occurs at the same time as another event, you must create a new component. You cannot add an event directly to a calendar. Instead, you must add the event to a component in the calendar.

Parameters

Parameter	Description
componentName	Component to which you want to add an event. If the component name includes blanks, it must be enclosed in quotation marks.
eventName	Name of the one-time event that you want to create. If the event name includes blanks, it must be enclosed in quotation marks.
startDateTime	Start date and time of the one-time event that you are adding. The *startDateTime* must be specified in *yyyy–mm–dd*`T`*HH:MM:SS* format. The time must be specified in 24-hour format.
endDateTime	End date and time of the one-time event that you are adding. The *endDateTime* must be specified in *yyyy–mm–dd*`T`*HH:MM:SS* format. The time must be specified in 24-hour format.
eventType	Type of time that the event represents.
–calendar *calendarName*	To add an event to a component that is attached to a calendar, you must include the -calendar option, where *calendarName* is the calendar where the component exists. If the calendar name includes blanks, it must be enclosed in quotation marks.

This table lists possible values for *eventType*:

Value	Description
Down	Event represents downtime. Manufacturing operations can occur only before or after a downtime event.
Delay	Event represents delay time. Manufacturing operations can begin before a delay time event and finish after the event.
Uptime	Event represents normal operational time.
Premium	Event represents time for which added cost exists, such as cost associated with an overtime shift.

Example

The following example creates a one-time downtime event called Holiday from midnight on January 1, 2005 to midnight on January 2, 2005 in a component called Shift Patterns:

```
bm::event add "Shift Patterns" Holiday 2005-01-01T00:00:00
2005-01-02T00:00:00 Down
```

The following example adds a one-time downtime event called Training from 8:00 to 11:00 on March 1, 2005, in a component called Shift Patterns in a calendar called Calendar 1:

```
bm::event add -calendar "Calendar 1" "Shift Patterns" Training
2005-03-01T08:00:00 2005-03-01T11:00:00 Down
```

See Also

“Creating Calendars that Model Resource Availability,” Creating and Managing Events and Event Patterns

event delete

Syntax

```
bm::event delete [-calendar calendarName] componentName startDateTime
```

Description

Use the event delete command to delete a one-time event from a component.

Parameters

Parameter	Description
componentName	Component from which you are deleting an event. If the component name includes blanks, it must be enclosed in quotation marks.
startDateTime	Start of the one-time event that you are deleting. The start date and time must be specified in *yyyy–mm–dd*T*HH:MM:SS* format. The time must be specified in 24-hour format.
–calendar *calendarName*	To delete an event from a component that is attached to a calendar, include the –calendar option, where *calendarName* is the calendar where the component exists. If the calendar name includes blanks, it must be enclosed in quotation marks.

Example

To delete a one-time event that starts on March 1, 2005 at 2 p.m., from a component called Shift Patterns, use the following command:

```
bm::event delete "Shift Patterns" 2005-01-01T14:00:00
```

To delete a one-time event that starts on March 1, 2005 at 6 a.m., from a component called Shift Patterns in a calendar called Calendar 1, use the following command:

```
bm::event delete -calendar "Calendar 1" "Shift Patterns" 2005-03-01T06:00:00
```

See Also

"Creating Calendars that Model Resource Availability," Creating and Managing Events and Event Patterns

exportScenario add

Syntax

```
bm::exportScenario add scenarioName modelName extractAreaName [-preScript preScript]
[-postScript postScript] [-interfaceVersion version] [-compressed]
```

Description

Use the exportScenario add command to create an export scenario in Supply Chain Business Modeler.

Parameters

Parameter	Description
scenarioName	Name of the export scenario that you want to create. If the export scenario name includes blanks, it must be enclosed in quotation marks.
modelName	Name of the model that is associated with the export scenario. If the model name includes blanks, it must be enclosed in quotation marks.
extractAreaName	Name of the extract area that is associated with the export scenario. If the extract area name includes blanks, it must be enclosed in quotation marks.
–preScript *preScript*	(Optional) To specify a Tcl script that runs before the export, include the –preScript option, where *preScript* is the name of the Tcl script that you want to run before the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–postScript *postScript*	(Optional) To specify a Tcl script to run after the export, you must include the –postScript option, where *postScript* is the name of the Tcl script that you want to run after the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–interfaceVersion *version*	(Optional) You can use the –interfaceVersion option to specify the Supply Chain Planning XML format for exporting the data, where *version* is the version of Supply Chain Planning XML that you want to export data in. Version values are 3.3, 3.2, 3.1, 3.0, and 2.0. If you do not specify a value for the –interfaceVersion option, Supply Chain Business Modeler exports the data in Supply Chain Planning XML 3.3 format.
–compressed	(Optional) To export data in compressed gzip format, include the –compressed option.

Example

The following example creates an export scenario called Export to Cranston PS that is associated with a model called Cranston Operation and an extract area called Outbound from Supply Chain Business Modeler, runs a postscript called CranstonPSSolve, and exports Supply Chain Planning 3.3 data in compressed format:

```
bm::exportScenario add "Export to Cranston PS" Cranston Operation
"Outbound from Supply Chain Business Modeler" -postScript CranstonPSSolve -compressed
```

The following example creates an export scenario called Export to Gunnison PDP that exports data in Supply Chain Planning XML 3.1 format from a model called Gunnison Tactical to an extract area called Outbound from Supply Chain Business Modeler :

```
bm::exportScenario add "Export to Gunnison PDP" Gunnison Tactical
"Outbound from Supply Chain Business Modeler" -interfaceVersion 3.1
```

See Also

“Exporting Data from Supply Chain Business Modeler”

exportScenario addPackage

Syntax

`bm::exportScenario addPackage` *`scenarioName package dataFolder`*

Description

Use the exportScenario addPackage command to specify which data packages to export in an export scenario. Before you specify the data packages for a scenario, the scenario must already exist in Supply Chain Business Modeler.

Parameters

Parameter	Description
scenarioName	Name of the export scenario to which you want to add a data package. The scenario must already exist. If the scenario name includes blanks, it must be enclosed in quotation marks.
dataFolder	Name of the data folder where the data package is saved in the export scenario. If the data folder name includes blanks, it must be enclosed in quotation marks.
package	Name of the data package that you want to add to the export scenario. You can include only one package name in each scenario addPackage command. Valid values depend on the level of the model that is specified in the scenario, as shown in the following table:

This table shows values for *package:*

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
DemandPointHistory					X
PurchaseOrders	X	X	X	X	
SafetyParameters	X				X
SalesOrderHistory	X				

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
SalesOrders	X	X	X	X	
Sales					X
TimeSeries	X				X
TransferOrders	X	X	X	X	
WorkOrders	X	X	X	X	
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X
MasterProductionPlan	X				
NetDeploymentRequirements		X	X	X	
NetProductionRequirements		X	X	X	
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X	X	
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	

Example

The following example adds the Manufacturing data package to an export scenario called Export from Base in a data folder called cranstonData:

```
bm::exportScenario addPackage "Export from Base" Manufacturing cranstonData
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario compressed

Syntax

`bm::exportScenario compressed` *`scenarioName`*

Description

Use the exportScenario compressed command to determine whether an export scenario exports data in compressed format.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to determine whether data is exported in compressed format. If the export scenario name includes blanks, it must be enclosed in quotation marks

Returns

Returns true if data is exported in compressed format. Returns false if data is exported in uncompressed format.

Example

The following command indicates whether data exported using an export scenario called Demand Planning Data exports data in compressed or uncompressed format:

```
bm::exportScenario compressed "Demand Planning Data"
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario dataFolder

Syntax

`bm::exportScenario dataFolder` *`scenarioName packageName`*

Description

Use the exportScenario dataFolder command to obtain the name of the data folder associated with a data package in an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to obtain the data folder name. If the export scenario name includes blanks, it must be enclosed in quotation marks.
packageName	Name of the data package for which you want to obtain the data folder name. Valid values for *packageName* depend on the level of the model that is specified in the export scenario, as shown in the following table.

This table shows values for *packageName:*

Values for*PackageName*	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
DemandPointHistory					X
PurchaseOrders	X	X	X	X	
SafetyParameters	X				X
SalesOrderHistory	X				
SalesOrders	X	X	X	X	
Sales					X
TimeSeries	X				X
TransferOrders	X	X	X	X	
WorkOrders	X	X	X	X	
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X
MasterProductionPlan	X				

Values for*PackageName*	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
NetDeploymentRequirements		X	X	X	
NetProductionRequirements		X	X	X	
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X	X	
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	

Returns

Returns the data folder associated with a data package in an export scenario. For example, the command could return:

```
cranstonData
```

Example

The following example returns the data folder name for the Base data package in an export scenario named Cranston Planning Data:

```
bm::exportScenario dataFolder "Cranston Planning Data" Base
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario delete

Syntax

```
bm::exportScenario delete scenarioName
```

Description

Use the exportScenario delete command to delete an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario that you want to delete. If the scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes a scenario called Gunnison Planning Data:

```
bm::exportScenario delete "Gunnison Planning Data"
```

See Also
"Exporting Data from Supply Chain Business Modeler"

exportScenario execute

Syntax
```
bm::exportScenario execute scenarioName
```

Description
Use the exportScenario execute command to run an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the scenario that you want to run. If the scenario name includes blanks, it must be enclosed in quotation marks.

Example
The following example runs an export scenario called Gunnison Planning Data:

```
bm::exportScenario execute "Gunnison Planning Data"
```

See Also
"Exporting Data from Supply Chain Business Modeler"

exportScenario extractArea

Syntax
```
bm::exportScenario extractArea scenarioName
```

Description
Use the exportScenario extractArea command to obtain the name of the extract area associated with an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to obtain the extract area name. If the export scenario name includes blanks, it must be enclosed in quotation marks.

Returns
The command returns the name of the extract area that is associated with the export scenario. For example, the command could return the following:

```
EnterpriseOne
```


Example

The following example returns the extract area name for an export scenario called EnterpriseOne Deployment Plan:

```
bm::exportScenario extractArea "EnterpriseOne Deployment Plan"
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario interfaceVersion

Syntax

```
bm::exportScenario interfaceVersion scenarioName
[-set version]
```

Description

Use the exportScenario interfaceVersion command to set or obtain the version of Supply Chain Planning XML data that is exported when you run an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to set or obtain the interface version. If the scenario name includes blanks, it must be enclosed in quotation marks.
–set *version*	(Optional) To set the version of Supply Chain Planning XML that is exported when you run the export scenario, you must include the –set option, where *version* is the version of the Supply Chain Planning XML format that you want to export. Values for *version* are 3.3, 3.2, 3.1, 3.0 and 2.0. To determine the version of the data that is exported in an export scenario, do not include the –set option.

Example

The following example sets the XML format to Supply Chain Planning XML 3.0 in an export scenario called ExportBase:

```
bm::exportScenario interfaceVersion ExportBase -set 3.0
```

The following example returns the version of Supply Chain Planning XML that is exported from a scenario called ExportBase:

```
bm::exportScenario interfaceVersion ExportBase
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario model

Syntax

```
bm::exportScenario model scenarioName
```

Description

Use the exportScenario model command to obtain the name of the model associated with an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to obtain the model name. If the export scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the name of the model associated with an export scenario. For example, the command could return:

```
Cranston
```

Example

This example returns the model name for an export scenario named Cranston Planning Data:

```
bm::exportScenario model "Cransont Planning Data"
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario names

Syntax

```
bm::exportScenario names [-modelName modelName]
```

Description

Use the exportScenario names command to obtain a list of export scenarios that have been created in Supply Chain Business Modeler.

Parameters

Parameter	Description
–modelName *modelName*	(Optional) To obtain a list of export scenarios that are associated with a particular model, you can include the –modelName option, in which *modelName* is the name of the model for which you want a list of related export scenarios.

Returns

Returns a list of export scenarios that have been created in Supply Chain Business Modeler. Export scenario names that include blanks are enclosed in curly brackets in the list. For example, the command could return the following:

```
{Gunnison Planning Data} CranstonData
```

Example

The following example returns a list of all scenarios that export data from the Base model:


```
bm::exportScenario names -modelName Base
```

See Also
"Exporting Data from Supply Chain Business Modeler"

exportScenario packages

Syntax
`bm::exportScenario packages` *`scenarioName`*

Description

Use the exportScenario packages command to obtain the names of the data packages specified in an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to obtain the data package names. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the names of data packages specified in an export scenario. The command returns a list similar to the following:

```
Base Manufacturing Distribution Customer Calendar Supplier
```

Example

The following example returns the data package names for an export scenario named Gunnison Planning Data:

```
bm::exportScenario packages "Gunnison Planning Data"
```

See Also
"Exporting Data from Supply Chain Business Modeler"

exportScenario rename

Syntax
`bm::exportScenario rename` *`oldScenarioName newScenarioName`*

Description

Use the exportScenario rename command to rename an export scenario.

Parameters

Parameter	Description
oldScenarioName	Name of the export scenario that you want to rename. If the export scenario name includes blanks, it must be enclosed in quotation marks.
newScenarioName	New name for the export scenario. If the export scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example renames a scenario called Export 1 to ExportFromBase:

```
bm::exportScenario rename "Export 1" ExportFromBase
```

See Also

"Exporting Data from Supply Chain Business Modeler"

exportScenario script

Syntax

```
bm::exportScenario script scenarioName [-pre ] [-post ]
```

Description

Use the exportScenario script command to obtain the name of any Tcl script that is specified as a pre-script or a post-script in an export scenario.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to obtain the Tcl script names. If the export scenario name includes blanks, it must be enclosed in quotation marks.
–pre	(Optional) To obtain the name of the pre-script only, include the –pre option.
–post	(Optional) To obtain the name of the post-script only, include the –post option.

To obtain the names of both the pre-script and the post-script, you do not have to include either option.

Returns

Returns the names of pre or post scripts that are associated with an export scenario. If no pre-script or post-script is associated with a scenario, Supply Chain Business Modeler returns empty curly brackets.

For example, the command could return the following:

```
pre.tcl post.tcl
```

Example

The following example returns the names of the Tcl scripts that are associated with an export scenario called Export from Base:

```
bm::exportScenario script "Export from Base"
```

See Also
"Exporting Data from Supply Chain Business Modeler"

extractArea add

Syntax
```
bm::extractArea add extractAreaName path
```

Description

Use the extractArea add command to create an extract area in Supply Chain Business Modeler and create an associated subdirectory in an existing location on a operating system.

Parameters

Parameter	Description
extractAreaName	Extract area that you want to create. If the extract area name includes blanks, it must be enclosed in quotation marks.
path	Subdirectory that you want to create on a operating system and associate with the extract area. The path must already exist, except for the subdirectory that you want to associate with the extract area. Use forward slashes in the path regardless of the operating system. If the path includes blanks, it must be enclosed in quotation marks.

Example

The following example creates an extract area called Enterprise and an associated subdirectory called Enterprise in an existing directory called c:\scp\domains\8.12.1\scbm_ds in Windows:

```
bm::extractArea add Enterprise c:/scp/domains/8.12.1/scbm_ds
```

The following example creates an extract area called Outbound from Base and create an associated subdirectory called Outbound in an existing directory called /opt/scp/domains/8.12.1/scbm_ds/ in UNIX:

```
bm::extractArea add "Outbound from Base" /opt/scp/domains/8.12.1/scbm_ds
```

See Also
"Managing Extract Areas and Data Folders"

extractArea dataFolders

Syntax
```
bm::extractArea dataFolders extractAreaName
```

Description

Use the extractArea dataFolders command to generate a list of data folders in an extract area. Data folder names that include blanks are enclosed in curly brackets.

Parameters

Parameter	Description
extractAreaName	Extract area for which you want to obtain a list of data folders. If the extract area name includes blanks, it must be enclosed in quotation marks.

Example

To generate a list of data folders in an extract area called Inbound to SCBM, use the following command:

```
bm::extractArea dataFolders "Inbound to SCBM"
```

If the extract area includes three data folders called From Supply Chain Management, From SNO, and From Toronto PDP, the command returns:

```
{From Supply Chain Management} {From SNO} {From Toronto PDP}
```

See Also

"Managing Extract Areas and Data Folders"

extractArea delete

Syntax

`bm::extractArea delete` *`extractAreaName`*

Description

Use the extractArea delete command to delete an extract area from Supply Chain Business Modeler and its associated files and directories from a operating system.

Parameters

Parameter	Description
extractAreaName	Extract area that you want to delete. If the extract area name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes an extract area called Outbound from SCBM and its associated directories on a operating system:

```
bm::extractArea delete "Outbound from SCBM"
```

See Also

"Managing Extract Areas and Data Folders"

extractArea move

Syntax

`bm::extractArea move` *`extractAreaName path`*

Description

Use the extractArea move command to move an existing extract area and its data folders, data files, scripts folder, and scripts to a new subdirectory in an existing location on a operating system. The command also deletes the original extract area subdirectory.

Parameters

Parameter	Description
extractAreaName	Extract area that you want to associate with a new subdirectory. If the extract area name includes blanks, it must be enclosed in quotation marks.
path	Subdirectory that you want to create on a operating system and associate with the extract area. The path must already exist, except for the subdirectory that you want to associate with the extract area. Use forward slashes in the path regardless of the operating system. If the path includes blanks, it must be enclosed in quotation marks.

Example

The following example moves an extract area called Inbound, its datafolders, data files, scripts folder, and scripts to a subdirectory called Inbound in a directory called c:\scp\domains\8.12.1\scbm_ds:

```
bm::extractArea move Inbound c:/scp/domains/8.12.1/scbm_ds
/Inbound
```

The following example moves an extract area called Outbound, its datafolders, data files, scripts folder, and scripts to a subdirectory called Outbound in a directory called /opt/scp/domains/8.12.1/scbm_ds:

```
bm::extractArea move Outbound /opt/scp/domains/8.12.1/scbm_ds
/Outbound
```

See Also

"Managing Extract Areas and Data Folders"

extractArea names

Syntax

```
bm::extractArea names
```

Description

Use the extractArea names command to obtain a list of all extract areas.

Returns

Returns a list of all extract areas. Extract area names that include blanks are enclosed in curly brackets in the list.

Example

If you have an extract area called Inbound and another extract area called Outbound, this command returns:

```
{Inbound} Outbound
```

See Also

"Managing Extract Areas and Data Folders"

extractArea path

Syntax

```
bm::extractArea path extractAreaName
```

Description

Use the extractArea path command to obtain the path of an extract area.

Parameters

Parameter	Description
extractAreaName	Extract area for which you want to obtain the path. If the extract area name includes blanks, it must be enclosed in quotation marks.

Returns

Depending on the location of the extract area, this command returns a path similar to the following:

```
/scp/domains/8.12.1/scbm_ds/EnterpriseOne
```

Example

The following command obtains the directory path for an extract area called Inbound to Base:

```
bm::extractArea path "Inbound to Base"
```

See Also

"Managing Extract Areas and Data Folders"

extractArea rename

Syntax

```
bm::extractArea rename oldExtractAreaName newExtractAreaName
```

Description

Use the extractArea rename command to change the name of an extract area.

Parameters

Parameter	Description
oldExtractAreaName	Extract area that you want to rename. If the old extract area name includes blanks, it must be enclosed in quotation marks.
newExtractAreaName	New name for the extract area. If the new extract area name includes blanks, it must be enclosed in quotation marks.

Example

The following example changes the name of an extract area from Inbound to Inbound to Base, use the following command:


```
bm::extractArea rename Inbound "Inbound to Base"
```

See Also
"Managing Extract Areas and Data Folders"

extractArea scriptFolder

Syntax
`bm::extractArea scriptFolder` *`extractAreaName`*

Description

Use the extractArea scriptFolder command to obtain the scripts folder path for an extract area.

Parameters

Parameter	Description
extractAreaName	Extract area for which you want to obtain the scripts folder path. If the extract area name includes blanks, it must be enclosed in quotation marks.

Returns

Depending on the location of the extract area and the scripts folder, this command returns a path similar to the following:

```
C:\scp\domains\8.12.1\scbm_ds\Enterprise\scripts
```

Example

The following example obtains the scripts folder directory for an extract area called Outbound:

```
bm::extractArea scriptFolder Outbound
```

See Also
"Managing Extract Areas and Data Folders"

group add

Syntax
`bm::group add` *`groupType groupName`* `[-description` *`description]`*

Description

Use the group add command to create a group in the group editor. You can create the following two types of groups in the group editor:

- Resource. Resource groups can contain resources.
- Operation. Operation groups can contain operations.

After you create a group, you can add members using the group addMembers command.

Parameters

Parameter	Description
groupType	Specifies whether the group can contain resources or operations. Values for *groupType* are resource and operation.
groupName	Every group in the group editor must have a unique name, regardless of the group type. For example, you cannot have a resource group called Paint and an operation group called Paint. If the *groupName* includes blanks, it must be enclosed in quotation marks.
–description *description*	Optional parameter that provides a description of the group, where *description* is a description of the group. If the description includes blanks, it must be enclosed in quotation marks.

Example

The following example creates a resource group called Gunnison crews:

```
bm::group add resource Gunnison crews
```

The following example creates an operation group called Cranston operations with a description:

```
bm::group add operation Cranston operations -description "All
operations at Cranston location"
```

See Also

"Creating Changeover Patterns and Groups"

group addMembers

Syntax

`bm::group addMembers` *`groupName memberList`*

Description

Use the group addMembers command to add members to a group in the group editor. You can add resources to resource groups or operations to operation groups.

Parameters

Parameter	Description
groupName	If the *groupName* includes blanks, it must be enclosed in quotation marks.
memberList	List of resources or operations that you want to add to the group. All group members must have the same type as the group. That is, you can add resources to a resource group, or operations to an operation group. You must enclose the member list in curly brackets. If you are adding a resource to a resource group, you must include both the name and branch of the resource in the memberList. The resource name and branch must be separated by two colons. If the resource or branch name includes spaces, you must enclose the resource and branch name in quotation marks. If you are adding operations to a group, do not specify the branch. Operation names that include spaces must be enclosed in quotation marks.

Example

The following example adds a resource called C-121 at a branch called Cranston and a resource called C-122 at a branch called Gunnison to a resource group called Drill press:

```
bm::group addMembers "Drill press" {C-121::Cranston C-122::Gunnison}
```

The following example adds an operation called Assemble 7700 and an operation called Assemble_7711 to an operation group called Assemble:

```
bm::group addMembers Assemble {Assemble 7700 Assemble_7711}
```

See Also

"Creating Changeover Patterns and Groups"

group clear

Syntax

```
bm::group clear
```

Description

Use the group clear command to delete all groups from the group editor. When you delete all groups from the group editor, the system does not delete resources and operations that are populated in the group editor.

See Also

"Creating Changeover Patterns and Groups"

group delete

Syntax

```
bm::group delete groupName
```

Description

Use the group delete command to delete a group from the group editor. This command only deletes the group; it does not delete the members of the group from the group editor.

Parameters

Parameter	Description
groupName	If the *groupName* includes spaces, it must be enclosed in quotation marks.

Example

The following example deletes a group called Drill press:

```
bm::group delete "Drill press"
```

See Also

"Creating Changeover Patterns and Groups"

group deleteMembers

Syntax

```
bm::group deleteMembers groupName memberList
```

Description

Use the group deleteMembers command to delete group members from a group in the group editor.

Parameters

Parameter	Description
groupName	If the *groupName* includes spaces, it must be enclosed in quotation marks.
memberList	List of resources or operations that you want to delete from the group. You must enclose the member list in curly brackets. If you are deleting a resource from a resource group, you must include both the name and branch of the resource in the memberList. The resource name and branch must be separated by two colons. If the resource or branch name includes spaces, you must enclose the resource and branch name in quotation marks. If you are adding operations to a group, do not specify the branch. Operation names that include spaces must be enclosed in quotation marks.

Example

The following example deletes a resource called C-101 from the Cranston branch from a resource group called resGroup1:

```
bm::group deleteMembers resGroup1 {C-101::Cranston}
```

The following example deletes operations Assemble_7700 and Assemble_7701 from an operation group called opGroup1:

```
bm::group deleteMembers opGroup1 {Assemble_7700 Assemble_7701}
```

See Also

"Creating Changeover Patterns and Groups"

group description

Syntax

```
bm::group description groupName
```

Description

Use the group description command to obtain the description of a group in the group editor.

Parameters

Parameter	Description
groupName	If the *groupName* includes blanks, it must be enclosed in quotation marks.

Example

The following example returns the description of a group called Cranston operations:

```
bm::group description "Cranston operations"
```

See Also

"Creating Changeover Patterns and Groups"

group groups

Syntax

```
bm::group groups [-type type] [-branch branch]
```

Description

Use the group groups command to obtain a list of groups in the group editor and their types.

Parameters

Parameter	Description
–type *type*	(Optional) To obtain a list of groups with a certain type, include the –type option, where *type* is the type of group: either resource or operation.
–branch *branch*	(Optional) To obtain a list of groups that have members from only one branch, include the –branch option, where *branch* is the branch of the group members.

To obtain a list of all resource and operation groups in the group editor, do not include the –type option or the –branch option.

Returns

Returns a list that includes the name and type of each group in the group editor. Each group name and type pair is enclosed in curly brackets in the list. If a group name includes spaces, it is also enclosed in curly brackets. If you have three resource groups in the group editor called Cranston crews, Cranston machines, Cranston tools, this command returns:

```
{{Cranston crews} resource} {{Cranston machines} resource}
{{Cranston tools} resource}
```

Example

The following example returns a list of resource groups in the group editor:

```
bm::group groups -type resource
```

The following example returns a list of groups in the group editor that include members only from the Gunnison branch:

```
bm::group groups -branch Gunnison
```

See Also

"Creating Changeover Patterns and Groups"

group members

Syntax

```
bm::group members groupName
```

Description

Use the group members command to obtain a list of all members in a group in the group editor.

Parameters

Parameter	Description
groupName	If the *groupName* includes blanks, it must be enclosed in quotation marks.

Returns

Returns a list that includes the name of each member in a group in the group editor. If a member name includes spaces, it is enclosed in curly brackets in the list. If the group is a resource group, the list includes the *resourceName*::*resourceBranch* of each group member. For example, the command could return the following:

```
{{Cranston Painters} resource} {{Gunnison Painters} resource}
{Painters resource}
```

Example

To obtain a list of members in a group called Drill press, use the following command:

```
bm::group members "Drill press"
```

See Also

"Creating Changeover Patterns and Groups"

group operations

Syntax

```
bm::group operations
```

Description

Use the group operations command to obtain a list of all operations that are populated in the group editor.

Example

The command returns a list of operations such as the following:

```
Assemble_7701 Assemble_7702 Inspect_7700 Inspect_7701 Inspect_7702
```

See Also

"Creating Changeover Patterns and Groups"

group publish

Syntax

```
bm::group publish
```

Description

Use the group publish command to publish groups from the group editor to the Base model.

See Also

"Creating Changeover Patterns and Groups"

group rename

Syntax

```
bm::group rename oldGroupName newGroupName
```

Description

Use the group rename command to rename a group in the group editor.

Parameters

Parameter	Description
oldGroupName	Current name of the group. If the *oldGroupName* includes blanks, it must be enclosed in quotation marks.
newGroupName	New name for the group. If the *newGroupName* includes blanks, it must be enclosed in quotation marks.

Example

To change the name of a group from Drill presses to DrillPress, use the following command:

```
bm::group rename "Drill presses" DrillPress
```

See Also

"Creating Changeover Patterns and Groups"

group resources

Syntax

```
bm::group resources [-branch branch]
```

Description

Use the group resources command to obtain a list of groups in the group editor.

Parameters

Parameter	Description
–branch *branch*	(Optional) To obtain a list of resources at a particular branch in the group editor, include the –branch option, where *branch* is the branch where the resources are located. If the branch name includes blanks, it must be enclosed in quotation marks. To obtain a list of all resources in the group editor, do not include the –branch option.

Returns

Returns a list that includes the following information for each resource: {*resourceName::branch resourceType*}. The information for each resource is enclosed in curly brackets in the list. If a resource name includes spaces, the *resourceName::branch* information is also enclosed in curly brackets. For example, the command could return a list of resources such as the following:

```
{Assembler::Gunnison crew} {Inspector::Cranston crew} {Compressor::Gunnison tool}
{{Air Ratchet::Gunnison} tool} {C-900-1::Cranston machine} {C-999::Gunnison machine}
```

Example

The following example returns a list of resources at the Gunnison branch in the group editor:

```
bm::group resources -branch Gunnison
```

See Also

"Creating Changeover Patterns and Groups"

importScenario add

Syntax

`bm::importScenario add` *`scenarioName modelName extractAreaName`* `[-preScript` *`preScript`*`] [-postScript` *`postScript`*`] [-incremental]`

Description

Use the importScenario add command to create an import scenario in Supply Chain Business Modeler.

Parameters

Parameter	Description
scenarioName	Name of the import scenario that you want to create. If the import scenario name includes blanks, it must be enclosed in quotation marks.
modelName	Name of the model that is associated with the import scenario. If the model name includes blanks, it must be enclosed in quotation marks.
extractAreaName	Name of the extract area that is associated with the import scenario. If the extract area name includes blanks, it must be enclosed in quotation marks.
–preScript *preScript*	(Optional) To create an import scenario with a Tcl script that runs before the import, include the –preScript option, where *preScript* is the name of the Tcl script that you want to run before the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the import scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–postScript *postScript*	To specify a Tcl script to run after the import, you must include the –postScript option, where *postScript* is the name of the Tcl script that you want to run after the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–incremental	(Optional) If you want to import incremental data packages, you must use the –incremental option. If you use the –incremental option, the system merges data from the incremental data packages with data in the destination model. If you do not include the –incremental option, Supply Chain Business Modeler imports full data packages. If you import full data packages in Supply Chain Planning XML 3.3, 3.2, 3.1, or 3.0 format, the system determines which enterprise and transactional data is different in the packages than in the destination models. Supply Chain Business Modeler then imports only the new or changed enterprise and transactional data, deletes from the model any data that is not in the imported packages, and assigns default values for required data that is missing from the data packages. If you import full data packages in Supply Chain Planning XML 2.0 format, Supply Chain Business Modeler does not determine which data has changed before importing the data.

Example

The following example creates a full import scenario called Import to Cranston Operation that imports data into a planning model called Cranston Operation and is associated with an extract area called Inbound:

```
bm::importScenario add "Import to Cranston Operation" "Cranston Operation" Inbound
```

The following example creates an incremental import scenario called Import to Base that is associated with the Base model and an extract area called Inbound, and runs a prescript called Extract Supply Chain Management Data:

```
bm::importScenario add "Import to Base" Base Inbound -preScript "Extract
Supply Chain Management Data" -incremental
```

Note. When you create an import scenario, you do not need to specify the version of the Supply Chain Planning XML documents that you want to import. Supply Chain Business Modeler reads the versions of Supply Chain Planning XML documents when it imports the files.

Do not import incremental data packages using full import scenarios. If you import incremental data packages using full import scenarios, Supply Chain Business Modeler assigns default values for data that is missing from the incremental data packages.

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario addPackage

Syntax

```
bm::importScenario addPackage scenarioName package
dataFolder
```

Description

Use the importScenario addPackage command to specify which data packages to import in an import scenario. Before you specify the data packages for an import scenario, the import scenario must already exist in Supply Chain Business Modeler.

Parameters

Parameter	Description
scenarioName	Name of the import scenario to which you want to add a data package. The import scenario must already exist. If the import scenario name includes blanks, it must be enclosed in quotation marks.
dataFolder	Name of the data folder where the data package is saved in the import scenario. If the data folder name includes blanks, it must be enclosed in quotation marks.
package	Name of the data package that you want to add to the import scenario. You can include only one package name in each importScenario addPackage command. Values depend on the level of the model that is specified in the import scenario, as shown in the following table:

This table shows values for *package:*

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
CurrencyForecast	X	X	X	X	
Demand				X	
DemandPointHistory					X
Fixed Production				X	
PurchaseOrders	X	X	X	X	
SafetyParameters	X				X
SalesOrderHistory	X				
SalesOrders	X	X	X		
Sales					X
TimeSeries	X				X
TransferOrders	X	X	X	X	
WorkOrders	X	X	X		
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X
MasterProductionPlan	X				
NetDeploymentRequirements		X	X		
NetProductionRequirements		X	X		

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X		
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	
WorkOrderSchedule	X	X			

Example

The following example adds the Manufacturing data package to an import scenario called Import to Base in a data folder called From Supply Chain Management:

```
bm::importScenario addPackage "Import to Base" Manufacturing "From
Supply Chain Management"
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario dataFolder

Syntax

`bm::importScenario dataFolder` *`scenarioName packageName`*

Description

Use the importScenario dataFolder command to obtain the name of the data folder associated with a data package in an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario for which you want to obtain the data folder name. If the import scenario name includes blanks, it must be enclosed in quotation marks.
packageName	Name of the data package for which you want to obtain the data folder name. Values for *packageName* depend on the level of the model that is specified in the import scenario, as shown in the following table:

This table shows values for *package:*

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
CurrencyForecast	X	X	X	X	
Demand				X	
DemandPointHistory					X
Fixed Production				X	
PurchaseOrders	X	X	X	X	
SafetyParameters	X				X
SalesOrderHistory	X				
SalesOrders	X	X	X		
Sales					X
TimeSeries	X				X
TransferOrders	X	X	X	X	
WorkOrders	X	X	X		
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
MasterProductionPlan	X				
NetDeploymentRequirements		X	X		
NetProductionRequirements		X	X		
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X		
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	
WorkOrderSchedule	X	X			

Returns

Returns the data folder associated with a data package in an import scenario. For example, the command could return:

```
To Supply Chain Management
```

Example

The following example returns the data folder name for the Customer data package in an import scenario named Import from Supply Chain Management:

```
bm::importScenario dataFolder "Import from Supply Chain Management"
Customer
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario delete

Syntax

`bm::importScenario delete` *`scenarioName`*

Description

Use the importScenario delete command to delete an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario that you want to delete. If the import scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes an import scenario called Import from PDP:

```
bm::importScenario delete "Import from PDP"
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario execute

Syntax

`bm::importScenario execute` *`scenarioName snapshotDate`*

Description

Use the importScenario execute command to run an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario that you want to run. If the import scenario name includes blanks, it must be enclosed in quotation marks.
snapshotDate	Snapshot date in *yyyy–mm–dd*T*HH:MM:SS* format. The snapshot date is the date and time as of which all of the transactional data that you are importing is consistent and provides a synchronized view of the enterprise. For example, if a work order is partially completed as of the snapshot date, the appropriate inventory adjustments must also be made to reflect the work order consumption.

Example

The following example runs an import scenario called Import to Cranston Operation with a snapshot date of January 1, 2004 at midnight:

```
bm::importScenario execute "Import to Cranston Operation" 2004-01-01T00:00:00
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario extractArea

Syntax

`bm::importScenario extractArea` *`scenarioName`*

Description

Use the importScenario extractArea command to obtain the name of the extract area associated with an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario for which you want to obtain the extract area name. If the import scenario name includes blanks, it must be enclosed in quotation marks.

Returns

The command returns the name of the extract area that is associated with the import scenario. For example, the command could return the following:

```
Inbound to Supply Chain Business Modeler
```

Example

The following example returns the extract area name for an import scenario called ImportToOperation:

```
bm::importScenario extractArea ImportToOperation
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario incremental

Syntax

`bm::importScenario incremental` *`scenarioName`*

Description

Use the importScenario incremental command to determine whether an import scenario is a full or an incremental import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario. If the import scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns true if the import scenario is an incremental import scenario. Returns false if the import scenario is a full import scenario.

Example

The following example indicates whether an import scenario called Import to Cranston Operation is a full or an incremental import scenario:

```
bm::importScenario incremental "Import to Cranston Operation"
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario model

Syntax

```
bm::importScenario model scenarioName
```

Description

Use the importScenario model command to obtain the name of the model associated with an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario for which you want to obtain the model name. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the name of the model associated with an import scenario. For example, the command could return:

```
Gunnison Tactical
```

Example

This example returns the model name for an import scenario named Import to Gunnison PDP:

```
bm::importScenario model "Import to Gunnison PDP"
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario names

Syntax

```
bm::importScenario names [-modelName modelName]
```

Description

Use the importScenario names command to obtain a list of scenarios that have been created in Supply Chain Business Modeler.

Parameters

Parameter	Description
–modelName *modelName*	(Optional) To obtain a list of scenarios that import data into a particular model, you can include the –modelName option, in which *modelName* is the name of the model for which you want a list of related import scenarios.

Returns

Returns a list of import scenarios that have been created in Supply Chain Business Modeler. Import scenario names that include blanks are enclosed in curly brackets in the list. For example, the command could return the following:

```
{Import to Base} ImportToOperation
```

Example

The following example returns a list of all import scenarios for importing data into the Base model:

```
bm::importScenario names -modelName Base
```

The following example returns a list of all import scenarios:

```
bm::importScenario names
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario packages

Syntax

`bm::importScenario packages` *`scenarioName`*

Description

Use the importScenario packages command to obtain the names of the data packages specified in an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario for which you want to obtain the data package names. If the import scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the names of data packages specified in an import scenario. The command returns a list similar to the following:

```
Base Manufacturing Distribution Customer Calendar Supplier
```

Example

The following example returns the data package names for a scenario named Import to Base:

```
bm::importScenario packages "Import to Base"
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario rename

Syntax

`bm::importScenario rename` *`oldScenarioName newScenarioName`*

Description

Use the importScenario rename command to rename an import scenario.

Parameters

Parameter	Description
oldScenarioName	Name of the import scenario that you want to rename. If the old import scenario name includes blanks, it must be enclosed in quotation marks.
newScenarioName	New name for the import scenario. If the old import scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example renames a scenario called Import 1 to ImportToBase:

```
bm::importScenario rename "Import 1" ImportToBase
```

See Also

"Importing Data into Supply Chain Business Modeler"

importScenario script

Syntax

`bm::importScenario script` *`scenarioName`* `[-pre ] [-post ]`

Description

Use the importScenario script command to obtain the name of any Tcl script that is specified as a prescript or a postscript in an import scenario.

Parameters

Parameter	Description
scenarioName	Name of the import scenario for which you want to obtain the Tcl script names. If the import scenario name includes blanks, it must be enclosed in quotation marks.
–pre	(Optional) To obtain the name of the prescript only, include the –pre option.
–post	(Optional) To obtain the name of the postscript only, include the –post option.

To obtain the names of both the prescript and the postscript, do not include either option.

Returns

Returns the names of pre or post scripts that are associated with an import scenario. For example, the command could return the following:

```
pre.tcl post.tcl
```

Example

The following example returns the names of the Tcl scripts that are associated with a scenario called Export from Base:

```
bm::importScenario script "Export from Base"
```

See Also
"Importing Data into Supply Chain Business Modeler"

model add

Syntax
```
bm::model add  level modelName exportHorizonDuration exportHorizonDurationUnit branchList
```

Description

Use the model add command to create a planning model in Supply Chain Business Modeler. You can create a planning model at the Operation or Tactical model level.

Parameters

Parameter	Description
level	Planning level of the model that you are creating. Levels are Operation and Tactical. The level must be specified in the mixed-case format shown.
modelName	Name of the model that you want to create. If the model name includes blanks, it must be enclosed in quotation marks.
exportHorizonDuration	The length of time following the snapshot date for which data is exported from the model. The exportHorizonDuration must be an integer and is expressed in the time unit specified by *exportHorizonDurationUnit*. This value is also used as the planning horizon duration: the length of time for which you want a Supply Chain Planning system to create optimized plans after you export data from the model.
exportHorizonDurationUnit	Time unit for the *exportHorizonDuration*. Values for *exportHorizonDurationUnit* are Hour, Day, Week, Month, Quarter, and Year.
branchList	Name of a branch or list of branches in Supply Chain Business Modeler. Branch names are case-sensitive. The branch or list of branches must be enclosed in curly brackets. If a branch name includes blanks, it must be enclosed in quotation marks.

Example

The following example creates a model at the Tactical level named East Region Tactical that includes the Gunnison and Cranston branches and has an export horizon of 14 days:

```
bm::model add Tactical "East Region Tactical" 14 Day {Gunnison Cranston}
```

The following example creates a model at the Operation planning level named Cranston Operation that includes data for only the Cranston branch and has an export horizon of 1 week:

```
bm::model add Operation "Cranston Operation" 1 Week {Cranston}
```

See Also
"Setting up Models"

model branchPartition

Syntax
```
bm::model branchPartition modelName
```


Description

Use the model branchPartition command to obtain the branches that are included in a planning model in Supply Chain Business Modeler.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain the branches. If the model name includes blanks, it must be enclosed in quotation marks.

Example

To find the branches for which data is included, in a model called East Region Tactical, use the following command:

```
bm::model branchPartition "East Region Tactical"
```

See Also

"Setting up Models"

model categories

Syntax

`bm::model categories` *`modelName`*

Description

Use the model categories command to list the three categories of data in a model in Supply Chain Business Modeler.

Parameters

Parameter	Description
modelName	Name of a model. If the model name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the following list:

```
EnterpriseData TransactionalData OptimizedPlans
```

Example

The following example returns a list of the data categories in the Base model:

```
bm::model categories Base
```

See Also

"Setting up Models"

model clear

Syntax

```
bm::model clear modelName
```

Description

Use the model clear command to clear all data from a model in Supply Chain Business Modeler.

Parameters

Parameter	Description
modelName	Name of the model from which you want to clear data. If the model name includes blanks, it must be enclosed in quotation marks.

Example

To clear all data from a model called Cranston Operation, use the following command:

```
bm::model clear "Cranston Operation"
```

See Also

"Setting up Models"

model delete

Syntax

```
bm::model delete modelName
```

Description

Use the model delete command to delete a planning model in Supply Chain Business Modeler.

Parameters

Parameter	Description
modelName	Name of the model that you want to delete. If the model name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes a model called EastRegionTactical:

```
bm::model delete EastRegionTactical
```

Note. You cannot delete the Base, Operation, Tactical, Strategic, or Demand model from Supply Chain Business Modeler.

See Also

"Setting up Models"

model demandHistoryPeriods

Syntax

```
bm::model demandHistoryPeriods modelName
[-set demandHistoryPeriodCount]
```

Description

Use the model demandHistoryPeriods command to obtain or set the number of periods prior to the snapshot date for which demand history data is populated in the Demand model when you refresh Demand model data.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain or set the number of periods of demand history data. The value is Demand.
–set *demandHistoryPeriodCount*	(Optional) To set the number of periods prior to the snapshot date for which demand history data is populated in the Demand model, include the –set parameter, where *demandHistoryPeriodCount* is the number of periods prior to the snapshot date. The *demandHistoryPeriodCount* is measured in the period unit for the model.

Example

To set the number of periods of demand history data prior to the snapshot date in the Demand model to six, you can use the following command:

```
bm::model demandHistoryPeriods Demand -set 6
```

See Also

"Setting up Models"

model dependentPackages

Syntax

```
bm::model dependentPackages modelName packageName
```

Description

Use the model dependentPackages command to obtain a list of packages that are dependent on a specified data package in a model. If a package does not have any dependent packages, the command does not return any information.

Parameters

Parameter	Description
modelName	Name of the model where the package exists. If the model name includes blanks, it must be enclosed in quotation marks.
packageName	Name of the package for which you want to determine the dependent packages.

Returns

Returns a list of packages, if any, that are dependent on a specified data package in a model. For example, the command could return the following:

```
SourcingRecommendations
```

Example

The following example returns the dependent packages of the Distribution package in the Strategic model:

```
bm::model dependentPackages Strategic Distribution
```

The following example returns the dependent packages of the Purchase Plan package in the Base model:

```
bm::model dependentPackages Base PurchasePlan
```

See Also

"Setting up Models"

model exportHorizon

Syntax

bm::model exportHorizon *modelName* [-set *exportHorizonDuration exportHorizonDurationUnit*]

Description

Use the model exportHorizon command to obtain or set a model's export horizon duration and planning horizon duration. A model's export horizon duration is the length of time following the snapshot date for which data is exported from the model. A model's planning horizon duration is the length of time following the snapshot date for which you want a Supply Chain Planning system to create optimized plans after you export the data from Supply Chain Business Modeler.

The model exportHorizon command sets the same value for a model's export horizon duration and planning horizon duration. To specify different values for the export horizon duration, planning horizon duration, and period duration unit, you must use the model horizon command or enter values in the Supply Chain Business Modeler graphical user interface.

Note. The model exportHorizon command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. To obtain or set a model's export horizon duration and planning horizon duration, use the model horizon command. The model horizon command enables you to set different durations for the export horizon and planning horizon.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain or set the export horizon. If the model name includes blanks, it must be enclosed in quotation marks.
–set *exportHorizonDuration exportHorizonDurationUnit*	(Optional) To set a model's export horizon duration and planning horizon duration, you must include the –set option, where *exportHorizonDuration* is an integer that represents the length of time following the snapshot date for the export horizon and the planning horizon. *exportHorizonDurationUnit* is the unit of measure for the *exportHorizonDuration*. Values for *exportHorizonDurationUnit* are Hour, Day, Week, Month, Quarter, and Year.

Example

The following example returns the current export horizon of a planning model at the Operation level named GunnisonOperation:

```
bm::model exportHorizon GunnisonOperation
```

The following example sets the export horizon and planning horizon of a planning model at the Operation level named GunnisonOperation to seven days:

```
bm::model exportHorizon GunnisonOperation -set 7 Day
```

See Also

"Setting up Models"

model hierarchy

Syntax

`bm::model hierarchy` *`modelName hierarchyType`* `[[-set` *`hierarchyLevelName`*`]|[-list]]`

Description

Use the model hierarchy command to obtain or set the level of customer aggregation in the Demand model, or to obtain all possible aggregation levels.

Parameters

Parameter	Description
modelName	Name of the model where you want to obtain or set the customer aggregation level. The value for *modelName* is Demand.
hierarchyType	Type of hierarchy for which you want to aggregate data. The value for *hierarchyType* is Customer.
–set *hierarchyLevelName*	(Optional) To set the customer aggregation level in the Demand model, include the –set parameter, where *hierarchyLevelName*is a customer aggregation hierarchy level that is populated in the Base model. **Note.** You must enter the *hierarchyLevelName* value exactly as it is specified in the customerAggregationHierarchy attribute in the Customer data package.
–list	(Optional) To obtain a list of possible customer aggregation levels in the Demand model, include the –list parameter. **Note.** If you do not include the –list or –set option, the command returns the current customer aggregation level that is set in the Demand model.

Example

To determine the customer aggregation level in the Demand model, use the following command:

```
bm::model hierarchy Demand Customer
```

To set the customer aggregation level in the Demand model to Region, use the following command:

```
bm::model hierarchy Demand Customer -set Region
```

To obtain a list of possible customer aggregation levels in the Demand model, use the following command:

```
bm::model hierarchy Demand Customer -list
```

See Also

"Setting up Models"

model horizon

Syntax

`bm::model horizon` *`modelName`* `[-set` *`planningHorizonDuration exportHorizonDuration horizonDurationUnit`*`]`

Description

Use the model horizon command to obtain or set a model's export horizon duration and planning horizon duration. A model's export horizon duration is the length of time for which data is exported from the model. A model's planning horizon is the length of time for which you want a Supply Chain Planning system to create optimized plans after you export the data from Supply Chain Business Modeler.

Note. In Supply Chain Business Modeler 8.11, the model horizon command set the period unit to the same value as the horizonDurationUnit for Strategic and Demand models. In Supply Chain Business Modeler 8.12.1, the model horizon command does not set the period unit. To set the period unit, use the model periodPattern command.

The export horizon duration must be greater than or equal to the planning horizon duration.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain or set the export horizon. If the model name includes blanks, it must be enclosed in quotation marks.
–set *planningHorizonDuration exportHorizonDuration horizonDurationUnit*	(Optional) To set a model's export horizon and planning horizon, you must include the –set option, where *planningHorizonDuration* is the length of time following the snapshot date for which you want a Supply Chain Planning system to create optimized plans using the data and *exportHorizonDuration* is the length of the time following the snapshot date for which you want to export data from the Supply Chain Business Modeler model. Values for *horizonDurationUnit* are Hour, Day, Week, Month, Quarter, and Year.

Example

The following example returns the current export horizon duration and planning horizon duration of an Operation model named GunnisonOperation:

```
bm::model horizon GunnisonOperation
```

The following example sets the planning horizon duration to 7 days and the export horizon duration to 10 days for a planning model at the Operation level called GunnisonOperation:

```
bm::model horizon GunnisonOperation -set 7 10 Day
```

See Also

"Business Modeler Commands," model periodPattern
"Setting up Models"

model level

Syntax

`bm::model level` *`modelName`*

Description

Use the model level command to obtain the planning level of a model in Supply Chain Business Modeler.

Note. Be careful not to confuse this command with the *model levels* command.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain the planning level. If the model name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the planning level of the specified model. For example, the command could return the following:

```
Tactical
```

Example

The following example returns the planning level of a model called East Region:

```
bm::model level "East Region"
```

See Also

"Setting up Models"

model levels

Syntax

```
bm::model levels
```

Description

Use the model levels command to list the levels in the model hierarchy in Supply Chain Business Modeler. Model levels in Supply Chain Business Modeler are Base, Operation, Tactical, Strategic, and Demand.

Note. Be careful not to confuse this command with the *model level* command.

See Also

"Setting up Models"

model marshallHorizon

Syntax

```
bm::model marshallHorizon Base [-set shortDuration shortDurationUnit
mediumDuration mediumDurationUnit]
```

Description

Use the model marshallHorizon command to obtain or set horizon lengths for marshalling supply chain plans. You must set the time fence dates before you can publish optimized plans from the Operation model to the Base model.

Parameters

Parameter	Description
–set	Optional parameter. To set the plan marshalling horizon durations, you must include the -set option. To obtain a list of the horizon durations that are currently defined in Supply Chain Business Modeler, do not include the -set option.
shortDuration	Amount of time following the snapshot date for which you want to include production schedules in the marshalled plans in the Base model.
shortDurationUnit	Time unit for measuring the shortDuration. Values for *shortDurationUnit* are Hour, Day, Week, Month, and Year.
mediumDuration	Amount of time following the end of the short plan marshalling horizon for which you want to include net production requirements in the marshalled plans in the Base model.
mediumDurationUnit	Time unit for measuring the mediumDuration. Values for *mediumDurationUnit* are Hour, Day, Week, Month, and Year.

Example

To set the short plan marshalling horizon for 14 days and the medium plan marshalling horizon to 6 weeks after the end of the short plan marshalling horizon, use the following command:

```
bm::model marshallHorizon Base -set 14 Day 6 Week
```

See Also

"Setting up Models"

model names

Syntax

`bm::model names` *`level`*

Description

Use the model names command to obtain a list of planning models at a particular planning level in Supply Chain Business Modeler.

Parameters

Parameter	Description
level	Planning level for which you want to obtain the model names. Valid levels are Operation and Tactical. The level must be specified in the mixed-case format shown.

Returns

Returns a list of planning models at a particular planning level. For example, if you have three planning models at the Operation level named Gunnison Operation, Cranston, and Toronto Operation, this command returns:

```
{Gunnison Operation} Cranston {Toronto Operation}
```

Example

To obtain the names of all Operation level planning models, use the following command:

```
bm::model names Operation
```

See Also

"Setting up Models"

model packages

Syntax

```
bm::model packages modelName [-category categoryName]
```

Description

Use the model packages command to obtain a list of data packages in a model.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain a list of data packages. If the model name includes blanks, it must be enclosed in quotation marks.
–category *categoryName*	(Optional) To obtain the data packages in a model for a particular category of data, include the –category option, where *categoryName* is the data category name. Category name values are EnterpriseData, TransactionalData, and OptimizedPlans.

Returns

Returns a list of data packages in a model. For example, the command could return the following list:

```
Configuration BeginningInventory SalesOrders PurchaseOrders TransferOrders WorkOrders
```

Example

The following example returns a list of the data packages in the Demand model:

```
bm::model packages Demand
```

The following example returns a list of all Transactional Data packages in an operation model named Gunnison Operation:

```
bm::model packages "Gunnison Operation" -category TransactionalData
```

See Also

"Setting up Models," Understanding Data Packages

model periodDuration

Syntax

```
bm::model periodDuration modelName
```

Description

Use the model periodDuration command to obtain the time period type in a model.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain the time period type. If the model name includes blanks, it must be enclosed in quotation marks. Values for *modelName* are Strategic and Demand.

Example

To obtain the period duration for the Demand model, use the following command:

```
bm::model periodDuration Demand
```

See Also

"Setting up Models"

model periodPattern

Syntax

`bm::model periodPattern` *`modelName`* `[-set` *`periodUnit alignmentDate`*`]`

Description

Use the model periodPattern command to obtain or set the period type in a model and the date when one time period in the model begins.

Parameters

Parameter	Description
model	Name of the model for which you want to obtain or set the period duration. Values for *modelName* are Tactical, Strategic and Demand.
–set *periodUnit alignmentDate*	(Optional) To set a model's period unit and period pattern alignment date, you must include the –set option, where *periodUnit* is the time period type and *alignmentDate* is the start date and time of one time period in a model. For the Tactical model, the value for *periodUnit* is Day. For the Strategic model, values for *periodUnit* are Day, Week, Month, Fiscal Month 445, Fiscal Month 4 Weeks, Quarter, Year. For the Demand model, values for *periodUnit* are Day, Week, Month, Fiscal Month 445, Fiscal Month 4 Weeks, and Year. If the period unit includes blanks, it must be enclosed in quotation marks. The *alignmentDate* must be in *yyyy-mm-dd*T*HH:MM:SS* format.

Returns

If you do not include the –set option, the command returns the specified model's period unit and period pattern alignment date. For example, the command could return the following:

```
Month 2005-01-01T00:00:00
```

Example

To obtain the period pattern in the Strategic model, use the following command:

```
bm::model periodPattern Strategic
```

To set the period type to Day and the period pattern start to January 1, 2005 at midnight in the Tactical model, use the following command:

```
bm::model periodPattern Tactical -set Day 2005-01-01T00:00:00
```

See Also

"Setting up Models"

model publish

Syntax

`bm::model publish` *modelName*

Description

Use the model publish command to publish data from a model to a connected model. Using this command, you can transfer data from a reference model to the reference model below it in the model hierarchy or from a planning model to the reference model at the same level in the hierarchy.

Parameters

Parameter	Description
modelName	Name of the model from which you want to publish data. If the model name includes blanks, it must be enclosed in quotation marks.

Example

The following example transfers data from a planning model called Gunnison Operation to the Operation model:

```
bm::model publish "Gunnison Operation"
```

See Also

"Publishing Data from Models"

model refresh

Syntax

`bm::model refresh` *modelName*

Description

Use the model refresh command to refresh the model with data from a connected model. You can use this command to transfer data from a reference model to the reference model above it in the model hierarchy or from a reference model to the planning model at the same level in the hierarchy.

Parameters

Parameter	Description
modelName	Name of the model to which you want to transfer data. If the model name includes blanks, it must be enclosed in quotation marks.

Example

The following example transfers data from the Operation model to a planning model at the Operation level called CranstonOperation:

```
bm::model refresh CranstonOperation
```

See Also

"Refreshing Data in Models"

model rename

Syntax

`bm::model rename` *`oldModelName newModelName`*

Description

Use the model rename command to rename a model in Supply Chain Business Modeler.

Parameters

Parameter	Description
oldModelName	Name of the model that you want to rename. If the model name includes blanks, it must be enclosed in quotation marks.
newModelName	New name for the model. If the model name includes blanks, it must be enclosed in quotation marks.

Example

The following example changes the name of a model from "East Region" to "East Region Tactical":

```
bm::model rename "East Region" "East Region Tactical"
```

See Also

"Setting Up Models"

model snapshotDate

Syntax

`bm::model snapshotDate` *`modelName`*

Description

Use the model snapshotDate command to obtain the snapshot date of the data in a model in Supply Chain Business Modeler. The snapshot date is specified when data is imported into Supply Chain Business Modeler, is saved in the model with the data, and populated in models when you refresh or publish the data.

The snapshot date is the date as of which all transactional data is assumed to be consistent and a synchronized view of the enterprise exists. For example, if a work order is partially completed as of the snapshot date, the appropriate inventory adjustments must be made to reflect the consumption on the work order by the snapshot date. You must specify the snapshot date when you import data into Supply Chain Business Modeler.

Parameters

Parameter	Description
modelName	Name of the model for which you want to obtain the snapshot date. If the model name includes blanks, it must be enclosed in quotation marks.

Example

The following example returns the snapshot date of the data in the Operation model:

```
bm::model snapshotDate Operation
```

See Also

"Setting Up Models"

modelGenerationScenario add

Syntax

```
bm::modelGenerationScenario add scenarioName modelName
extractAreaName dataFolderName
applicationIntegrationProfileName
```

Description

Use the modelGenerationScenario add command to create a model generation scenario in Supply Chain Business Modeler.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario that you want to create. If the scenario name includes blanks, it must be enclosed in quotation marks.
modelName	Name of the model that is associated with the model generation scenario. If the model name includes blanks, it must be enclosed in quotation marks.
extractAreaName	Name of the extract area that is associated with the model generation scenario. If the extract area name includes blanks, it must be enclosed in quotation marks.
dataFolderName	Name of the data folder that is associated with the model generation scenario. If the data folder name includes blanks, it must be enclosed in quotation marks.
applicationIntegration ProfileName	The profile that specifies the Supply Chain Planning application and version where the model will be generated. If the application integration profile name includes blanks, it must be enclosed in quotation marks.

Example

The following example creates a model generation scenario called Cranston PS that creates a data model using data from a planning model called Cranston Operation, is associated with an extract area called Outbound Operation and a data folder called Cranston Production Scheduling, and uses an application integration profile called Production Scheduling::8.12.1::full_refresh

```
bm::modelGenerationScenario add "Cranston PS" "Cranston Operation"
"Outbound Operation" "Cranston Production Scheduling"
"Production Scheduling::8.12.1::full_refresh"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario dataFolder

Syntax

`bm::modelGenerationScenario dataFolder` *`scenarioName`*

Description

Use the modelGenerationScenario dataFolder command to obtain the name of the data folder associated with a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario for which you want to obtain the data folder name. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the data folder associated with a data package in a model generation scenario. For example, the command could return:

```
To Supply Chain Management
```

Example

The following example returns the data folder name in a scenario named Cranston SNO Model:

```
bm::modelGenerationScenario dataFolder "Cranston SNO Model"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario delete

Syntax

`bm::modelGenerationScenario delete` *`scenarioName`*

Description

Use the modelGenerationScenario delete command to delete a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario that you want to delete. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes a model generation scenario called Gunnison Scheduling:

```
bm::modelGenerationScenario delete "Gunnison Scheduling"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario execute

Syntax

`bm::modelGenerationScenario execute` *`scenarioName`*

Description

Use the modelGenerationScenario execute command to run a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario that you want to run. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example runs a model generation scenario called Cranston scheduling:

```
bm::modelGenerationScenario execute "Cranston scheduling"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario extractArea

Syntax

`bm::modelGenerationScenario extractArea` *`scenarioName`*

Description

Use the modelGenerationScenario extractArea command to obtain the name of the extract area associated with a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario for which you want to obtain the extract area name. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.

Returns

The command returns the name of the extract area that is associated with the model generation scenario. For example, the command could return the following:

```
Gunnison Strategic
```

Example

The following example returns the extract area name for a model generation scenario called Gunnison Strategic, use the following command:

```
bm::modelGenerationScenario extractArea "Gunnison Strategic"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario model

Syntax

```
bm::modelGenerationScenario model scenarioName
```

Description

Use the modelGenerationScenario model command to obtain the name of the model associated with a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario for which you want to obtain the model name. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the name of the model associated with a model generation scenario. For example, the command could return:

```
GunnisonTactical
```

Example

This example returns the model name associated with a model generation scenario named GunnisonTactical:

```
bm::modelGenerationScenario model "GunnisonTactical"
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario names

Syntax

```
bm::modelGenerationScenario names [-modelName modelName]
```

Description

Use the modelGenerationScenario names command to obtain a list of model generation scenarios that have been created in Supply Chain Business Modeler.

Parameters

Parameter	Description
–modelName *modelName*	(Optional) To obtain a list of model generation scenarios associated with a particular model, you can include the –modelName option, in which *modelName* is the name of the model for which you want a list of model generation scenarios.

Returns

Returns a list of model generation scenarios that have been created in Supply Chain Business Modeler. Model generation scenario names that include blanks are enclosed in curly brackets in the list. For example, the command could return the following:

```
{Cranston Scheduling} GunnisonScheduling
```

Example

The following example returns a list of all model generation scenarios:

```
bm::modelGenerationScenario names
```

The following example returns a list of all model generation scenarios that use data exported from the Strategic model:

```
bm::modelGenerationScenario names -modelName Strategic
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario profile

Syntax

`bm::modelGenerationScenario profile` *`scenarioName`*

Description

Use the modelGenerationScenario profile command to obtain the name of the application integration profile that is specified in a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario for which you want to obtain the application integration profile name.

Returns

Returns the name of the application integration profile that is associated with a model generation scenario. For example, the command could return the following:

```
{Production Scheduling::8.12.1::full refresh}
```

Example

The following example returns the names of the application integration profile that is associated with a model generation scenario called GunnisonProductionPlanning:

```
bm::modelGenerationScenario profile GunnisonProductionPlanning
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario profiles

Syntax

```
bm::modelGenerationScenario profiles planningLevel
```

Description

Use the modelGenerationScenario profiles command to obtain available application integration profile names. Application integration profile names include the name and version of the Supply Chain Planning application where the model is to be generated, and indicate the way that data is used in the application.

Application integration profiles become available when you install Supply Chain Planning products such as Strategic Network Optimization, Production and Distribution Planning, and Production Scheduling in the same scp directory as Supply Chain Business Modeler.

Parameters

Parameter	Description
planningLevel	(Optional) To obtain a list of available application integration profiles for a specific planning level, include the *planningLevel* option. Values are Base, Operation, Tactical, Strategic, and Demand.

Returns

Returns the names of available application integration profiles.

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario rename

Syntax

```
bm::modelGenerationScenario
 rename oldScenarioName newScenarioName
```

Description

Use the modelGenerationScenario rename command to rename a model generation scenario.

Parameters

Parameter	Description
oldScenarioName	Name of the model generation scenario that you want to rename. If the old model generation scenario name includes blanks, it must be enclosed in quotation marks.
newScenarioName	New name for the model generation scenario. If the old model generation scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example renames a scenario called Scheduling to Gunnison Scheduling:

```
bm::modelGenerationScenario rename Scheduling "Gunnison Scheduling"
```

See Also
"Creating Supply Chain Planning Application Models"

modelGenerationScenario rule

Syntax
```
bm::modelGenerationScenario rule scenarioName ruleName
[-set value]
```

Description
Use the modelGenerationScenario rule command to view or set the value for a rule in a model generation scenario.

Parameters

Parameter	Description
scenarioName	Name of the model generation scenario for which you want to view or set a rule value. If the model generation scenario name includes blanks, it must be enclosed in quotation marks.
ruleName	Name of the rule for which you want to view or set a value. The rule name consists of page names and any parameter name, separated by periods.
–set *value*	(Optional) To set a value for a rule, include the –set option, where *value* is the value that you want to set for a rule.

Example
The following example returns the value for a rule called psParameters.demandDetails.globalBuildStrategy in a model generation scenario called ProductionScheduling:

```
bm::modelGenerationScenario rule ProductionScheduling
 psParameters.demandDetails.globalBuildStrategy
```

The following example sets the value for a rule called psParameters.demandDetails.globalBuildStrategy to Pre-Build in a model generation scenario called ProductionScheduling:

```
bm::modelGenerationScenario rule ProductionScheduling
psParameters.demandDetails.globalBuildStrategy -set Pre-Build
```

See Also
"Creating Supply Chain Planning Application Models"

modelGenerationScenario rules

Syntax
```
bm::modelGenerationScenario rules profileName [-newLine]
```

Description

Use the modelGenerationScenario rules command to view all possible rules for a model generation scenario for a particular application integration profile. The application integration profile includes the name and version of the Supply Chain Planning application where the model is to be generated, and indicates the way that the data is used in the application.

Parameters

Parameter	Description
profileName	Name of the application integration profile for which you want to view all possible rules. If the application integration profile name includes blanks, it must be enclosed in quotation marks.
–newLine	If you want each possible model generation rule to appear on a separate line, include the –newLine option. If you do not include the –newLine option, the model generation scenario rules appear in one line, separated by spaces.

Example

The following command returns all possible rules in the SNO::8.12.1:Refresh All Packages profile, with each rule on a separate line:

```
bm::modelGenerationScenario rules
"SNO::8.12.1::Refresh All Packages" -newline
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario ruleType

Syntax

`bm::modelGenerationScenario ruleType` *`profileName ruleName`*

Description

Use the modelGenerationScenario ruleType command to determine the rule type of a model generation rule in an application integration profile. Possible rule types include:

- enum.

  Rules of this type have two or more possible values.
- boolean
- string
- integer
- double
- date
- time

Parameters

Parameter	Description
profileName	Name of the application integration profile where the rule exists. If the application integration profile name includes blanks, it must be enclosed in quotation marks.
ruleName	Name of the model generation rule for which you want to obtain the type.

Returns

Returns the type of the model generation rule. If the rule type is enum, the command also returns all possible values for the rule.

Example

The following command returns all possible rules in the SNO::8.12.1:Strategic Refresh All Packages profile, with each rule on a separate line:

```
bm::modelGenerationScenario rules
"SNO::8.12.1::Strategic Refresh All Packages" -newline
```

See Also

"Creating Supply Chain Planning Application Models"

modelGenerationScenario rename

Syntax

bm::modelGenerationScenario rename *oldScenarioName newScenarioName*

Description

Use the modelGenerationScenario rename command to change the name of a model generation scenario.

Parameters

Parameter	Description
oldScenarioName	Name of the model generation scenario that you want to rename. If the old scenario name includes blanks, it must be enclosed in quotation marks.
newScenarioName	New name for the model generation scenario. If the new scenario name includes blanks, it must be enclosed in quotation marks.

See Also

"Creating Supply Chain Planning Application Models"

scenario add

Syntax

```
bm::scenario add scenarioType scenarioName modelName extractAreaName
[-preScript preScript] [-postScript postScript] [-interfaceVersion
version] [-incremental] [-compressed]
```

Description

Use the scenario add command to create an import or export scenario in Supply Chain Business Modeler.

Note. The scenario add command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario add command, you can use the importScenario add command or the exportScenario add command.

Parameters

Parameter	Description
scenarioType	Type of scenario that you want to create. Values for scenarioType are import and export.
scenarioName	Name of the scenario that you want to create. If the scenario name includes blanks, it must be enclosed in quotation marks.
modelName	Name of the model that is associated with the scenario. If the model name includes blanks, it must be enclosed in quotation marks.
extractAreaName	Name of the extract area that is associated with the scenario. If the extract area name includes blanks, it must be enclosed in quotation marks.
–preScript *preScript*	(Optional) To create a scenario with a Tcl script that runs before the import or export, include the –preScript option, where *preScript* is the name of the Tcl script that you want to run before the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–postScript *postScript*	(Optional) To specify a Tcl script to run after the import or export, you must include the –postScript option, where *postScript* is the name of the Tcl script that you want to run after the scenario. The Tcl script must be saved in the scripts directory of the extract area associated with the scenario. If the script name includes blanks, it must be enclosed in quotation marks.
–incremental	(Optional) If you are creating an import scenario and want to import incremental data packages, you must use the –incremental option. If you use the –incremental option, the system imports incremental data packages and merges the data with existing data in the destination model. If you do not include the –incremental option, Supply Chain Business Modeler imports full data packages. If you import full data packages in Supply Chain Planning 3.3, 3.2, 3.1, or 3.0 format, the system determines which enterprise and transactional data is different in the packages than in the destination models. Supply Chain Business Modeler then imports only the new or changed enterprise and transactional data, deletes from the model any data that is not in the imported packages, and assigns default values for required data that is missing from the data packages. If you import full data packages in Supply Chain Planning XML 2.0 format, Supply Chain Business Modeler does not determine which data has changed before importing the data.
–interfaceVersion *version*	(Optional) If you are creating an export scenario, you can use the –interfaceVersion option to specify the Supply Chain Planning XML format for exporting the data, where*version*is the version of Supply Chain Planning XML that you want to export data in. Versions are 3.3, 3.2, 3.1, 3.0, and 2.0. If you do not specify a value for the –interfaceVersion option, Supply Chain Business Modeler exports the data in Supply Chain Planning XML 3.3 format. **Note.** When you create an import scenario, you do not need to specify the version of the Supply Chain Planning XML documents that you want to import. Supply Chain Business Modeler reads the versions of Supply Chain Planning XML documents when it imports the files.
–compressed	(Optional) To export data in compressed gzip format in an export scenario, include the –compressed option.

Example

The following example creates an import scenario called Import to Cranston Operation that imports data into a planning model called CranstonOperation and is associated with an extract area called Inbound to Supply Chain Business Modeler :

```
bm::scenario add import "Import to Cranston Operation" CranstonOperation
"Inbound to Supply Chain Business Modeler"
```

The following example creates an import scenario called Update Cranston Operation that updates data in a planning model called CranstonOperation and uses data from an extract area called Inbound to Supply Chain Business Modeler :

```
bm::scenario add import "Update Cranston Operation" CranstonOperation
"Inbound to Supply Chain Business Modeler" -update
```

The following example creates an import scenario called Import to Base that is associated with the Base model and an extract area called Inbound to Supply Chain Business Modeler, and runs a prescript called Extract Supply Chain Management Data:

```
bm::scenario add import "Import to Base" Base "Inbound to Supply Chain
Business Modeler" -preScript "Extract Supply Chain Management Data"
```

The following example creates an export scenario called Export to Cranston PS that is associated with a model called Cranston Operation and an extract area called Outbound from Supply Chain Business Modeler, and runs a postscript called CranstonPSSolve:

```
bm::scenario add export "Export to Cranston PS" "Cranston Operation"
"Outbound from Supply Chain Business Modeler" -postScript CranstonPSSolve
```

The following example creates an export scenario called Export to Gunnison PDP that exports data in Supply Chain Planning XML 2.0 format from a model called Gunnison Tactical to an extract area called Outbound from Supply Chain Business Modeler :

```
bm::scenario add export "Export to Gunnison PDP" "Gunnison Tactical"
"Outbound from Supply Chain Business Modeler" -interfaceVersion 2.0
```

Note. Do not import incremental data packages using full import scenarios. If you import incremental data packages using full import scenarios, Supply Chain Business Modeler assigns default values in the model for data that is missing from the incremental data packages.

See Also

"Business Modeler Commands," importScenario add
"Business Modeler Commands," exportScenario add

scenario addPackage

Syntax

```
bm::scenario addPackage scenarioName package dataFolder
```

Description

Use the scenario addPackage command to specify which data packages to import in an import scenario or export in an export scenario. Before you specify the data packages for a scenario, the scenario must already exist in Supply Chain Business Modeler.

Note. The scenario addPackage command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario addPackage command, you can use the importScenario addPackage command or the exportScenario addPackage command.

Parameters

Parameter	Description
scenarioName	Name of the import or export scenario to which you want to add a data package. The scenario must already exist. If the scenario name includes blanks, it must be enclosed in quotation marks.
dataFolder	Name of the data folder where the data package is saved in the scenario. If the data folder name includes blanks, it must be enclosed in quotation marks.
package	Name of the data package that you want to add to the scenario. You can include only one package name in each scenario addPackage command. Values depend on the level of the model that is specified in the scenario, as shown in the following table:

This table shows values for *package:*

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
CurrencyForecast	X	X	X	X	
Demand				X	
DemandPointHistory					X
Fixed Production				X	
PurchaseOrders	X	X	X	X	

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
SafetyParameters	X				X
SalesOrderHistory	X				
SalesOrders	X	X	X		
Sales					X
TimeSeries	X				X
TransferOrders	X	X	X	X	
WorkOrders	X	X	X		
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X
MasterProductionPlan	X				
NetDeploymentRequirements		X	X		
NetProductionRequirements		X	X		
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X		
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	
WorkOrderSchedule	X	X			

Example

The following example adds the Manufacturing data package to a scenario called Import to Base in a data folder called From Supply Chain Management:

```
bm::scenario addPackage "Import to Base" Manufacturing "From Supply Chain Management"
```

See Also
"Business Modeler Commands," importScenario addPackage
"Business Modeler Commands," exportScenario addPackage

scenario compressed

Syntax
```
bm::scenario compressed scenarioName
```

Description

Use the scenario compressed command to determine whether an export scenario exports data in compressed or uncompressed format.

Note. The scenario compressed command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario compressed command, you can use the exportScenario compressed command.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to determine whether data is exported in compressed format. If the export scenario name includes blanks, it must be enclosed in quotation marks

Returns

Returns true if data is exported in compressed format. Returns false if data is exported in uncompressed format.

Example

The following command indicates whether data exported using an export scenario called Demand Planning Data exports data in compressed or uncompressed format:

```
bm::scenario compressed "Demand Planning Data"
```

See Also
"Business Modeler Commands," exportScenario add

scenario dataFolder

Syntax
```
bm::scenario dataFolder scenarioName packageName
```

Description

Use the scenario dataFolder command to obtain the name of the data folder associated with a data package in an import or export scenario.

Note. The scenario dataFolder command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario dataFolder command, you can use the importScenario dataFolder command or the exportScenario dataFolder command.

Parameters

Parameter	Description
scenarioName	Name of the import or export scenario for which you want to obtain the data folder name. If the scenario name includes blanks, it must be enclosed in quotation marks.
packageName	Name of the data package for which you want to obtain the data folder name. Values for *packageName* depend on the level of the model that is specified in the scenario, as shown in the following table:

This table shows values for *package:*

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
Enterprise Data					
Base	X	X	X	X	X
Calendar	X	X	X	X	
Customer	X	X	X	X	X
Distribution	X	X	X	X	
Manufacturing	X	X	X	X	
Supplier	X	X	X	X	
Transactional Data					
BeginningInventory	X	X	X	X	
Configuration	X	X	X	X	X
CurrencyForecast	X	X	X	X	
Demand				X	
DemandPointHistory					X
Fixed Production				X	
PurchaseOrders	X	X	X	X	
SafetyParameters	X				X
SalesOrderHistory	X				
SalesOrders	X	X	X		
Sales					X
TimeSeries	X				X

Values	Base Model Level	Operation Model Level	Tactical Model Level	Strategic Model Level	Demand Model Level
TransferOrders	X	X	X	X	
WorkOrders	X	X	X		
Optimized Plans					
DeploymentPlan	X				
DetailedProductionPlan	X				
EnterpriseForecast	X	X	X	X	X
InventoryBuildTargets	X	X	X	X	
InventorySafetyTargets	X	X	X	X	X
MasterProductionPlan	X				
NetDeploymentRequirements		X	X		
NetProductionRequirements		X	X		
ProductionSchedule		X			
PurchaseOrder Recommendations		X	X		
PurchasePlan	X				
SourcingRecommendations	X	X	X	X	
WorkOrderSchedule	X	X			

Returns

Returns the data folder associated with a data package in a scenario. For example, the command could return:

```
To Supply Chain Management
```

Example

The following example returns the data folder name for the Base data package in a scenario named Export to Supply Chain Management:

```
bm::scenario dataFolder "Export to Supply Chain Management" Base
```

See Also

“Business Modeler Commands,” importScenario dataFolder
“Business Modeler Commands,” exportScenario dataFolder

scenario delete

Syntax

```
bm::scenario delete scenarioName
```

Description

Use the scenario delete command to delete an import or export scenario.

Note. The scenario delete command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario delete command, you can use the importScenario delete command or the exportScenario delete command.

Parameters

Parameter	Description
scenarioName	Name of the scenario that you want to delete. If the scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example deletes a scenario called Export to Gunnison PDP:

```
bm::scenario delete "Export to Gunnison PDP"
```

See Also

"Business Modeler Commands," importScenario delete
"Business Modeler Commands," exportScenario delete

scenario execute

Syntax

```
bm::scenario execute scenarioName [-snapshotDate snapshotDate]
```

Description

Use the scenario execute command to run an import or export scenario.

Note. The scenario execute command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario execute command, you can use the importScenario execute command or the exportScenario execute command.

Parameters

Parameter	Description
scenarioName	Name of the scenario that you want to run. If the scenario name includes blanks, it must be enclosed in quotation marks.
–snapshotDate *snapshotDate*	(Optional) If you are running an import scenario, you must include the –snapshotDate option, where *snapshotDate* is the snapshot date in *yyyy–mm–dd*T*HH:MM:SS* format. The snapshot date is the date and time as of which all of the transactional data that you are importing is consistent and provides a synchronized view of the enterprise. For example, if a work order is partially completed as of the snapshot date, the appropriate inventory adjustments must also be made to reflect the work order consumption. If you are running an export scenario, do not include the –snapshotDate option.

Example

The following example runs an import scenario called Import to Cranston Operation with a snapshot date of January 1, 2004 at midnight:

```
bm::scenario execute "Import to Cranston Operation" -snapshotDate 2004-01-01T00:00:00
```

The following example runs an export scenario called Export to Cranston PS:

```
bm::scenario execute "Export to Cranston PS"
```

See Also

"Business Modeler Commands," importScenario execute
"Business Modeler Commands," exportScenario execute

scenario extractArea

Syntax

`bm::scenario extractArea` *`scenarioName`*

Description

Use the scenario extractArea command to obtain the name of the extract area associated with an import or export scenario.

Note. The scenario extractArea command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario extractArea command, you can use the importScenario extractArea command or the exportScenario extractArea command.

Parameters

Parameter	Description
scenarioName	Name of the scenario for which you want to obtain the extract area name. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

The command returns the name of the extract area that is associated with the scenario. For example, the command could return the following:

```
Inbound to Supply Chain Business Modeler
```

Example

The following example returns the extract area name for an export scenario called Import to Cranston Operation, use the following command:

```
bm::scenario extractArea "Import to Cranston Operation"
```

See Also

"Business Modeler Commands," importScenario extractArea
"Business Modeler Commands," exportScenario extractArea

scenario incremental

Syntax

`bm::scenario incremental` *`scenarioName`*

Description

Use the scenario incremental command to determine whether an import scenario is a full or an incremental import scenario.

Note. The scenario incremental command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario incremental command, you can use the importScenario incremental command.

Parameters

Parameter	Description
scenarioName	Name of the import scenario. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns true if the import scenario is an incremental import scenario. Returns false if the import scenario is a full import scenario.

Example

The following example indicates whether an import scenario called Import to Cranston Operation is a full or an incremental import scenario:

```
bm::scenario incremental "Import to Cranston Operation"
```

See Also

"Business Modeler Commands," importScenario incremental

scenario interfaceVersion

Syntax

`bm::scenario interfaceVersion` *`scenarioName`* `[-set`*`version`*`]`

Description

Use the scenario interfaceVersion command to set or obtain the version of Supply Chain Planning XML data that is exported when you run an export scenario.

Note. The scenario interfaceVersion command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario interfaceVersion command, you can use the exportScenario interfaceVersion command.

Parameters

Parameter	Description
scenarioName	Name of the export scenario for which you want to set or obtain the interface version. If the scenario name includes blanks, it must be enclosed in quotation marks.
–set *version*	(Optional) To set the version of Supply Chain Planning XML that is exported when you run the export scenario, you must include the –set option, where *version* is the version of the Supply Chain Planning XML format that you want to export. Values for *version* are 3.3, 3.2, 3.1, 3.0 and 2.0. To determine the version of the data that is exported in an export scenario, do not include the –set option.

Example

The following example sets the XML format to Supply Chain Planning XML 3.0 in an export scenario called ExportBase:

```
bm::scenario interfaceVersion ExportBase -set 3.0
```

The following example returns the version of Supply Chain Planning XML that is exported from a scenario called ExportBase:

```
bm::scenario interfaceVersion ExportBase
```

Note. You do not have to specify the interface version of Supply Chain Planning XML documents that are imported in an import scenario. Supply Chain Business Modeler reads the versions of Supply Chain Planning XML documents when it imports the files.

See Also

"Business Modeler Commands," exportScenario interfaceVersion

scenario model

Syntax

`bm::scenario model` *`scenarioName`*

Description

Use the scenario model command to obtain the name of the model associated with an import or export scenario.

Note. The scenario model command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario model command, you can use the importScenario model command or the exportScenario model command.

Parameters

Parameter	Description
scenarioName	Name of the scenario for which you want to obtain the model name. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the name of the model associated with an import or export scenario. For example, the command could return:

```
Gunnison Tactical
```

Example

This example returns the model name for an export scenario named Export to Gunnison PDP:

```
bm::scenario model "Export to Gunnison PDP"
```

See Also

"Business Modeler Commands," importScenario model
"Business Modeler Commands," exportScenario model

scenario names

Syntax

`bm::scenario names [-type scenarioType] [-modelName modelName]`

Description

Use the scenario names command to obtain a list of scenarios that have been created in Supply Chain Business Modeler.

Note. The scenario names command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario names command, you can use the importScenario names command or the exportScenario names command.

Parameters

Parameter	Description
–type *scenarioType*	(Optional) To obtain a list of only one type of scenario, you can include the –type option, in which *scenarioType* is the type of scenario for which you want a list. Scenario types are import and export.
–modelName *modelName*	(Optional) To obtain a list of scenarios that import data into or export data from a particular model, you can include the –modelName option, in which *modelName* is the name of the model for which you want a list of related scenarios.

Returns

Returns a list of scenarios that have been created in Supply Chain Business Modeler. Scenario names that include blanks are enclosed in curly brackets in the list. For example, the command could return the following:

```
{Import to Base} ExportFromBase
```

Example

The following example returns a list of all scenarios that export data from the Base model:

```
bm::scenario names -type export -modelName Base
```

The following example returns a list of all import scenarios:

```
bm::scenario names -type import
```

See Also

"Business Modeler Commands," importScenario names
"Business Modeler Commands," exportScenario names

scenario packages

Syntax

`bm::scenario packages` *scenarioName*

Description

Use the scenario packages command to obtain the names of the data packages specified in an import or export scenario.

Note. The scenario packages command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario packages command, you can use the importScenario packages command or the exportScenario packages command.

Parameters

Parameter	Description
scenarioName	Name of the scenario for which you want to obtain the data package names. If the scenario name includes blanks, it must be enclosed in quotation marks.

Returns

Returns the names of data packages specified in an import or export scenario. The command returns a list similar to the following:

```
Base Manufacturing Distribution Customer Calendar Supplier
```

Example

The following example returns the data package names for a scenario named Import to Base:

```
bm::scenario packages "Import to Base"
```

See Also

"Business Modeler Commands," importScenario packages
"Business Modeler Commands," exportScenario packages

scenario rename

Syntax

```
bm::scenario rename oldScenarioName newScenarioName
```

Description

Use the scenario rename command to rename an import or export scenario.

Note. The scenario rename command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario rename command, you can use the importScenario rename command or the exportScenario rename command.

Parameters

Parameter	Description
oldScenarioName	Name of the scenario that you want to rename. If the old scenario name includes blanks, it must be enclosed in quotation marks.
newScenarioName	New name for the scenario. If the old scenario name includes blanks, it must be enclosed in quotation marks.

Example

The following example renames a scenario called Import 1 to ImportToBase:

```
bm::scenario rename "Import 1" ImportToBase
```

See Also

"Business Modeler Commands," importScenario rename
"Business Modeler Commands," exportScenario rename

scenario script

Syntax

```
bm::scenario script scenarioName [-pre] [-post]
```

Description

Use the scenario script command to obtain the name of any Tcl script that is specified as a prescript or a postscript in an import or export scenario.

Note. The scenario script command is deprecated and will be obsolete in future versions of Supply Chain Business Modeler. Instead of the scenario script command, you can use the importScenario script command or the exportScenario script command.

Parameters

Parameter	Description
scenarioName	Name of the scenario for which you want to obtain the Tcl script names. If the old scenario name includes blanks, it must be enclosed in quotation marks.
–pre	(Optional) To obtain the name of the pre-script only, include the –pre option.
–post	(Optional) To obtain the name of the post-script only, include the –post option.

To obtain the names of both the pre-script and the post-script, you do not have to include either option.

Returns

Returns the names of pre-scripts or post-scripts that are associated with an import or export scenario. If no pre-script or post-script is associated with the scenario, Supply Chain Business Modeler returns empty curly brackets. For example, the command could return the following:

```
pre.tcl post.tcl
```

Example

The following example returns the names of the Tcl scripts that are associated with a scenario called Export from Base:

```
bm::scenario script "Export from Base"
```

See Also

"Business Modeler Commands," importScenario script
"Business Modeler Commands," exportScenario script

version

Syntax

```
bm::version
```

Description

The version command returns the version of Business Modeler commands that you are running.

See Also

"Business Modeler Commands"
"Using the Business Modeler Shell"

workbench refresh

Syntax

```
bm::workbench refresh
```

Description

Use the workbench refresh command to populate the changeover and group editors with branch, resource, operation, and calendar data from the Base model.

When you refresh the changeover and group editors with data from the Base model, any branch codes, resource codes, operation codes, and calendar names in the changeover and group editors are cleared and replaced with values from the Base model. Changeover patterns and groups that were already created in the editors remain in the editors. Changeover patterns that include branches, resources, operations, or combinations of these variables that are no longer populated in the changeover editor remain in the editor, but are marked as invalid.

When you refresh the changeover editor with data from the Base model, only branches that are plants or plant/distribution centers are populated in the changeover editor. Branch codes for distribution centers are not populated in the changeover editor.

See Also
"Creating Changeover Patterns and Groups"

xml transform

Syntax

bm::xml transform [–parameters {*parameter1 value1... parametern valuen*}]
inputXml inputXsl outputXml

Description

Use the xml transform command to transform an XML document into a new XML format specified by an Extensible Stylesheet Language (XSL) stylesheet.

Parameters

Parameter	Description
inputXml	Name of the XML file that you want to transform into the new format.
inputXsl	Name of the XSL stylesheet that specifies how to transform the XML file.
outputXml	Name of the file where you want to save the XML in the new format. To write the XML in the new format to the screen, enter `stdout` as the *outputXml* value.
–parameters *{parameter1 value1... parametern valuen}*	(Optional) If the XSL stylesheet requires parameters, you must include the –parameters option. The parameter names and values must be enclosed in curly brackets. Parameter values that include blanks must be enclosed in single quotation marks.

Example

The following example transforms an XML file called EnterpriseOneData.xml with a stylesheet called filter.xsl and saves the file as Base.xml:

```
bm::xml transform EnterpriseOneData.xml filter.xsl Base.xml
```

The following example transforms an XML file called EnterpriseOneData.xml into the format specified by a stylesheet called filter.xsl and writes the XML to the screen:

```
bm::xml transform EnterpriseOneData.xml filter.xsl stdout
```

The following example transforms an XML file called InventorySafetyTargets.xml using a stylesheet called BranchFilter.xsl, assigns the BranchCode parameter a value of Cranston, and saves the file as CranstonInventorySafetyTargets.xml:

```
bm::xml transform -parameters {BranchCode Cranston}
```



```
InventorySafetyTargets.xml BranchFilter.xsl
CranstonInventorySafetyTargets.xml
```

See Also

"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler"

xml validate

Syntax

bm::xml validate *inputXml* [–xsd *inputXsd*]

Description

Use the xml validate command to determine whether an XML file conforms to the document structure specified by a Document Type Definition (DTD) or XML Schema Definition (XSD). Supply Chain Business Modeler is shipped with XSDs for documents in Supply Chain Planning XML 3.3, 3.2, 3.1, 3.0, and 2.0 format for full and incremental import scenarios for each model level.

Parameters

Parameter	Description
inputXml	XML file that you want to validate.
–xsd *inputXsd*	(Optional) To validate an XML file using an XSD, include the –xsd parameter, where *inputXsd* is the name of the XSD that you want to use for validating the XML file. To validate the XML file using a DTD or XSD specified in the *inputXml* file, do not include the –xsd parameter.

Returns

If the XML file is valid, the command does not return any information. If the XML file is not valid, the command returns a message for each problem to the screen and to the Supply Chain Business Modeler error log. Each message describes the problem and indicates where the problem occurred.

Example

The following example validates an XML file called CranstonInventorySafetyTargets.xml using an XSD called InventorySafetyTargets.xsd:

```
bm::xml validate CranstonInventorySafetyTargets.xml
-xsd InventorySafetyTargets.xsd
```

See Also

"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler," XML Schema Definitions

xml transform

Syntax

bm::xml transform [–parameters {*parameter1 value1... parametern valuen*}]
inputXml inputXsl outputXml

Description

Use the xml transform command to transform an XML document into a new XML format specified by an Extensible Stylesheet Language (XSL) stylesheet.

Parameters

Parameter	Description
inputXml	Name of the XML file that you want to transform into the new format.
inputXsl	Name of the XSL stylesheet that specifies how to transform the XML file.
outputXml	Name of the file where you want to save the XML in the new format. To write the XML in the new format to the screen, enter `stdout` as the *outputXml* value.
–parameters *{parameter1 value1... parametern valuen}*	(Optional) If the XSL stylesheet requires parameters, you must include the –parameters option. The parameter names and values must be enclosed in curly brackets. Parameter values that include blanks must be enclosed in single quotation marks.

Example

The following example transforms an XML file called EnterpriseOneData.xml with a stylesheet called filter.xsl and saves the file as Base.xml:

```
bm::xml transform EnterpriseOneData.xml filter.xsl Base.xml
```

The following example transforms an XML file called EnterpriseOneData.xml into the format specified by a stylesheet called filter.xsl and writes the XML to the screen:

```
bm::xml transform EnterpriseOneData.xml filter.xsl stdout
```

The following example transforms an XML file called InventorySafetyTargets.xml using a stylesheet called BranchFilter.xsl, assigns the BranchCode parameter a value of Cranston, and saves the file as CranstonInventorySafetyTargets.xml:

```
bm::xml transform -parameters {BranchCode Cranston}
InventorySafetyTargets.xml BranchFilter.xsl
CranstonInventorySafetyTargets.xml
```

See Also

"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler"

xml validate

Syntax

bm::xml validate *inputXml* [–xsd *inputXsd*]

Description

Use the xml validate command to determine whether an XML file conforms to the document structure specified by a Document Type Definition (DTD) or XML Schema Definition (XSD). Supply Chain Business Modeler is shipped with XSDs for documents in Supply Chain Planning XML 3.3, 3.2, 3.1, 3.0, and 2.0 format for full and incremental import scenarios for each model level.

Parameters

Parameter	Description
inputXml	XML file that you want to validate.
–xsd *inputXsd*	(Optional) To validate an XML file using an XSD, include the –xsd parameter, where *inputXsd* is the name of the XSD that you want to use for validating the XML file. To validate the XML file using a DTD or XSD specified in the *inputXml* file, do not include the –xsd parameter.

Returns

If the XML file is valid, the command does not return any information. If the XML file is not valid, the command returns a message for each problem to the screen and to the Supply Chain Business Modeler error log. Each message describes the problem and indicates where the problem occurred.

Example

The following example validates an XML file called CranstonInventorySafetyTargets.xml using an XSD called InventorySafetyTargets.xsd:

```
bm::xml validate CranstonInventorySafetyTargets.xml
-xsd InventorySafetyTargets.xsd
```

See Also

"Understanding Data for Importing into and Exporting from Supply Chain Business Modeler," XML Schema Definitions

CHAPTER 18

Glossary

This chapter discusses terms used in Supply Chain Business Modeler.

Understanding Terms Used in Supply Chain Business Modeler

This table lists terms used in Supply Chain Business Modeler:

alignment date The start date and time of one time period in a Tactical, Strategic, or Demand model.

Business Modeler commands Tool Command Language (Tcl) commands for running Supply Chain Business Modeler functions commands from the Business Modeler Shell. You can enter Business Modeler commands interactively in the Business Modeler Shell, or create and run scripts of Business Modeler and other Tcl commands.

component A timeline that can contain a recurring event or one or more one-time events. By adding components to calendars, you can model the availability of resources in a supply chain.

connected models Models that data can flow between in Supply Chain Business Modeler. You can populate a model with data from a connected model by refreshing data in or publishing data from one of the models. Lines appear between connected models in the Supply Chain Business Modeler Models navigator.

connector Program for converting data from EnterpriseOne Supply Chain Planning systems into the XML format required for importing into Supply Chain Business Modeler and for converting data from Supply Chain Business Modeler for importing into EnterpriseOne Supply Chain Planning systems.

data folder An extract area subdirectory that is managed from within Supply Chain Business Modeler and stores XML data files that you can import into Supply Chain Business Modeler or have been exported from Supply Chain Business Modeler.

data package A group of related data that must be stored and transferred together to ensure the integrity and reliability of the data. For example, the Customer data package includes related information about customer locations, product preferences, and sourcing.

Demand model A subset of predefined data objects in the Supply Chain Business Modeler object database that represents a supply chain at the level of granularity that is required for creating demand forecasts using EnterpriseOne Supply Chain Planning systems such as Demand Forecasting and Demand Consensus.

detail area An area of the Supply Chain Business Modeler window that shows information about an item that is selected in the Navigator. For example, if you select an import scenario in the Navigator, the Detail area shows information about that scenario.

Enterprise data Information about the structure of a supply chain (e.g., supplier, customer, and distribution network information). This data does not change frequently, and is traditionally called static data.

Event A period of time when a resource is available, unavailable, or costs extra to use. A component in Supply Chain Business Modeler can include one or more events.

event pattern A recurring period of time when a resource is available, unavailable, or costs extra to use. A component in Supply Chain Business Modeler can include one event pattern.

export horizon The length of time following the snapshot date for which data is exported from a model. All data in a model that is dated before the snapshot date is also be exported from the model.

export scenario A specification of:

- Data to export from Supply Chain Business Modeler
- The Supply Chain Business Modeler model to export the data from
- An extract area data folder for saving the exported data
- (optional) Tcl scripts to run before or after the data is exported

You can run an export scenario to export the specified data to the specified data folder, and run any Tcl scripts.

extract area A directory on the operating system that is managed from within Supply Chain Business Modeler. This directory is used to manage Supply Chain Planning XML data files and Tcl scripts.

import scenario A specification of:

- Data to import into Supply Chain Business Modeler
- The extract area data folder where the data is saved
- The Supply Chain Business Modeler model to populate with the imported data
- (optional) Tcl scripts to run before or after the data is imported

You can run an import scenario to populate the specified model with the specified data, and run any Tcl scripts.

logs area An area of the Supply Chain Business Modeler window that displays errors generated during the current Supply Chain Business Modeler session.

model A subset of predefined data objects in the Supply Chain Business Modeler object database that represents a supply chain at the level of granularity that is required for a specific supply chain management or planning purpose. You can populate Supply Chain Business Modeler models with data from Supply Chain Management and Supply Chain Planning systems.

model generation scenario Model generation scenarios provide a way to specify how Supply Chain Planning applications use data exported from Supply Chain Business Modeler. You can create model generation scenarios that specify options for displaying and using data in a particular Supply Chain Planning product such as Demand Management, Strategic Network Optimization, Production and Distribution Planning, and Production Scheduling. You can then run the model generation scenario to create a model generation rules (.mgr) file. The application connector can call the .mgr file that specifies how to use the data in the destination application.

model hierarchy The organization of Supply Chain Business Modeler models in order of their level of data granularity. Models toward the bottom of the model hierarchy (e.g., Base and Operation models) represent a supply chain at a fine level of granularity or detail. Models toward the top of the model hierarchy (e.g., Demand and Strategic models) represent a supply chain with less detail and with data summarized in time periods.

navigator An area of the Supply Chain Business Modeler window that displays information related to the shortcut or function that you choose. For example, if you click the Extract Areas shortcut, the Navigator displays the names of extract areas and data folders in a tree structure. When you are working with models, the Navigator shows the Supply Chain Business Modeler model hierarchy.

Operation model A subset of predefined data objects in the Supply Chain Business Modeler object database that represents a supply chain at the level of granularity that is required for creating detailed production schedules using Supply Chain Planning systems such as Production Scheduling and Production Scheduling - Process.

Optimized plan data Supply chain plans and forecasts that are created by EnterpriseOne Supply Chain Planning systems such as Production Scheduling, Production & Distribution Planning, Strategic Network Optimization, and Demand Management.

package A group of related data that must be stored and transferred together to ensure the integrity and reliability of the data. For example, the Customer data package includes related information about customer locations, product preferences, and sourcing.

planning level A position in the Supply Chain Business Modeler model hierarchy where data objects represent a supply chain at a particular level of granularity. There are five planning levels in the Supply Chain Business Modeler model hierarchy: Base, Operation, Tactical, Strategic, and Demand. Models at each level represent a supply chain at the level of granularity that is required for a specific supply chain management or planning purpose and application.

planning model A subset of predefined data objects in the Supply Chain Business Modeler object database that includes data for one or more specified branches in a supply chain at the level of granularity required for a particular planning purpose. Planning models enable you to export data for a particular branch or branches in a supply chain for use in EnterpriseOne Supply Chain Planning systems.

publish A method of populating one Supply Chain Business Modeler model with optimized plans from a connected model. You can publish plans from a

reference model to the connected reference model that is lower in the model hierarchy. You can publish plans from a planning model to the reference model at the same planning level in the model hierarchy.

Also a Connector command for preparing data exported from Supply Chain Planning systems for importing into Supply Chain Business Modeler.

reference model A subset of predefined data objects in the Supply Chain Business Modeler object database that includes data for all branches in a supply chain at the level of granularity required for a particular planning purpose. Only one reference model can appear at each level of the model hierarchy.

refresh A method of populating one Supply Chain Business Modeler model with enterprise, transactional, and optimized plan data from a connected model. You can refresh a reference model with data from the connected reference model that is lower in the model hierarchy. You can refresh a planning model with data from the reference model at the same planning level in the model hierarchy.

Also a Connector command for preparing data exported from Supply Chain Business Modeler for use in an EnterpriseOne Supply Chain Planning system.

reusable component A timeline that you can add to one or more calendars for modeling resource availability in a supply chain.

Supply Chain Planning XML format The format used for importing data into Supply Chain Business Modeler or exporting data from Supply Chain Business Modeler. Data in Supply Chain Planning XML format is divided into separate XML documents or packages. Each package includes related data. Supply Chain Business Modeler 8.12.1 supports five versions of Supply Chain Planning XML format: 3.3, 3.2, 3.1, 3.0, and 2.0.

schema viewer A series of HTML pages that provide information about all Supply Chain Business Modeler model objects and attributes, including their data types and default values.

scripts folder An extract area subdirectory that is managed from within Supply Chain Business Modeler and stores Tool Command Language (Tcl) scripts for that can run before or after you import data into or export data from Supply Chain Business Modeler.

shortcut An icon in the Supply Chain Business Modeler window that enables you to quickly view Supply Chain Business Modeler items and access related functions. For example, if you click the Extract Areas button in the Shortcuts bar, the Navigator displays Supply Chain Business Modeler extract areas in a tree structure and you can create and manage the extract areas.

snapshot date The date and time as of which imported transactional data is consistent and provides a synchronized view of the supply chain. For example, if a work order is partially completed as of the snapshot date, appropriate inventory adjustments must be made in the data to reflect the consumption on the work order by the snapshot date.

Strategic model A subset of predefined data objects in the Supply Chain Business Modeler object database that represents a supply chain at the level of granularity that is required for creating sourcing guidelines, capacity requirements, and inventory build targets using EnterpriseOne Supply Chain Planning systems such as Strategic Network Optimization.

Tactical model A subset of predefined data objects in the Supply Chain Business Modeler object database that represents a supply chain at the level of granularity that is required for creating net production requirements, purchase order recommendations, and deployment plans using EnterpriseOne Supply Chain Planning systems such as Production and Distribution Planning and Tactical Network Optimization.

Transactional data Information about supply and demand in a supply chain, such as sales order, transfer order, and work order information. This data changes frequently, and is traditionally called dynamic data.

Glossary of JD Edwards EnterpriseOne Terms

activity	A scheduling entity in JD Edwards EnterpriseOne tools that represents a designated amount of time on a calendar.
activity rule	The criteria by which an object progresses from one given point to the next in a flow.
add mode	A condition of a form that enables users to input data.
Advanced Planning Agent (APAg)	A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of rational databases, flat file format, and other data or message encoding, such as XML.
application server	A server in a local area network that contains applications shared by network clients.
as if processing	A process that enables you to view currency amounts as if they were entered in a currency different from the domestic and foreign currency of the transaction.
alternate currency	A currency that is different from the domestic currency (when dealing with a domestic-only transaction) or the domestic and foreign currency of a transaction. In JD Edwards EnterpriseOne Financial Management, alternate currency processing enables you to enter receipts and payments in a currency other than the one in which they were issued.
as of processing	A process that is run as of a specific point in time to summarize transactions up to that date. For example, you can run various JD Edwards EnterpriseOne reports as of a specific date to determine balances and amounts of accounts, units, and so on as of that date.
back-to-back process	A process in JD Edwards EnterpriseOne Supply Management that contains the same keys that are used in another process.
batch processing	A process of transferring records from a third-party system to JD Edwards EnterpriseOne. In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.
batch server	A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.
batch-of-one immediate	A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks. See also direct connect and store-and-forward.
business function	A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules,

	and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.
business function event rule	See named event rule (NER).
business view	A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.
central objects merge	A process that blends a customer's modifications to the objects in a current release with objects in a new release.
central server	A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.
charts	Tables of information in JD Edwards EnterpriseOne that appear on forms in the software.
connector	Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.
contra/clearing account	A general ledger account in JD Edwards EnterpriseOne Financial Management that is used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations in JD Edwards EnterpriseOne Financial Management.
Control Table Workbench	An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.
control tables merge	A process that blends a customer's modifications to the control tables with the data that accompanies a new release.
cost assignment	The process in JD Edwards EnterpriseOne Advanced Cost Accounting of tracing or allocating resources to activities or cost objects.
cost component	In JD Edwards EnterpriseOne Manufacturing, an element of an item's cost (for example, material, labor, or overhead).
cross segment edit	A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced.
currency restatement	The process of converting amounts from one currency into another currency, generally for reporting purposes. You can use the currency restatement process, for example, when many currencies must be restated into a single currency for consolidated reporting.
database server	A server in a local area network that maintains a database and performs searches for client computers.
Data Source Workbench	An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

date pattern A calendar that represents the beginning date for the fiscal year and the ending date for each period in that year in standard and 52-period accounting.

denominated-in currency The company currency in which financial reports are based.

deployment server A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

detail information Information that relates to individual lines in JD Edwards EnterpriseOne transactions (for example, voucher pay items and sales order detail lines).

direct connect A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one immediate and store-and-forward.

Do Not Translate (DNT) A type of data source that must exist on the iSeries because of BLOB restrictions.

dual pricing The process of providing prices for goods and services in two currencies.

edit code A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

edit mode A condition of a form that enables users to change data.

edit rule A method used for formatting and validating user entries against a predefined rule or set of rules.

Electronic Data Interchange (EDI) An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

embedded event rule An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.

Employee Work Center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

enterprise server A server that contains the database and the logic for JD Edwards EnterpriseOne.

EnterpriseOne object A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

EnterpriseOne process A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

Environment Workbench An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

escalation monitor A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

event rule A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

facility An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. A facility is sometimes referred to as a "business unit."

fast path A command prompt that enables the user to move quickly among menus and applications by using specific commands.

file server A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

final mode The report processing mode of a processing mode of a program that updates or creates data records.

FTP server A server that responds to requests for files via file transfer protocol.

header information Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows.

interface table See Z table.

integration server A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

integrity test A process used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

interoperability model A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

in-your-face-error In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

IServer service This internet server service resides on the web server and is used to speed up delivery of the Java class files from the database to the client.

jargon An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

Java application server A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

JDBNET A database driver that enables heterogeneous servers to access each other's data.

JDEBASE Database Middleware A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

JDECallObject An API used by business functions to invoke other business functions.

jde.ini A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

JDEIPC Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

jde.log The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

JDENET A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

Location Workbench An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

logic server A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

MailMerge Workbench An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

master business function (MBF) An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

master table See published table.

matching document A document associated with an original document to complete or change a transaction. For example, in JD Edwards EnterpriseOne Financial Management, a receipt is the matching document of an invoice, and a payment is the matching document of a voucher.

media storage object Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

message center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

messaging adapter An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

messaging server A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

named event rule (NER) Encapsulated, reusable business logic created using event rules, rather that C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

nota fiscal In Brazil, a legal document that must accompany all commercial transactions for tax purposes and that must contain information required by tax regulations.

nota fiscal factura In Brazil, a nota fiscal with invoice information.

See also *nota fiscal.*

Object Configuration Manager (OCM) In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

Object Librarian A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as production and development) and enables objects to be easily moved from one environment to another.

Object Librarian merge A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

Open Data Access (ODA) An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

Output Stream Access (OSA) An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

package JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

package build A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: "Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions." The process of creating a package build is often referred to, as it is in this example, simply as "a package build."

package location The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

Package Workbench An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

planning family A means of grouping end items whose similarity of design and manufacture facilitates being planned in aggregate.

preference profile The ability to define default values for specified fields for a user-defined hierarchy of items, item groups, customers, and customer groups.

print server The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

pristine environment A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

processing option A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

production environment A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

production-grade file server A file server that has been quality assurance tested and commercialized and that is usually provided in conjunction with user support services.

program temporary fix (PTF) A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

project In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

promotion path The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, *11* equals new project pending review, *21* equals programming, *26* equals QA test/review, *28* equals QA test/review complete, *38* equals in production, *01* equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

proxy server A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

published table Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

publisher The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

pull replication One of the JD Edwards EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers using JD Edwards EnterpriseOne data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the F98DRPCN table.

QBE An abbreviation for query by example. In JD Edwards EnterpriseOne, the QBE line is the top line on a detail area that is used for filtering data.

real-time event A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and to provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when certain transactions occur.

refresh A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level, such as B73.2 or B73.2.1.

replication server A server that is responsible for replicating central objects to client machines.

quote order In JD Edwards Procurement and Subcontract Management, a request from a supplier for item and price information from which you can create a purchase order.

	In JD Edwards Sales Order Management, item and price information for a customer who has not yet committed to a sales order.
selection	Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.
Server Workbench	An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number data source. It also updates the Server Plan detail record to reflect completion.
spot rate	An exchange rate entered at the transaction level. This rate overrides the exchange rate that is set up between two currencies.
Specification merge	A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.
specification	A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.
Specification Table Merge Workbench	An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.
store-and-forward	The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.
subscriber table	Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.
supplemental data	Any type of information that is not maintained in a master file. Supplemental data is usually additional information about employees, applicants, requisitions, and jobs (such as an employee's job skills, degrees, or foreign languages spoken). You can track virtually any type of information that your organization needs. For example, in addition to the data in the standard master tables (the Address Book Master, Customer Master, and Supplier Master tables), you can maintain other kinds of data in separate, generic databases. These generic databases enable a standard approach to entering and maintaining supplemental data across JD Edwards EnterpriseOne systems.
table access management (TAM)	The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.
Table Conversion Workbench	An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.
table conversion	An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.
table event rules	Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.
terminal server	A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

three-tier processing The task of entering, reviewing and approving, and posting batches of transactions in JD Edwards EnterpriseOne.

three-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records to create vouchers.

transaction processing (TP) monitor A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

transaction set An electronic business transaction (electronic data interchange standard document) made up of segments.

trigger One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

triggering event A specific workflow event that requires special action or has defined consequences or resulting actions.

two-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information.

User Overrides merge Adds new user override records into a customer's user override table.

variance In JD Edwards Capital Asset Management, the difference between revenue generated by a piece of equipment and costs incurred by the equipment.

In JD Edwards EnterpriseOne Project Costing and JD Edwards EnterpriseOne Manufacturing, the difference between two methods of costing the same item (for example, the difference between the frozen standard cost and the current cost is an engineering variance). Frozen standard costs come from the Cost Components table, and the current costs are calculated using the current bill of material, routing, and overhead rates.

Version List merge The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

visual assist Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

vocabulary override An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

wchar_t An internal type of a wide character. It is used for writing portable programs for international markets.

web application server A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

web server A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

Windows terminal server A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows

terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

workbench A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

work day calendar In JD Edwards EnterpriseOne Manufacturing, a calendar that is used in planning functions that consecutively lists only working days so that component and work order scheduling can be done based on the actual number of work days available. A work day calendar is sometimes referred to as planning calendar, manufacturing calendar, or shop floor calendar.

workflow The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

workgroup server A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

XAPI events A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

XML CallObject An interoperability capability that enables you to call business functions.

XML Dispatch An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

XML List An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

XML Service An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

XML Transaction An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

XML Transaction Service (XTS) Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

Z event A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

Z table A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

Z transaction Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.

Index

D

E

F

G

H

I

L

M

R

S

T