

Oracle® Retail Plan 2.6.1
Configuration Guide

Copyright © 2006 Oracle. All rights reserved.

Primary Author: Judith Meskill

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Preface

Audience

This document is intended for administrators of the Oracle Retail Plan application.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following documents in the Oracle Retail Plan 2.6.1 documentation set:

- *Oracle Retail Plan Installation Guide*
- *Oracle Retail Plan Operations Guide*
- *Oracle Retail Plan User Guide*
- *Oracle Retail Plan Administration Guide*
- *Oracle Retail Plan Online Help*
- *Oracle Retail Plan Release Notes*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Contents

Preface	iii
Audience	iii
Documentation Accessibility	iii
Related Documents	iv
Conventions	iv
 1 Introduction	
About the Plan Configuration Guide	1-1
What's in This Book	1-1
 2 Getting Started	
Configuration Overview	2-1
 3 User Management	
Introduction	3-1
About User Roles and User Actions	3-1
About User Management Roles	3-2
plansecurityload.sh	3-3
Creating Users and Roles	3-3
The xml Files	3-3
Standard Load Prerequisites	3-4
Shell Script	3-4
User Management Security	3-4
Plan Sample xml Files	3-5
User Sample xml File	3-5
Roles Sample xml File	3-5
Role Assignment Sample xml File	3-6
 4 Business Rule Manager	
Introduction	4-1
Business Rule Manager	4-1
Getting Started	4-2
Default Plan Business Rules	4-3
Business Rule Definitions	4-4

Loading Business Rule Definitions	4-6
Configuring Business Rule Definitions	4-7
Business Rule Instances	4-7
Guidelines for Entering Business Rule Instances	4-7
Custom Attributes	4-8
Business Rule Manager Bulk Loader	4-9
Business Rule Instances Standard Interface Specification (ASH_BRM_INSTANCE_TBL)	4-9
Loading Instances	4-9
Business Rule Manager Properties	4-10
Guidelines for Setting BRM Properties	4-10

5 Database Configuration

Introduction	5-1
Seed Data	5-1
Analytic Configuration	5-2
Merchant Desktop One-Time Configuration	5-3
After the First Merchandise Hierarchy Load	5-3
pl_java_properties	5-3
Storesets	5-4
storesetupdater.sh	5-4
Storesets with Filters Based on CDAs	5-4
Creating Storesets.....	5-4
Interfaces for One-Time Data Inserts	5-5
Pricing Event Types	5-5
Client Hierarchy Actions Standard Interface.....	5-6
Client Hierarchy Actions Specification (CLIENT_HIERARCHY_ACTIONS_TBL)	5-6
Cross Products Information Standard Interface.....	5-6
Cross Products Information Specification (ASH_CP_TBL)	5-6
Location Hierarchy Levels Standard Interface	5-7
Technical Notes	5-7
LH Levels Specification (ASH_LHL_TBL)	5-7
Merchandise Hierarchy Levels Standard Interface.....	5-7
Technical Notes	5-7
MH Levels Specification (ASH_MHL_TBL)	5-7
Pack Hierarchy Levels Standard Interface	5-8
Technical Notes	5-8
PH Levels Specification (ASH_PHL_TBL)	5-8
Optimized History	5-8
Inference Rules	5-9

6 Configurable Data Attributes

Introduction	6-1
Defining Configurable Data Attributes	6-1

7 Merchant Desktop and Retail Data Mart Configuration

Introduction	7-1
---------------------------	-----

Merchant Desktop Properties Settings	7-1
Merchant Desktop and CDAs.....	7-2
Starting and Stopping Components	7-3
Starting and Stopping the Application Servers	7-3
Starting and Stopping the Microstrategy Server	7-3
Starting and Stopping the RMI Server	7-3
Monitoring Merchant Desktop.....	7-3
RDM Data Loads	7-3
Aggregation Levels Configuration.....	7-4
Actual History Load in the DB.....	7-4
Optimized History in RDM.....	7-5
RDM Data Load.....	7-5
RDM Refresh	7-5
Refresh Procedures	7-6
Microstrategy Configuration.....	7-6
Microstrategy Web Preferences	7-6
Upgrading Microstrategy Metadata.....	7-6
Facts/Aggregates	7-7
Merchandise Hierarchy and Location Hierarchy Level Names.....	7-7
Performance Tuning.....	7-7
Microstrategy Server.....	7-8
Reports.....	7-8
Report Metrics.....	7-8

8 Plan Updater

Introduction.....	8-1
About Plan Updater	8-1
Using Plan Updater.....	8-2
Environment Variables.....	8-2
plan.properties File	8-2
planupdater.properties.....	8-2
Plan Updater Script.....	8-3

9 Properties File Configuration

Introduction.....	9-1
Calc Engine Properties	9-2
Message Timeout Properties	9-2
Auto-Population and FNN Properties.....	9-3
Session Timeout Properties	9-3
General Properties.....	9-4
Client Plug-In Properties	9-4
Application Plug-In Properties.....	9-5
Background Event Subsystem Properties.....	9-6
Forecast Properties	9-7
Pack Optimization Properties.....	9-8
Calc Parameters Properties	9-8

Flow View Properties.....	9-9
Login Properties.....	9-9
Auxiliary URL Properties	9-9
Cache Limit Sizes and Cache Management Properties.....	9-10
Internal Developer Properties	9-10
Grid Configuration Properties.....	9-11
Metrics Key List.....	9-11
Miscellaneous Properties.....	9-11
Validation Properties	9-12
Date Properties.....	9-12
Forecast Properties	9-13

10 **Custom Metrics**

Introduction.....	10-1
Plan Default Metrics	10-2

Introduction

This chapter contains the following:

- “About the Plan Configuration Guide” on page 1
- “What’s in This Book” on page 1

About the Plan Configuration Guide

The Plan Configuration Guide provides information about configuring the Plan software to meet a customer’s specific business requirements. Once installation is complete, certain configuration setting must be modified. Some of these modifications must occur before the data can be loaded into the database. Other configuration settings can impact performance or can determine the functionality that is exposed to the Plan user.

What’s in This Book

The Plan Configuration Guide addresses the following topics:

- Chapter 1 - Introduction - an orientation to the Plan Configuration Guide.
- Chapter 2 - Getting Started - an overview of all the configuration points in Plan.
- Chapter 3 - User Management - used to create, modify, and inactivate user accounts.
- Chapter 4 - Business Rule Manager - used to customize business rule settings, that is, the constraints used by Price to determine markdowns and forecasts.
- Chapter 5 - Database Configuration - various post-installation required configurations.
- Chapter 6 - Configurable Data Attributes - used to specify custom data that can be viewed through the Plan UI.
- Chapter 7 - MD/RDM - The Merchant Desktop is an optional in-season management tool that provides access to alerts and reports. The Retail Data Mart is an interface between Plan data and Business Intelligence tools.
- Chapter 8 - Plan Updater - updates existing plans based on data changes.
- Chapter 9 - Properties File -contains a list of the properties found in the plan.properties file, which are configurable.
- Chapter 10 - Custom Metrics - overview of Configurable Metrics Framework and a list of Plan default metrics.

Getting Started

The chapter contains the following:

- “Configuration Overview” on page 1

Configuration Overview

Once you have installed Oracle Retail Plan Pre-Season, which is described in the Installation Guide, and loaded the data, which is described in the Operations Guide, you are ready to configure the application.

The general areas of configuration for Plan include the following:

- User Management - a utility for managing user accounts. Plan comes with four default roles: PLAN_SYSADMIN, PLAN_BIZADMIN, PLANNER, and PLAN_VIEWER.
- Business Rule Manager - a utility for managing system-wide business parameters. Most business rules in Plan are configured only once. The default business rules are AUR Adjustment Percent, Annual Basics Plan Minimum, Safety Stock, Tolerance Percent, OOS Valid Days of Month, Product Group Attribute, Store Quantity High Threshold, and Store Quantity Low Threshold.
- Database Configuration - a number of one-time configurations are necessary, including the running of scripts to populate the database with seed data.
- CDAs - used to assign custom merchandise or location attributes.
- Merchant Desktop - configuration includes properties settings, CDAs, RDM aggregation levels, and Microstrategy. This chapter also includes information on the RDM refresh and performance tuning.
- Metrics - the Configurable Metrics Framework allows you to change and remove existing metrics as well as create new custom ones. A listing of the Plan default metrics is provided here.
- Plan Updater - a nightly batch process that updates existing plans based on data changes.
- Properties File - a list of the configurable properties found in the plan.properties file.

User Management

This chapter contains the following sections:

- “Introduction” on page 1
- “About User Roles and User Actions” on page 1
- “plansecurityload.sh” on page 3
- “User Management Security” on page 4
- “Plan Sample xml Files” on page 5

Introduction

User Management is a utility that lets you create, modify, and remove user accounts from a central location. The User Management utility is installed automatically when you install the application.

Each user who accesses the application must have a user account. Each user account is assigned one or more roles that determine the types of functions the user can perform with the application.

Single sign-on is supported so that users can access the entire suite of products, if they are available, without additional authentication.

About User Roles and User Actions

Roles are defined by a specific set of user actions. The actions that define each role serve to delimit the activities a user can perform. All actions are self-contained. For example, Write does not imply Read. So a role must include all the actions that are necessary for complete functionality.

Plan comes with a default set of actions, loaded into ACTIONS_TBL:

- PLAN_SUBMIT - can submit selected items in a Plan. Requires PLAN_READ and PLAN_WRITE.
- PLAN_DELETE - can delete an existing Plan.
- PLAN_CREATE - can create a new Plan.
- PLAN_READ - can access a plan in read-only mode. All roles should include this action. If a user does not have explicit read permission, that user will not be able to access any plan. A user needs both PLAN_READ and CALC_PARAM_READ to view a plan.

- PLAN_PUBLISH - can publish plans to RDM in advance of the nightly batch process.
- CALC_PARAM_WRITE - can edit calculation parameters, including Like Item, Store Base, Flow Dates, Price Plan, and Pack Configuration. A user needs PLAN_READ, PLAN_WRITE, and CALC_PARAM_READ for the complete functionality of this action.
- CALC_PARAM_READ - can view read-only calculation parameters, including Like Item, Store Base, Flow Dates, Price Plan, and Pack Configuration. A user needs both PLAN_READ and CALC_PARAM_READ to view a plan.
- PLAN_WRITE - Requires PLAN_READ and CALC_PARAM_READ. Can do the following:
 - add items to a Plan
 - add colors to a Plan
 - actualize items in a Plan
 - delete items in a Plan
 - change the status
 - validate CIS
 - manage product groups
 - copy and paste
 - actions on the Assortment View
 - actions on the Item View

Default actions cannot be deleted.

Plan comes with a default set of roles:

- PLAN_SYSADMIN - contains the entire set of Plan default actions.
- PLAN_BIZADMIN - contains the entire set of Plan default actions and BRM default actions.
- PLANNER - contains all Plan default actions except Submit and Publish.
- PLAN_VIEWER - contains only the two Plan default read actions, both of which are necessary.

Roles are assigned to users with restrictions that are defined at or above a specific node of the merchandise hierarchy and the location hierarchy. The scope of actions can be across the merchandise and location hierarchies. The scope must be defined at or above the class planning level.

For Plan, the entire location hierarchy is within the scope of every action. All Plan roles should be defined with a location scope of CHAIN. Plan roles should never be set below the CHAIN level in the location hierarchy.

The sample file, "Role Assignment Sample xml File" provides an illustration of defining the scope.

About User Management Roles

User accounts with user management roles have access to features such as creating users, assigning roles, removing user accounts, resetting passwords.

When a user with a user management role logs on, a link to the User Management utility appears on the Main Menu.

The following list describes the default User Management roles:

- **UM_READ_ONLY_ADMIN** – This role allows read-only access to the User Management utility. This role has privileges to view the list of users and their roles and hierarchy levels, but not to create new user accounts or modify or inactivate existing ones.
- **UM_ROLE_ASSIGN_ADMIN** – This role allows assigning new roles (and related hierarchy levels) to existing user accounts, but it does not allow the creation of new user accounts.
- **UM_USER_ADMIN** – This role allows creating new user accounts, but it does not allow the assignment of roles to the new accounts.

plansecurityload.sh

If you are creating a small number of user accounts using the default actions, you can create those accounts using the User Management console. (For more information on using the User Management utility, consult the application User Guide.) However, if you want to create user accounts for a group of users all at one time, you can use the User Management plansecurityload.sh script.

Creating Users and Roles

You need to create and validate (using a tool like XML Spy) three xml files containing entries for Users, Roles, and Role Assignments.

Note: The Business Rule Manager actions associated with roles must be created, using brmadmin.sh, in order for the roles to be successfully created.

- The user file contains user names. All user names must be unique. The schema includes a flag that indicates whether or not the password should be hashed.
- The Roles file contains the possible roles that can be assigned. All role keys must be unique. The action key attributes must be loaded into the database before the load script can be used. All elements and attributes must be lower case.
- The Role Assignment file contains user names and the role or roles associated with the user name. The user names must be loaded into the database before this file can be processed by the load script. All elements and attributes must be lower case. The merchandise ID and the Location ID are provided by a pipe-delimited string of CLIENT_LOAD_ID, as found in the MERCHANDISE_HIERARCHY_TBL or LOCATION_HIERARCHY_TBL.

The information in the three files is loaded into database tables by the load script. (Users and Role Assignments can be added or modified via the application UI. Roles can only be added or modified via the load script.)

The xml Files

The xml schemas and samples of the three required xml files can be found in <installed>/modules/tools/conf.

Table 3–1 User Management xml Files

Schema	xml File	Database Table
user.set.xsd	plan_user_set.xml	USERS_TBL
role.set.xsd	plan_role_set.xml	ROLES_TBL
role-assignment.set.xsd	plan_assignment_set.xml	USER_RESOURCE_ROLE_TBL

Standard Load Prerequisites

Before you run the User Management load script, you must have run the standard load so that the merchandise hierarchy table and the location hierarchy table have been populated. (For more information on the standard load, see the application Operations Guide).

Shell Script

The shell script for running the User Management load script is located in `<installed>/modules/tools/bin/plansecurityload.sh`. It can be run without arguments. The script picks up the three xml files and loads them onto the application server in the `jndi.properties` file. The script expects the three xml files to be in `<installed>/modules/tools/conf`, so if the files are in a different location, you must supply the complete path for the new location. All three xml files are expected by the script. The script succeeds silently, but provides error messages if necessary.

Usage:

plansecurityload.sh [security-data-dir] [app-server-home] [um-script-dir] [jndi-url] [init-ctx-factory]

where

security-data-dir	The directory containing the UM security configuration files (e.g., <code>plan_user_set.xml</code>) to load
app-server-home	The application server home directory (e.g., <code>c:/bea8/weblogic81/server</code>)
um-script-dir	The directory where the UM loader script is located (e.g., <code>/modules/tools/bin</code>)
jndi-url	The jndi url for the application server. This defaults to <code>t3://localhost:7001</code>
init-ctx-factory	The initial context factory class for the server. This defaults to <code>weblogic.jndi.WLInitialContextFactory</code>

User Management Security

In order to ensure the security of the application, the following security features are available in User Management:

- The AUTOCOMPLETE attribute is configurable on forms where passwords or user names are entered. By default, AUTOCOMPLETE is set to ON, so that sensitive information is stored.

`<ConfigRoot>/suite/suite.properties/suite.loginform.autocomplete = ON`

- The session time out value is set in suite.httpsession.timeout. By default, it is set to 1800 seconds.

```
<ConfigRoot>/suite/suite.properties/suite.httpsession.timeout = 1800
```

- The configure login time out value is independent of the session time out and should be of a shorter time period than the session time out. If configure time out value is not set, it defaults to the session time out value. By default, it is set to 120 seconds.

```
<ConfigRoot>/suite/suite.properties/suite.userlogin.timeout = 120
```

- The attribute on the session ID cookie is set for secure deployments only so that the cookie can be transmitted via HTTPS and over an encrypted network. The default value is FALSE.

```
<ConfigRoot>/suite/suite.properties/suite.cookie.secure = FALSE
```

Plan Sample xml Files

This section provides sample input files for adding or updating users and roles.

User Sample xml File

This is the sample data in plan_user_set.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
- <user-set hash-passwords="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="user-set.xsd">
  <user username="view" password="view" last-name="Viewer" first-name="Joe"
middle-initial="R" employeeID="1" title="El Presidente"/>
  <user username="willis" password="willis" last-name="Submitter" first-
name="Jane" middle-initial="Y" employeeID="2" title="serf"/>
  <user username="max" password="max" last-name="Approver" first-
name="Nancy" middle-initial="R" employeeID="3" title="El Presidente"/>
  <user username="chris" password="chris" last-name="user" first-
name="test" middle-initial="U" employeeID="4" title="serf"/>
  <user username="mary" password="mary" last-name="Franklin" first-
name="Aretha" middle-initial="A" employeeID="5" title="Respect"/>
</user-set>
<!-- This XML supports adding/replacing "users" for the User Management
subsystem. -->
--<!--

Note:
1. User username must be unique among all applications.
2. <user-set> has a flag indicating whether the password should be hashed
prior to persistence. This supports migration from prior
implementations of the application so that users can keep existing
passwords
3. Passwords must be alphanumeric
```

Roles Sample xml File

This is the sample data in plan_role_set.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
-<role-set xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="role-set.xsd">
  <role key="PLAN_BIZADMIN">
    <action key="PLAN_SUBMIT"/>
    <action key="PLAN_PUBLISH"/>
```

```

    <action key="PLAN_DELETE" />
    <action key="PLAN_CREATE" />
    <action key="PLAN_WRITE" />
    <action key="PLAN_READ" />
    <action key="CALC_PARAM_WRITE" />
    <action key="CALC_PARAM_READ" />
    <action key="BRM_AUR_ADJUSTMENT_PCT_VIEW" />
    <action key="BRM_AUR_ADJUSTMENT_PCT_EDIT" />
    <action key="BRM_ANNUAL_BASICS_PLAN_MIN_VIEW" />
    <action key="BRM_ANNUAL_BASICS_PLAN_MIN_EDIT" />
    <action key="BRM_SAFETY_STOCK_VIEW" />
    <action key="BRM_SAFETY_STOCK_EDIT" />
    <action key="BRM_TOLERANCE_PCT_VIEW" />
    <action key="BRM_TOLERANCE_PCT_EDIT" />
    <action key="BRM_OOS_VALID_DAYS_MONTH_VIEW" />
    <action key="BRM_OOS_VALID_DAYS_MONTH_EDIT" />
    <action key="BRM_PRODUCT_GROUP_ATTR_VIEW" />
    <action key="BRM_PRODUCT_GROUP_ATTR_EDIT" />
  </role>
- <role key="PLAN_SYSADMIN">
  <action key="PLAN_SUBMIT" />
  <action key="PLAN_PUBLISH" />
  <action key="PLAN_DELETE" />
  <action key="PLAN_CREATE" />
  <action key="PLAN_WRITE" />
  <action key="CALC_PARAM_WRITE" />
  <action key="CALC_PARAM_READ" />
  <action key="PLAN_READ" />
</role>
- <role key="PLANNER">
  <action key="PLAN_DELETE" />
  <action key="PLAN_CREATE" />
  <action key="PLAN_WRITE" />
  <action key="CALC_PARAM_WRITE" />
  <action key="CALC_PARAM_READ" />
  <action key="PLAN_READ" />
</role>
- <role key="PLAN_VIEW">
  <action key="PLAN_READ" />
  <action key="CALC_PARAM_READ" />
</role>
</role-set>
<!-- This XML supports adding/updating "roles" in the User Management
      subsystem. -->
- <!--
  Note:
  1. All role keys must be unique among all applications. Names like
     PRICE_APPROVER, PLAN_EDITOR, and PLACE_READER are expected.
  2. The action key attributes must be present in the DB
     before the bulkloader is run.
  3. All elements and attributes are case sensitive and all are lower case.

```

Role Assignment Sample xml File

This is the sample data in plan_role_assignment_set.xml

```

<?xml version="1.0" encoding="UTF-8"?>
- <role-set xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="role-assignment-set.xsd">
  - <role key="PLAN_BIZADMIN">

```

```

    -<user-assignment username="view">
      <node location=""merchandise=""/>
    </user-assignment>
    -<user-assignment username="mary">
      <node location=""merchandise=""/>
    </user-assignment>
  </role>
-<role key="PLAN_SYSADMIN">
  -<user-assignment username="willis">
    <node location="" merchandise=""/>
  </user-assignment>
</role>
-<role key="PLANNER">
  -<user-assignment username="max">
    <node location="" merchandise=""/>
  </user-assignment>
</role>
-<role key="PLAN_VIEWER">
  -<user-assignment username="chris">
    <node location="" merchandise=""/>
  </user-assignment>
</role>
</role-assignment-set>
-<!--

```

Note:

1. All role keys must be unique among all applications. Names like PRICE_APPROVER, PLAN_EDITOR, and PLACE_READER are expected. They must match those already persisted into the DB.
2. The Users with a given username must be present in the DB prior to this file being processed by the bulkloader.
3. The location and merchandise attributes are pipe delimited strings of client load IDs. The first node is just below the root (Chain Level) node. An empty attribute represents a chain level assignment.
4. All elements and attributes are case sensitive and all are lowercase.

Business Rule Manager

This chapter contains the following sections:

- “Introduction” on page 1
- “Business Rule Manager” on page 1
- “Getting Started” on page 2
- “Business Rule Definitions” on page 3
- “Default Plan Business Rules” on page 3
- “Custom Attributes” on page 7
- “Loading Business Rule Definitions” on page 6
- “Configuring Business Rule Definitions” on page 7
- “Business Rule Instances” on page 7
- “Business Rule Manager Bulk Loader” on page 9
- “Business Rule Manager Properties” on page 10

Introduction

The Business Rule Manager (BRM) provides a centralized interface that can be used to configure system parameter related to business logic in the Plan application. Business rules are set by administrative users. Most rules in Plan are set once and then not changed again. Rules can be set at any intersection of the Merchandise Hierarchy and the Location Hierarchy. Lower levels of the hierarchy inherit values set at higher levels unless a specific value for that level is specified in the BRM. Custom attributes can also be configured for any business rule. This approach provides flexibility and means, for example, that different stores can have different flow plans.

This chapter explains how to configure the business rules, using the Business Rule Manager.

Business Rule Manager

The Business Rule Manager is a Plan utility that is used to view and change business rule settings. Business rules can be configured in Plan to provide flexibility in flow plan calculations.

The application provides a file that contains the business rule definitions. The business rule definitions specify the constraints that apply to business rule instances (mappings between location and merchandise hierarchy levels and business rule values). The

definitions are configurable; however, most of the business rules have default values that can be used to perform any initial application work.

The BRM is accessed through the application Main Menu. A user's ability to view and change business rule settings is specified by the permissions attached to the user role(s) assigned to them. These roles are assigned using the User Management utility. (For more information, see the application Online Help.) The actions used by BRM roles are defined in the business rule definition file (discussed later in this chapter).

The BRM is used to:

- View current business rule settings for specific items
- Change business rule settings
- View the history of business rule changes
- View a business rule value that was in effect for a particular date. The UI displays all the rule values that would apply via inheritance. The value on the target date is the one with the highest precedence.

For more information on the user interface to the BRM, see the application User Guide.

Getting Started

Once Plan is installed, you must configure the business rule definitions, if it is necessary to change the default values or make any other changes, and then load them. The default business rule definitions are contained in `/modules/tools/conf/DefaultRules/plan_rule_definitions.xml`.

Plan provides three sample rules files in the installation directory:

- `/modules/tools/conf/SampleRules/plan_rule_definitions.xml` (Old Hierarchies)
- `/modules/tools/conf/SampleRules/plan_rule_definitions_flexco.xml` (Flexible Hierarchies)
- `/modules/tools/conf/SampleRules/plan_rule_definitions_with_attrs.xml`

Once you have edited the rule definition file, you can use `/modules/tools/bin/brmadmin.sh` to load the file into the BRM.

Note: Two Plan business rules are optional, `ANNUAL_BASICS_PLAN_MIN` and `OOS_VALID_DAYS_OF_MONTH`.

Note: Plan only supports setting location-level mapping at `CHAIN` for the location hierarchy except for `SAFETY_STOCK`, which can be set below `CHAIN`.

Default Plan Business Rules

Plan is configured, by default, with 8 default business rules accessible through the BRM.

Table 4–1 *Default Plan Business Rules*

Business Rule Name and UI Display Name	Business Rule Description	Default Value
AUR_ADJUSTMENT_PCT AUR Adjustment Pct	The average unit retail discount percent. Used, for example, in the display of new pricing plans.	0.025
ANNUAL_BASICS_PLAN_MIN Annual Basics Plan Min	The minimum annual receipt quantity. Used to calculate the minimum value for new basic items and to validate basic items. An optional business rule.	12
SAFETY_STOCK Safety Stock	The additional time, in days, to stock an item so that it is not out-of-stock.	14
TOLERANCE_PCT Tolerance Pct	The acceptable variance, expressed as a percent, from the Store Financial Plan.	.1
OOS_VALID_DAYS_OF_MONTH OOS Valid Days of Month	List of days that are valid as out-of-stock days. Expressed as days of the month. Used in out-of-stock data validation. An optional business rule.	1,15

Table 4–1 (Cont.) Default Plan Business Rules

Business Rule Name and UI Display Name	Business Rule Description	Default Value
PRODUCT_GROUP_ATTR Product Group Attribute	Indicates whether Product Groups are auto-generated or not (the default). If auto-generated, also indicates which merchandise column is the source.	None
STORE_QTY_LOW_THRESHOLD	Sets the lower threshold for Store-level quantities that will be flagged as a potential concern. Expressed as a percentage. Any stores that deviate from the average-per-store by this value will be flagged as Yellow in the Assortment View - Store Quantities screen.	0.5
STORE_QTY_HIGH_THRESHOLD	Sets the upper threshold for Store-level quantities that will be flagged as a potential concern. Expressed as a percentage. Any stores that deviate from the average-per-store by this value will be flagged as Red in the Assortment View - Store Quantities screen.	0.75

Business Rule Definitions

You may want to configure the business rules to meet the needs of your business. The sample file, located in `/modules/tools/conf/SampleRules`, provides an illustration of a set of business rules, including a configured sample attribute and some test rules that illustrate validation constraints. You can use this file as an advanced example of some possible approaches to take when planning your own configuration. However, your customization should be based on the default business rules. An editable copy of the business rule definitions can be found in `config/businessrulemgr/plan_rule_definitions.xml`. Once you have edited this file, you can use `/modules/tools/bin/brmadmin.sh` to reload the file in order to implement the changes you have made.

The xml schema for the business rule definitions file is located in `tools/brmadmin/conf/brm_config.xsd`

Here is a sample business rule definition, including two attributes, taken from `/modules/tools/conf/SampleRules/plan_rule_definitions.xml`:

```
<AttributeInfo name="PURCH_TYPE"
  table="MERCHANDISE_HIERARCHY_TBL"
  shortDescription="brm.rules.attribute.attr1.label"
  longDescription="brm.rules.attribute.attr1.description"
  allowOtherValues="N"/>
```

```

<AttributeInfo name="MERCHANDISE_SUBTYPE"
  table="MERCHANDISE_HIERARCHY_TBL_TBL"
  shortDescription="brm.rules.attribute.attr2.label"
  longDescription="brm.rules.attribute.attr2.description"
  allowOtherValues="N" />
<RuleDefinition name="AUR_ADJUSTMENT_PCT"
  shortDescription="brm.rules.params.auradjustmentpct.label"
  longDescription="brm.rules.params.auradjustmentpct.description"
  readAction="BRM_AUR_ADJUSTMENT_PCT_VIEW"
  editAction="BRM_AUR_ADJUSTMENT_PCT_EDIT"
  maxMerchandiseFetchLevel="COLOR"
  maxLocationFetchLevel="CHAIN"
  <KeyLevel merchandiseLevel="CHAIN"
    locationLevel="CHAIN"
    matchAttribute1="N"
    matchAttribute2="N" />
  <KeyLevel merchandiseLevel="CLASS"
    locationLevel="CHAIN"
    matchAttribute1="N"
    matchAttribute2="N" />
  <KeyLevel merchandiseLevel="CLASS"
    locationLevel="CHAIN"
    matchAttribute1="Y"
    matchAttribute2="Y" />
  <ValueDefinition valueType="FLOAT"
    validationType="NONE"
    shortDescription="brm.rules.value.auradjustmentpct.label"
    longDescription="brm.rules.value.auradjustmentpct.description"
    allowNullValues="N"
    defaultValue=".025">
  </RuleDefinition>

```

Each business rule definition contains the following information:

- The name of the business rule, in this case AUR_ADJUSTMENT_PCT
- The short description resource ID for the business rule's name, which is displayed in the UI.
- The long description resource ID for the business rule description, which is displayed when a user hovers over the name in the UI.
- The read action and the write action associated with the business rule. Roles, which are assigned to specific users and determine their permissions, are made up of actions. In order for users to be able to view and/or edit a business rule in the UI, they must be assigned a role that includes some combination of the following actions at the desired level or higher:

Note: BRM_EDIT requires BRM_VIEW.

- BRM_AUR_ADJUSTMENT_PCT_VIEW
- BRM_AUR_ADJUSTMENT_PCT_EDIT

For more information on actions and roles, see the application Online Help.

- The fetch levels are not currently used.
- An arbitrary number of key levels, which specify at what levels an instance of the business rule can be set. Each key level contains a merchandise hierarchy level, a location hierarchy level, and optional custom attributes that are used to determine the match between an item and a rule. To determine the rule mapping, matching occurs in the following order of precedence:

1. Search the merchandise hierarchy from low to high for a match.
2. Search the location hierarchy from low to high for a match.
3. If an attribute is set to Y, match that item's value.
4. If a attribute is set to N, match any attribute value.

For the example rule definition shown above, matching of rule to item occurs at the Chain Chain level with any attribute, at the Worksheet Worksheet level with any attribute, and at the Worksheet Worksheet level with the Vendor attribute.

For the example rule definition shown above, matching of rule to item occurs at the Chain Chain level with any attribute, at the CLASS CHAIN level with any attribute, and at the CLASS CHAIN level with both attributes.

- The type of value for the rule:
 - Integer
 - Floating point number
 - Date
 - String
- Validation, by range, enumeration, or none. If range, then the minimum and maximum values are given. If enumeration, a list of values is provided.
- Whether or not null values are allowed.
- The default value for the rule. If no default value is assigned, then NULL is assumed.
- If range is being used for validation, in combination with a valid type, the minimum and maximum values of the range are provided.

Loading Business Rule Definitions

When you first begin using the application and whenever you make changes, you must load the business rule definitions file into the database, using `brmadmin.sh`.

Here is the usage for the `brmadmin.sh` script.

Server Mode (the default), which sends the request to the application server:

```
brmadmin.sh [-server] <config_root> <rule_definitions> [<host> <port>]
```

Client Mode, which processes the request on the client side:

```
brmadmin.sh [-client] <config_root> <rule_definitions>
```

where

- `<config_root>` – the root directory of the application configuration files.
- `<rule_definitions>` – the name of the xml file that contains the rule definitions.
- `<host>` - the application server host
- `<port>` - the application server port
- `-h` - displays help message
- `-p` - disables execution of database load procedures

The file must contain the complete set of definitions. Definitions for rules not contained in the file are removed.

Business rule instances are affected when you modify business rule definitions. If you change rule value types, business rule instances may be deleted. In addition, changes to definitions may cause inconsistencies between the rules and the instances. As a result, the application may not perform properly.

Configuring Business Rule Definitions

When configuring business rules to meet business needs, consider the following:

- When configuring key levels, you must manage the settable levels in conjunction with the inheritance hierarchy and user access.
- Editing business rule definitions to change validations or default values may affect rule instances.
- Editing business rule definitions to change validations or default values may affect system performance.
- If you add a new business rule or change an existing one, you may need to add resources or modify the grid configuration.

Business Rule Instances

A business rule instance is a specific mapping between a key and a rule value. When BRM is installed, instances for the business rules exist at the top level and have the default values assigned to them (even if the top level is not a settable key level as defined in the business rule definition). If a business rule instance is deleted, the object that was assigned that instance will then inherit the settings of the instance at the next higher precedence level in the hierarchy. If the top level is deleted, the instance returns to the default value in the business rule definition file.

Guidelines for Entering Business Rule Instances

You can enter values for business rules either by using the BRM application or the BRM API. Both methods validate the instance against the BRM rule definitions. When using the BRM, you must be assigned a role that permits you to make changes to business rule values. For more information on Roles, see the application Online Help.

Business rule instances must be consistent with business rule definitions:

- Instances must be settable at the desired level, as defined in the rule definitions.
- Instances must conform to the validations defined in the rule definitions, which include the value type.
- Each instance must have an associated business rule definition.
- The key level of each instance must be permitted by the rule definition.
- The attribute values used in the instance keys should be consistent with the attributes in the BRM configuration.

Custom Attributes

Attributes are optional variables that can be added to a specific business rule definition. Two attributes are permitted. Attributes extend the business rule key and are used to determine the match between a rule and an item. Custom attributes should be added to the `plan_rule_definitions.xml` file.

The attribute definition includes:

- The attribute name, which must be consistent with the column name in the source table.

Note:

When configuring attributes for Plan, merchandise attributes should be specified as sourced from `MERCHANDISE_HIERARCHY_TBL` or `MERCH_ATTRS_TBL`. Plan uses the attribute values from `PLANNED_ITEMS_TBL` if an analogous column exists in that table.

If an item is actualized, its merchandise ID is used to fetch BRM values for rules that are fetched at the `STYLE` or `STYLE/COLOR` level. If the item is mock, the merchandise ID of the level above `STYLE` is used. The attributes from `PLANNED_ITEMS_TBL` are used for both mock items and actualized items.

- The name of the table that includes the column used for the attribute name. The following tables can be used:
 - `MERCHANDISE_HIERARCHY_TBL`
 - `MERCH_ATTR_TBL`
 - `LOCATION_HIERARCHY_TBL`
 - `LOCATION_ATTR_TBL`
- The resource ID for the attribute's name, which is displayed in the UI.
- The resource ID for the attribute description, which is displayed when a user hovers over the name in the UI.
- Whether an attribute value other than one from the current set of values is valid.

To configure custom attributes, you should define the resources used for their display as part of `businessrulemgrResources.properties`:

```
# Rules grid - Attributes
brm.rules.attribute.group.label=Attributes
brm.rules.attribute.group.description=Attributes
brm.rules.attribute.attr1.label=Purchase Type
brm.rules.attribute.attr1.description=Purchase Type
brm.rules.attribute.attr2.label=Merchandise SubType
brm.rules.attribute.attr2.description=Merchandise Subtype
```

Once the custom attributes have been defined, you must run `com.profitlogic.db.birch.LoadBRMAttributeValues` after you run `brmadmin.sh` in order to see the custom attributes changes in the application. `LoadBRMAttributes` loads values into `BRM_ATTRIBUTE_VALUE_TBL`. The application derives the values for the attributes displayed on the BRM page from this table. You will also need to run the script again if the merchandise hierarchy or the location hierarchy have new attributes that you want use.

Business Rule Manager Bulk Loader

The BRM Bulk Loader provides a means for staging and loading a set of business rule instances. This utility is included within the standard interface and standard load (for more information, see the application Operations Guide), but can also be implemented separately if new or updated business rule instances need to be loaded outside the normal scheduled batch processes. The Bulk Loader validates the business rule instances according to the guidelines described in “Guidelines for Entering Business Rule Instances” on page 7.

Business Rule Instances Standard Interface Specification (ASH_BRM_INSTANCE_TBL)

The data to be loaded by the Business Rule Manager bulk loader utility must conform to the following standard interface specification.

The merchandise and location keys map to the CLIENT_LOAD_ID. The merchandise and location levels map to LEVEL_DESC. The rule name is the name of the business rule as specified in the business rule definition. The rule value is the value assigned to the business rule instance. The attribute values are the specific values for the custom variables, which have been derived from columns in the permitted source tables. The delete flag defines whether the instance is to be deleted (a value of 1) or added/updated (a value of 0 - the default).

Table 4–2 Business Rule Instances Standard Interface Specification

Attribute	Attribute Description	Data Type	Maximum Length	Nullable Y/N
MERCHANDISE_KEY	Key for this level of the hierarchy	String	50	N
MERCHANDISE_LEVEL	ID for this level of the hierarchy	String	50	N
LOCATION_KEY	Key for this level of the hierarchy	String	50	N
LOCATION_LEVEL	ID for this level of the hierarchy	String	50	N
RULE_NAME	The name of the business rule associated with the item.	String	64	N
RULE_VALUE	The business rule value assigned to the item.	String Values < 1 should be expressed as 0.n.	100	N
ATTRIB1_VALUE	The specific value associated with the item for custom attribute 1.	String	100	Y
ATTRIB2_VALUE	The specific value associated with the item for custom attribute 2.	String	100	Y
DELETE_FLAG	A flag to indicate whether the instance is to be deleted or inserted. 0 = insert (the default). 1 = delete.	Integer	1	N

Loading Instances

The Standard Load scripts that stage and load the data into the application stage and load business rule instances. In order to invoke the BRM Bulk Loader utility separately, as a manual process, do the following:

```
bash pl_stage_file.sh --controldir=<directory with control files> --logdir=<log output directory> <file containing standard interface-compliant BRM rule instances>
```

```
bash pl_load_data.sh --logdir=<log output directory>
“com.profitlogic.db.birch.LoadBRInstances”
```

The utility validates whether or not the instance key is a legal key at the specified level and whether the instance value is a legal value, as specified in the definition. If the validation fails, the procedure terminates and no changes are made.

Note: Business rule definitions are contained in `config/businessrulemgr/plan_rule_definitions.xml` are loaded using `bradmin.sh`.

Business Rule Manager Properties

BRM properties may need to be configured prior to the deployment of the application. The properties are located in `configroot/businessrulemgr/businessrulemgr.properties`. The settings in this file can be overwritten by client settings.

Table 4–3 Business Rule Manager Properties

Property	Description	Default Value
<code>numBrowsableMerchLevels</code>	The number of merchandise hierarchy levels that can be browsed in the BRM UI.	4
<code>numBrowsableLocLevels</code>	The number of location hierarchy levels that can be browsed in the BRM UI.	2
<code>numFindableMerchLevels</code>	The number of additional merchandise hierarchy levels that can be accessed using the BRM find feature.	2
<code>numFindableLocLevels</code>	The number of additional location hierarchy levels that can be accessed using the BRM find feature.	1
<code>numExpandableMerchLevels</code>	The number of levels that the merchandise hierarchy can be expanded to in the BRM UI.	4
<code>numExpandableLocLevels</code>	The number of levels that the location hierarchy can be expanded to in the BRM UI.	3

Guidelines for Setting BRM Properties

Use the following guidelines in planning the configuration of the BRM properties:

- The number of browsable merchandise hierarchy levels should equal the planning level number.
- The number of findable merchandise hierarchy levels should equal (the total number of merchandise levels – the number of browsable merchandise hierarchy levels down to color).
- The number of findable merchandise hierarchy levels should equal (the total number of merchandise levels – the number of browsable merchandise hierarchy levels).
- The number of findable location hierarchy levels should equal (the total number of location hierarchy levels – the number of browsable location hierarchy levels).
- The number of expandable merchandise hierarchy levels should equal the number of browsable merchandise hierarchy levels.
- The number of expandable location hierarchy levels should equal the number of browsable location hierarchy levels.

In addition, keep in mind that

- The BRM validates that the total number of levels defined in the properties file does not exceed the number of levels defined in the database.

- To forestall performance or memory problems, set the number of levels in the properties file close to Class in the merchandise hierarchy.
- The default values for the following parameters in `<configroot>/suite/suite.properties` may need to be configured:
 - `common.hierarchy.fetch.merch.maxlevels`
 - `common.hierarchy.fetch.loc.maxlevels`
 - `common.hierarchy.cache.timeout.hours`

Database Configuration

The chapter contains the following:

- “Introduction” on page 1
- “Seed Data” on page 1
- “After the First Merchandise Hierarchy Load” on page 3
- “pl_java_properties” on page 3
- “Storesets” on page 4
- “Interfaces for One-Time Data Inserts” on page 5
- “Optimized History” on page 8
- “Inference Rules” on page 9

Introduction

This chapter provides details about various database configuration points, including scripts for populating the database with seed data, inference rule customization, and storeset customization.

Seed Data

The following sql scripts need to be configured and run before the first data load in order to initialize Plan with seed data. Note that the gridResources.properties files should be updated to match these settings.

- planned_item_types.sql. Populates PLANNED_ITEM_TYPES_TBL. Used to classify types of items. Values include F (Fashion), B (Basic), and K (Key). The information is used by the drop-down menu on the Plan Worklist page.
- media_types.sql. Populates MEDIA_TYPES_TBL. Used to specify the type of media used in a promotion. Values include NPP (National Preprint), CI (Circular), DM (Direct Mail), TV (Television), MAG (Magazine), WEB (Web), and ROP (Run of Press).
- emphasis_types.sql. Populates EMPHASIS_TYPES_TBL. Used to define how an item is displayed during a promotion. Values include CF (Co-Featured), F (Featured), FLT (Float), CB (Copy Block), LL (Line Listed), OMB (Omnibus), SF (Sub-Featured), and SI (Supplemental Information).
- page_indicator_types.sql. Populates PAGE_INDICATOR_TYPES_TBL. Used to indicate promotion ad placement. Values include F (Front page) and B (Back page).

- `pricing_types.sql`. Populates `PRICING_TYPES_TBL`. Details about this table are provided in this chapter.
- `pl_dd_attributes.sql`. Enables Merchandise Hierarchy and Location Hierarchy CDAs, which are disabled by default.
- `data_levels_tbl.sql`. Provides the Size Profile level to the Calc Engine. This value must be the same as the Size Profile level value in `CLIENT_HIERARCHY_ACTIONS_TBL`.
- `client_hierarchy_actions_tbl.sql`. Populates `CLIENT_HIERARCHY_ACTIONS_TBL`, which is described in this chapter. This table is used to map a retailer's hierarchy and level of operation to Plan.
- `cp.sql`. Populates `ASH_CP_TBL`, which is a legacy table that is required by the application. No configuration is necessary.
- `set_threshold_levels.sql`. Used for setting error thresholds for data feeds.
- `lhl.sql`. Populates `ASH_LHL_TBL`, which is described in this chapter. Used to specify a retailer's location hierarchy levels and their order.
- `mhl.sql`. Populates `ASH_MHL_TBL`, which is described in this chapter. Used to specify a retailer's merchandise hierarchy levels and their order.
- `phl.sql`. Populates `ASH_PHL_TBL`, which is described in this chapter. Used to specify a retailer's pack hierarchy levels and their order.
- `isc_facilities_load.sql`. Describes attributes of a distribution center, such as ID, client key, and location.
- `store_attribute_names_tbl.sql`. Lists the location hierarchy attributes, including CDAs, that are used to define storeset filters in Store Management. also lists the names that appear in the UI. For more information, see "Storesets" on page 4.
- `store_set_data.sql`. Used to create any necessary storesets for a retailer. In order for the Plan application to function, it is necessary to create a storeset called `store_ad_group`. This storeset can only be created after the first data load because the script needs all the stores and their attributes. The script creates a dynamic storeset that has subsets that use the distinct value for the `store_ad_group` attribute of stores. the `store_ad_group` storeset is used by an end user to associate item promotions with specific subsets in this storeset.
- `isc_load_CHLevels.sql`. Processes staged data from the scripts for merchandise hierarchy levels, location hierarchy levels, and pack hierarchy levels, validates the data, and inserts the data into `CLIENT_HIERARCHY_ACTIONS_TBL`.

Analytic Configuration

Inventory effect is a number that represents the fraction of sales lost due to insufficient inventory in a store. An inventory effect of 1.0 means there was no impact on sales because of insufficient inventory.

In order for inventory effect to be turned on, a setting in `delphi.properties` needs to be configured. The property, `delphi.inventoryEffectModel`, is located in `$CONFIGROOT/Engine/delphi.properties`.

Five inventory effect models are supported by the Calc Engine. The specific model used by a Plan implementation is determined by Analytical Services. The five models are:

- none
- piecewiseLinear
- piecewiseLinearByStoreCount
- exponentialByStoreCount
- sigmoidByStoreCount

Note that there are two piecewise linear settings. The piecewiseLinearByStoreCount setting has a correction based on how many stores an item is currently selling in.

An example of assigning a value to the inventory effect property is as follows:

```
delphi.inventoryEffectModel=piecewiseLinear
```

Merchant Desktop One-Time Configuration

After installation, you must configure the Merchant Desktop as follows:

1. Remove the following permission from the administrator role:
delete from role_action_tbl where
role_id in (select role_id from roles_tbl where DESCRIPTION = 'MD_ADMINISTRATOR')
and
action_id in (select action_id from actions_tbl where DESCRIPTION = 'MD_ALERTS_ADMINISTRATOR')
2. Remove the following component by making it hidden:
update MD_PORTLETS_TBL set available=0, hidden=1
where portlet_name like 'pl-Alerts'

After the First Merchandise Hierarchy Load

A one-time configuration step is necessary after the first Merchandise Hierarchy load.

The Merchandise Hierarchy standard interface does not include a field for FIRST_CREATE_DT, but the date on which the merchandise first appeared is required. The Merchandise Hierarchy standard load procedure sets the value of this field for the date on which the load occurs, but only for merchandise that does not already have a FIRST_CREATE_DT. If a value already exists, that existing value is retained.

After the first Merchandise Hierarchy Load, the FIRST_CREATE_DT for all historic merchandise must be set back two years for all merchandise. Any merchandise loaded after the first load will have the correct date.

pl_java_properties

pl_java_properties is an external properties file that contains database settings, is used by stored procedures, and is configurable. This file contains properties loaded from the following property files:

- dbErrorStrings.properties
- yapp_db2.properties or yapp_hsqldb.properties or yapp_oracle.properties
- log4j.properties

- load_helper.xml
- dbError.properties
- cdw.modelstart.properties

For details about configuring dbError.properties, see the Plan Operations Guide.

Storesets

This section includes details about Storeset configuration.

storesetupdater.sh

The storesetupdater utility is used to refresh all dynamic storesets based on any new data feeds. Changes to storesets that require using this utility include changes to store attributes or the addition of a new store. This utility should be run after the nightly data load and before the plan updater utility. It should be included as part of the automation process. The application must be running when this utility is being used.

The script, storesetupdater.sh, is found in <Install Base>/modules/tools/bin, and is normally run without arguments. The script picks up the application server location from jndi.properties.

If the script fails for some reason, it can be run as follows:

```
storesetupdater.sh -u jndiURL
```

where jndiURL is formatted as t3://<host>:<port>

Storesets with Filters Based on CDAs

Location CDAs/storesets/filters. If storesets with filters based on CDAs are created, then the following configuration is necessary:

1. Update store_attribute_names_tbl for each implemented CDA (the entries already exist in the table for CDAs - they just have filtering disabled by default)

Change the filterable_flag field to 1

Update the Display_Name field to reflect the desired label for the CDA - this governs the attribute name in the filter drop-down on the Filters tab in Storeset Administration.

The Subset Filter display names are not resourced so this table must be changed for any standard location attribute names that are changing.

2. Add the attribute columns to the storelayout-grid.xml file if they should be displayed in the Stores tab.
3. Update the gridResources.properties file in the Storesets configuration directory with the desired CDA labels, which will take care of the column headings and the customize table labels on the Stores grid.

Creating Storesets

After the location hierarchy has been loaded for the first time, you must create two admin storesets: ALL and store_ad_group. Subset filters for the store_ad_group storeset must be specified, based on the distinct values of the store_ad_group column in LOCATION_HIERARCHY_TBL.

After you create the store_ad_group storeset, you must assign the store_set_id for the storeset, which can be found in STORE_SETS_TBL, to the plan.params.pricingPlan.storesetID property in plan.properties.

Interfaces for One-Time Data Inserts

The following tables detail the specifications for some of the scripts discussed in “Seed Data” on page 1.

Pricing Event Types

The PRICING_TYPES_TBL provides customizable values that are required by the PLANNED_PROMOS view.

Table 5–1 PRICING_TYPES_TBL

Name	Description	Data Type	Maximum Length	Nullable Y/N
TYPE	The type of pricing event.	String	20	N
PRECEDENCE_ORDER	The order in which the events should be processed when determining the ticket price.	Integer		N
TYPE_INDICATOR	0 = Markdown (Permanent) 1 = Promotion (Temporary)	Integer	1	N
IS_CORPORATE_EVENT	0 = Not a corporate event 1 = A corporate event	Integer	1	N
PROMOTION_START_REF	RETAIL = Pricing event takes effect against the original retail price CURRENT = Pricing event takes effect against the current ticket price	String	15	N
IS MOCKABLE	0 = User cannot create the pricing event via the UI 1 = User can create the pricing event via the UI	Integer	1	N
DESCRIPTION	Description of the event type	String	255	N

Client Hierarchy Actions Standard Interface

A one-time data feed that provides a formal method for deriving the level at which application-specific operations occur.

Client Hierarchy Actions Specification (CLIENT_HIERARCHY_ACTIONS_TBL)

Table 5–2 Client Hierarchy Actions Standard Interface Specification

Attribute	Attribute Description	Data Type	Maximum Length	Nullable Y/N
ACTION_TYPE	Group of action names	String	25	N
ACTION_NAME	Application action name (e.g., ASSORTMENT_PLAN, ITEM_PLAN1)	String	25	Y
ACTION_LEVEL	Level of operation (e.g., Plan levels are 1 and 2, which map to style and color. 0 indicates no level)	Integer	2	N
HIER_TYPE	Type of hierarchy (Location, Merchandise, Period, or Pack)	String	32	N
HIER_LEVEL_SQC	A number value of the level	Integer	2	N
APP_NAME	Which application uses the level (e.g., Plan, Price)	String	15	N
DESCRIPTION	Action description	String	255	N

Cross Products Information Standard Interface

Items are globally defined to be at a specific level of the merchandise hierarchy and the location hierarchy through the cross products interface.

For Plan, this is needed only for legacy reasons – and should only have three Price-specific entries with CHAIN/CHAIN level settings.

Cross Products Information Specification (ASH_CP_TBL)

Table 5–3 Cross Products Standard Interface Specification

Attribute	Attribute Description	Data Type	Maximum Length	Nullable Y/N
INTERSECT_NAME	The key name (OPTIMIZATION, SALES, or WORKSHEET)	String	50	N
MERCHANDISE_LEVEL	The defining level within the hierarchy	String	50	N
LOCATION_LEVEL	The defining level within the hierarchy	String	50	N

Location Hierarchy Levels Standard Interface

The location hierarchy levels interface is used to specify the names of a retailer's location levels and their order.

Technical Notes

The following list provides details to consider regarding the lh levels data.

- The Chain level should always be defined as 1.
- The sequence of level numbers must begin with 1 and increase in increments of 1, without any gaps in the sequence.
- The location hierarchy levels information is generally loaded only once.

LH Levels Specification (ASH_LHL_TBL)

Table 5–4 Location Hierarchy Levels Standard Interface Specification

Field Name	Field Description	Data Type	Maximum Length	Nullable Y/N
LOCATION_LEVEL	The name of the location level	String	50	N
LEVEL_SQC	The sequence number of the level	Integer	2	N

Merchandise Hierarchy Levels Standard Interface

The merchandise hierarchy levels interface is used to specify the names of a retailer's merchandise levels and their order.

Technical Notes

The following list provides details to consider regarding the mh levels data.

- The Chain level should always be defined as 1.
- The sequence of level numbers must begin with 1 and increase in increments of 1, without any gaps in the sequence.
- The merchandise hierarchy levels information is generally loaded only once.

MH Levels Specification (ASH_MHL_TBL)

Table 5–5 Merchandise Hierarchy Levels Standard Interface Specification

Field Name	Field Description	Data Type	Maximum Length	Nullable Y/N
MERCHANDISE_LEVEL	The name of the merchandise level	String	50	N
LEVEL_SQC	The sequence number of the merchandise level	Integer	2	N

Pack Hierarchy Levels Standard Interface

The pack hierarchy levels interface is used to specify the names of a retailer's pack levels and their order.

Technical Notes

The following list provides details to consider regarding the ph levels data.

- The Chain level should always be defined as 1.
- The sequence of level numbers must begin with 1 and increase in increments of 1, without any gaps in the sequence.
- The pack hierarchy levels information is generally loaded only once.

PH Levels Specification (ASH_PHL_TBL)

Table 5–6 Pack Hierarchy Levels Standard Interface Specification

Attribute Name	Attribute Description	Data Type	Maximum Length	Nullable Y/N
PACK_LEVEL	The name of the pack level	String	50	N
LEVEL_SEQ	The sequence number of the level	Integer	2	N

Optimized History

The Optimized History data feed is provided by Analytical Services. It is not an incremental feed, but instead completely replaces all existing optimized history data. The load is not part of the standard load. The load procedure for optimized history is `com.profitlogic.db.dogwood.LoadOptHistory`.

Table 5–7 Optimized History Standard Interface Specification

Attribute Name	Attribute Description	Data Type	Maximum Length	Nullable Y/N
MERCHANDISE_KEY	Unique identifier for merchandise hierarchy	String	25	N
MERCHANDISE_LEVEL	Level within the merchandise hierarchy	String	50	N
LOCATION_KEY	Unique identifier for location hierarchy	String	25	N
LOCATION_LEVEL	Level within the location hierarchy	String	50	N
FISCAL_YR	Fiscal year	Integer	4	N
FISCAL_WK	Fiscal week (1 - 53)	Integer	2	N
SALES_UNITS	Sales units	Number	8 decimal places	N
SALES_AMOUNT	Sales amount	Decimal	22,3	N
TICKET_PRICE	Ticket price	Decimal	22,3	N

Table 5–7 (Cont.) Optimized History Standard Interface Specification

Attribute Name	Attribute Description	Data Type	Maximum Length	Nullable Y/N
INVENTORY_UNITS	Inventory units	Number	8 decimal places	N
RECEIPT_UNITS	Receipt units	Number	8 decimal places	N
UNIT_COST	Unit cost	Decimal	22,3	N

Inference Rules

Inference rules define queries specifying particular views into the database that provide customization points for Plan.

Plan provides the following inference rules:

IR_AS_VERSIONS

The IR_AS_VERSIONS inference rule queries AS_VERSIONS_TBL for the version number of the Analytical Services feeds for Size Profile, Prepack, and Location Size Distribution (LSD). The version number of each feed is updated by the generator for each feed.

Customization: The format of the version number in the IR_AS_VERSIONS inference rule can be customized.

IR_PROMO_ATTR_ATTRIBUTE

The IR_PROMO_ATTR_ATTRIBUTE inference rule obtains the relative lift index for merchandise, location, and historic promotions.

This inference rule has the following columns:

- Merchandise_ID - Merchandise ID
- Location_ID - Location ID
- Promo_Start_Dt - Beginning of promotion
- Promo_End_Dt - End of promotion
- Relative_Lift_Idx - a lift is an indication of the effect of an external event, such as advertising, on sales when a promotion is in effect. Used in forecasting

Customization: The IR_PROMO_ATTR_ATTRIBUTE inference rule can be customized in terms of which tables the attribute fields and the location level are obtained from.

IR_PROMO_CORP_ATTRIBUTE

The IR_PROMO_CORP_ATTRIBUTE inference rule obtains the base (corporate) promotion attribute value for each merchandise location combination.

This inference rule has the following columns:

- Merchandise_ID - Merchandise ID
- Location_ID - Location ID
- Attribute_Mask - Used for constructing attribute values

Customization: By default, the IR_PROMO_CORP_ATTRIBUTES returns % for each record, so this inference rule must be configured. The customization is driven by which attributes Analytical Services uses for escalation when generating the base historic promotion lift values.

IR_DEMAND_PARAM_ATTRIBUTE

The IR_DEMAND_PARAM_ATTRIBUTE inference rule obtains the demand parameter attribute value for each merchandise location combination.

This inference rule has the following columns:

- Merchandise_ID - Merchandise ID
- Location_ID - Location ID
- Attribute_Mask - Used for constructing attribute values

Customization: By default, the IR_DEMAND_PARAM_ATTRIBUTE returns % for each record, so this inference rule must be configured. The customization is driven by which attributes Analytical Services uses for escalation when generating the demand parameter values.

IR_PLAN_PERIOD_AGGR_TOTALS_VW

The IR_PLAN_PERIOD_AGGR_TOTALS_VW inference rule is used for the aggregation of each merchandise item in each aggregation period for each plan period across all stores.

Customization: This inference rule is not currently configurable.

IR_PLAN_PERIOD_STORE_INDX_VW

The IR_PLAN_PERIOD_STORE_INDX_VW inference rule calculates the store index by merchandise, aggregation period, and plan period.

Customization: This inference rule is not currently configurable.

IR_SIZE_PROFILE_ATTRIBUTE

The IR_SIZE_PROFILE_ATTRIBUTE inference rule calculates the size profile attribute for each merchandise location combination. The level of the merchandise hierarchy is obtained from ASH_CP_TBL and the location hierarchy level is always the lowest.

This inference rule has the following columns:

- Merchandise_ID - Merchandise ID
- Merchandise_Level_Desc - Merchandise level description
- Location_ID - Location ID
- Location_Level_Desc - Location level description
- AS_Merchandise_ID - AS Merchandise ID
- AS_Merchandise_Level_Desc - AS Merchandise level description
- AS_Location_ID - AS Location ID
- AS_Location_Level_Desc - AS Location level description
- Attribute_Value - the actual value for the attribute

Customization: The construction of the size profile attribute value is customizable. By default, it is size_range_key || ':' || size_range_size_cnt and is driven by Analytical Services.

IR_PREPACK_ATTRIBUTE

The IR_PREPACK_ATTRIBUTE inference rule calculates the prepack attribute for a item of merchandise at a specific location. The level of the merchandise hierarchy is third from the bottom. This view is used by the Prepack generator.

This inference rule has the following columns:

- Merchandise_ID - Merchandise ID
- Merchandise_Level_Desc - Merchandise level description
- AS_Merchandise_ID - As Merchandise ID
- AS_Merchandise_Level_Desc - AS merchandise level description
- Attribute_Value - Used for constructing attribute values

Customization: The construction of the prepack attribute value is customizable. By default, it is `r.size_range_key || ':' || r.size_range_size_cnt`. It is driven by Analytical Services.

IR_GET_PROMO_ATTR

The IR_GET_PROMO_ATTR inference rule constructs a promotion attribute value from a list of promotion attributes. This value is used against the attribute mask provided by Analytical Services.

Customization: The attribute name and the order of concatenating values for each attribute can be customized.

IR_GET_SEASONALITY_VALUES

The IR_GET_SEASONALITY_VALUES inference rule constructs a seasonality attribute value from a list of seasonality attributes. This value is used against the attribute mask provided by Analytical Services.

Customization: The construction of the attribute value from the input name value pair parameter is customizable.

IR_GET_EXT_STORE_WEIGHTS

The IR_GET_EXT_STORE_WEIGHTS inference rule calculates the external store weight value that is used by the Calc Engine. The calculation uses input parameters for the level specified and the months that fall between the specified start date and end date.

Customization: The internal logic for the calculation of the store weight is customizable.

Configurable Data Attributes

This chapter contains the following:

- “Introduction” on page 1
- “Defining Configurable Data Attributes” on page 1

Introduction

Configurable Data Attributes (CDAs) provide a way for retailers to see, in addition to the default data that is visible through the application interface, custom data that they themselves specify and that is not required by the application. This data can be used in business rules and can be displayed in the application UI.

Defining Configurable Data Attributes

Configurable Data Attributes are defined in the database using the CDA Administration Utility. The data is then staged and loaded. All client-specified data is included in the standard interface specification in fields with field names beginning with the word ATTRIBUTE.

Note: CDAs are disabled by default. The column PL_DD_ATTRIBUTES.DISABLED should be set to 1 to disable the CDA and should be set to 0 to enable the CDA.

You can access the CDAs in the database via database queries or change the grid configuration to make them visible in the user interface.

The number of CDAs per entity is limited by the number of database columns pre-allocated in every CDA storage table. Every application schema provides eight data columns of type VARCHAR and DATE, and ten number columns of type NUMBER. When you are creating a new attribute, you can choose the storage columns from the following disassociated columns of the corresponding type:

Table 6–1 CDA Data Type

Data Attribute Type	Data Type
String	VARCHAR
Integer	NUMBER
Boolean	NUMBER
Double	NUMBER
Date	NUMBER
Currency	VARCHAR

Table 6–1 (Cont.) CDA Data Type

Data Attribute Type	Data Type
Currency	NUMBER (2 columns)

The following tables supports extension by the CDA Administrative Utility:

Table 6–2 Standard Interface Tables with CDAs

Entity Name	Staging Table	Active Table	CDA Table
Location	ASH_LH_TBL	LOCATION_HIERARCHY_TBL	LH_CDA_TBL
Merchandise	ASH_MH_TBL	MERCHANDISE_HIERARCHY_TBL	MH_CDA_TBL

Merchant Desktop and Retail Data Mart Configuration

The chapter contains the following:

- “Introduction” on page 1
- “Merchant Desktop Properties Settings” on page 1
- “Merchant Desktop and CDAs” on page 2
- “Starting and Stopping Components” on page 3
- “RDM Data Loads” on page 3
- “RDM Refresh” on page 5
- “Microstrategy Configuration” on page 6
- “Performance Tuning” on page 7
- “Report Metrics” on page 8

Introduction

The Merchant Desktop (MD) is an in-season management tool that provides single sign-on as well as tracking, monitoring, and reporting functionality.

The Retail Data Mart (RDM) is a set of database objects, existing in a separate schema, that provides information from the application that can be used for analysis by Business Intelligence tools, such as Microstrategy.

This chapter provides details about configuration settings for Merchant Desktop, Retail Data Mart, and Microstrategy.

Merchant Desktop Properties Settings

After installation, several configuration properties must be set. Note that the first three properties will be set during installation if `install.properties` is configured correctly.

User Management Properties. The following User Management properties, located in `<Installation Directory>/config/usermanagement/usermanagement.properties`, must be assigned values:

`rmiHost =`

`rmiPort =`

`reportServer =`

```

administratorName =
administratorPassword =
mdProject =
merchandise.chain.level.name =
location.chain.level.name =
microstrategy.users.max =

```

Merchant Desktop Properties. The following Merchant Desktop properties, located in <Install Directory>/config/merchantdesktop/merchantdesktop.properties, must be assigned values:

```

microstrategy.server =
microstrategy.project =

```

Merchant Desktop ear Files. Changes must be made to the following files:

Rename the file, <Install Directory>/modules/MerchantDesktop.ear/WebMstr7.war/WEB-INF/xml/sys_defaults_192168.0.157.xml by replacing the IP address in the file name with the IP address of the Microstrategy server.

Update the IP address in the file MerchantDesktop.ear/WebMstr7.war/WEB-INF/xml/AdminServers.xml by replacing the IP address in the file name with the IP address of the Microstrategy server.

Note: In both cases use the IP address, not the DNS name.

Date Format. The SQL date format must be set to YYYY/MM/DD during the initial configuration of the RDM Data Warehouse. The setting is available in VLDB properties in the Select/Insert section.

Merchant Desktop and CDAs

For Merchant Desktop to use configurable data attributes, the following two changes must be made:

- Set the isdisabled field to 0 so that the rows to be used are enabled in the pl_dd_attributes table. The change must occur before the merchandise hierarchy and the location hierarchy are loaded.
- Update the attributes entries in the Microstrategy project metadata with the name of the attribute to be stored in the cda and the levels of the hierarchy where the attribute will be populated. The attributes are located under the Schema Objects/Attributes folder. Once the changes have been made, the schema must be updated.

Note that the same attribute should be represented at more than one level, because reports that use the attribute can only be run at the levels at which the attribute is configured. If an attribute is only configured at low levels, then performance can be adversely impacted. If an attribute is only configured at high levels, then users will not be able to drill down into the data.

Starting and Stopping Components

This section provides general information about starting and stopping the Plan servers. For greater detail, consult the server documentation.

Starting and Stopping the Application Servers

Since the application servers are installed as a cluster, it may be necessary to log into several application servers in order to start or stop the application. Alternatively, you can use the tools available with the application server.

Starting and Stopping the Microstrategy Server

The Microstrategy server is installed as a Windows service. You can start and stop the Microstrategy server or query its status via the following:

- Windows Services Control Panel
- Windows command line, using the netstart and netstop commands
- Microstrategy desktop
- Microstrategy Service Manager

Starting and Stopping the RMI Server

The RMI server is installed as a Windows service. You can start and stop the RMI server or query its status via the following:

- Windows Services Control Panel
- Windows command line, using the netstart and netstop commands

Monitoring Merchant Desktop

You can monitor the Merchant Desktop via `http://<servername>:<port>/MerchantDesktop/servlet/monitor_target`. The web page displays “OK” if the Database, Merchant Desktop, and User Management components are reachable and working from the server. The web page displays “BAD” if this is not the case. The monitoring url can be use with automated tools or as is.

RDM Data Loads

Before the RDM data is first loaded, you must configure the following. Once a week of data is loaded, these settings cannot be changed again, as it would leave the application in an inconsistent state. These configuration setting changes (from the product default settings) must be matched by corresponding changes in the Microstrategy metadata that is used for reporting.

The Microstrategy metadata must also be updated so that the reports can reflect data aggregations. To do this, edit the location and merchandise attributes and make sure that they are associated with the tables at the proper levels. For example, if the third level of optimized history aggregation is the Division level of the location hierarchy, then `rdm_opt_history_2` must be the source table for the id form of the location division attribute. In addition, the table must not be mapped to the attributes for the other levels of the location hierarchy. The merchandise hierarchy should be similarly edited.

Aggregation Levels Configuration

Data aggregation during the load process is supported for the load of the actual history and the optimized history. The aggregation is necessary for acceptable performance in the reports. Three levels of aggregation are supported. Each level of aggregation created must be at least as high a level as the one created before it. The first level of aggregation for the Actual History must be Store/Color.

Actual History Load in the DB

The aggregation levels, which specify the levels of aggregation for the weekly sales and inventory data, are configured through the CLIENT_HIERARCHY_ACTIONS_TBL.

Table 7–1 Aggregation Levels in CLIENT_HIERARCHY_ACTIONS_TBL

ACTION_TYPE	ACTION_NAME	ACTION_LEVEL	HIER_TYPE	HIER_LEVEL_SQC	APP_NAME	DESCRIPTION
HIST_AGG_LEVEL_0	HIST_AGG_MERCH_LEVEL_0	0	MERCHAN DISE	10	Plan	Lowest MH level at which History should be persisted
HIST_AGG_LEVEL_1	HIST_AGG_MERCH_LEVEL_1	0	MERCHAN DISE	9	Plan	MH level at which 1st aggregation should be prepared
HIST_AGG_LEVEL_2	HIST_AGG_MERCH_LEVEL_2	0	MERCHAN DISE	9	Plan	MH level at which 2nd aggregation should be prepared
HIST_AGG_LEVEL_3	HIST_AGG_MERCH_LEVEL_3	0	MERCHAN DISE	5	Plan	MH level at which 3rd aggregation should be prepared
HIST_AGG_LEVEL_0	HIST_AGG_LOC_LEVEL_0	0	LOCATION	5	Plan	Lowest LH level at which History should be persisted
HIST_AGG_LEVEL_1	HIST_AGG_LOC_LEVEL_1	0	LOCATION	5	Plan	LH level at which 1st aggregation should be prepared
HIST_AGG_LEVEL_2	HIST_AGG_LOC_LEVEL_2	0	LOCATION	2	Plan	LH level at which 2nd aggregation should be prepared
HIST_AGG_LEVEL_3	HIST_AGG_LOC_LEVEL_3	0	LOCATION	1	Plan	LH level at which 3rd aggregation should be prepared

Optimized History in RDM

The aggregation levels are configured using the `optHistory.properties` file. Enter the location hierarchy level and merchandise hierarchy level at which the data should be aggregated. The properties file is loaded by `LoadDBProperties.sh`.

- `OPTHIST_MERCH_LEVEL0=SKU`
- `OPTHIST_MERCH_LEVEL1=COLOR`
- `OPTHIST_MERCH_LEVEL2=COLOR`
- `OPTHIST_MERCH_LEVEL3=SUB_DEPARTMENT`
- `OPTHIST_LOC_LEVEL0=STORE`
- `OPTHIST_LOC_LEVEL1=STORE`
- `OPTHIST_LOC_LEVEL2=DIVISION`
- `OPTHIST_LOC_LEVEL3=DIVISION`

The input data has only one level setting for all merchandise and location levels in the chain and is set as follows:

- `OPTHIST_MERCH_DATA_LEVEL=SKU`
- `OPTHIST_LOC_DATA_LEVEL=STORE`

The level of the Merchandise Hierarchy at which Analytical Services can restate the optimized history, which is set only once and cannot be changed, is as follows:

- `OPTHIST_FILE_GROUP_LEVEL=DIVISION`

RDM Data Load

The `pl_load_redwood.sh` script loads the Retail Data Mart schema. This is a one-time data load.

This script includes the following procedures:

- `com.profitlogic.db.redwood.LoadPeriods`
- `com.profitlogic.db.redwood.LoadLHData`
- `com.profitlogic.db.redwood.LoadMHData`
- `com.profitlogic.db.redwood.LoadLTClose`
- `com.profitlogic.db.redwood.LoadTClose`
- `com.profitlogic.db.redwood.LoadMCda`
- `com.profitlogic.db.redwood.LoadLCda`

RDM Refresh

Since the Retail Data Mart has its own schema, it needs to be refreshed from the Plan schema. The Plan Updater refreshes the RDM as part of the nightly batch process. Plan also provides a way via the Plan Library screen for a user to update a plan during the day in order to see the impact of changes in an RDM report. It refreshes planned item data, segment data, DC segment data, pack optimization, and company / store budgets data.

Refresh Procedures

The following procedures should be run every night after the data load. The RDM Refresh procedure is run from the Plan Updater or from the Plan application. The other procedures should be part of the nightly automation.

Table 7–2 RDM Refresh Procedures

Refresh Procedure	Description
RDM Refresh. Part of Plan Updater or from the Plan UI.	Refreshes Planned_Item_Data, Segment data, DC Segment data, and Pack Optimization data.
com.profitlogic.db.redwood.LoadMHData	Refreshes Merchandise Hierarchy data.
com.profitlogic.db.redwood.LoadLHData	Refreshes Location Hierarchy data.
com.profitlogic.db.redwood.LoadMCda	Refreshes Merchandise Hierarchy CDA data.
com.profitlogic.db.redwood.LoadLCda	Refreshes Location Hierarchy CDA data.
com.profitlogic.db.teak.LoadCompanyBudgets	Refreshes Company Budget data.
com.profitlogic.db.teak.LoadStoreBudgets	Refreshes Store Budget data.
com.profitlogic.db.teak.LoadDistCenters	Refreshes Distribution Center data.
com.profitlogic.db.teak.LoadVolumeGroups	Refreshes Volume Group data.
com.profitlogic.db.teak.LoadStores	Refreshes Storeset data.

Microstrategy Configuration

This section provides details on some key Microstrategy Configuration settings.

Microstrategy Web Preferences

Each Microstrategy project has a set of default web preferences. The following parameters may require modification. Since the web preferences are defined on a per project basis, any changes to these parameters only apply to the given project. All changes are made through the Project Defaults section of Preferences in the Merchant Desktop and can only be made by someone with System Administrator privileges.

- Export to PDF should be disabled as it is not supported in the Merchant Desktop.
- The # of Cells that can be exported to Excel should be increased as the default is low.
- The # of Elements that can be displayed in a prompt window can be increased from the default of 30. Change the Number of Attribute Elements per block parameter, which is found under Incremental Fetch.

Upgrading Microstrategy Metadata

One method of upgrading the Microstrategy metadata involves duplicating the project initially created on a staging server to the production server. The project on the

production server should be named so that it is not necessary to change the usermanagement and merchantdesktop properties.

When the changes to the metadata are limited, it is better to use the Microstrategy Object Manager to update the production metadata. Using the Object Manager to duplicate the project to the production environment ensure that security filters for existing users and user-created reports are not deleted.

Filters cannot be migrated to a production environment unless they have been explicitly changed. If they are migrated, they must be refreshed.

Facts/Aggregates

RDM supports four levels of aggregation. If these levels change, then the mapping in Microstrategy must also change. To change the levels at which facts/aggregates are available:

1. From the Microstrategy Desktop, select Project > Schema Objects > Attributes > Product Attributes.
2. Double-click Sub-Department - the level at which aggregates are available by default.
3. From the Sub-Department Attribute Editor, select PI_ID and click Modify.
4. From the Sub-Department (ID) - Modify Attribute Form, select the levels at which facts are no longer being maintained. To remove them, un-check them and click OK.
5. From Project > Schema Objects > Attributes > Product Attributes, double-click Department (the new level).
6. From the Department Attribute Editor, select PI_ID and click Modify.
7. From the Department (ID) - Modify Attribute Form, select the levels at which facts are to be maintained and click OK.
8. Repeat this process for each fact that needs to be modified. Also repeat for the Location Attribute.
9. Update the Microstrategy Schema and purge the reports from the cache before re-running reports using the newly defined aggregates.

Merchandise Hierarchy and Location Hierarchy Level Names

The names for the merchandise hierarchy and location hierarchy levels must match the names that are visible through the Microstrategy project. If they do not, then the user management security filters will not be created properly and users may not have correct access to data.

To update the names of the levels, edit the attributes individually. The names must match the names in the database (not case sensitive though). The location levels are located in Schema Objects/Attributes/Location Attributes. the merchandise levels are located in Schema Objects/Attributes/Merchandise Attributes.

Performance Tuning

This section provides general guidelines about performance tuning. Further details about performance tuning the components are best obtained from the component product documentation.

Microstrategy Server

The following Microstrategy parameters may require tuning:

- Caching - Maximum RAM usage. The default setting is 25600 Kbytes. This setting may need to be increased if large reports need to be cached. The setting is found in MicroStrategy Desktop > Project Configuration > Caching > Reports (General) > Maximum RAM Usage. If the data is volatile, it may be necessary to disable caching at the Project or the Report level in order to ensure that reports have the most current data.
- Job Prioritization. This controls the number of simultaneous database connections to the data warehouse that are allowed. This setting should be high enough that users do not experience delays in accessing the database and low enough that the database is not overloaded. A setting of 30 Low Priority Connections is recommended in order to support of 50 -75 users.
- Web User Session Idle Timeout. The default setting is 600 seconds. A higher setting may be need to prevent timeouts during report runs. The setting is found in MicroStrategy Desktop > Administration > Server > Configure MicroStrategy Intelligence Server > Governing > Modify Web User Session Idle Time. This setting should match the application (suite) session timeout setting in suite.properties.
- Max Report Execution Time. It is recommended that this setting match the Web User Session Idle Timeout. The setting is found in MicroStrategy Desktop > Project Configuration > Governing > Results Set > Report Execution Time.

Reports

Here are some performance tips regarding reports:

- The reports provided with Plan have been tuned with the assumption that most attributes are available at the Color level. Attributes that are available only at the SKU level cause the report to ignore all the aggregations. This can increase the load on the server.
- If the data is not updated regularly, the cache timeout period can be set to a long time period. As a result, the cache should be flushed before data is re-loaded.
- The basic reports have been hinted to improve their performance through the use of the star join transformation.
- A separate set of reports exists that can be used when a report must be run at the SKU level across a large portion of the division. These reports are hinted to improve their performance. However, they should not be used at higher levels as they use more resources.

Report Metrics

Table 7–3 *Report Metrics*

Metric	Calculation
# of Customer Choice	Count (Color)
# of Styles	Count (Style)
Receipt Unit	Sum (Receipt Units)
Receipt Units for IBT	RunningSum (Receipt Units) <SortBy=Week Period ID>
Receipt \$	Receipt U * Init Retail

Table 7-3 (Cont.) Report Metrics

Metric	Calculation
Receipt \$ for IBT	RunningSum (Receipt \$)
% Receipt Unit	Receipt Unit / Total Receipt Units
% Receipt \$	Receipt Amt / Total Receipt Amt
Receipt Unit (Cum.)	RunningSum (Receipt Unit To Date)
Receipt \$ (Cum.)	RunningSum (Receipt Amt)
Receipt U % Diff Need to AP	$((\text{Need Receipt Amt} / \text{AP Receipt Amt}) - 1) * 100$
Receipt \$ % Diff Need to AP	$((\text{Need Receipt Amt} / \text{AP Receipt Amt}) - 1) * 100$
Sales U	Sales Units
Sales \$	Sales Amt
% Sales U	Sales U / Total Sales U
% Sales \$	Sales \$ / Total Sales \$
Sales U (Cum.) for IBT	RunningSum ([Sls Units ToDate])
Sales \$ (Cum.) for IBT	RunningSum ([Sls \$ ToDate])
Sales U % Diff Need to AP	$((\text{Need Sale U} / \text{AP Sale U}) - 1) * 100$
Sale \$ % Diff Need to AP	$((\text{Need Sale $} / \text{AP Sale U}) - 1) * 100$
Initial Retail	item_setup_retail
Current Retail	curr_perm_price
ASR (AUR)	Sale \$ / Sales U
AUC	item_setup_cost
IMU	$(\text{initial_retail} - \text{auc}) / \text{initial_retail}$
POS Sales U	POS Sales Units
POS Sale \$	POS Sales Amt
MD Sales U	MKDN Sales U
MD Sales \$	MKDN Sales \$
Reg Sales U	Sales U - POS Sales U - MD Sales U
Reg Sales \$	Sales \$ - POS Sales \$ - MD Sales \$
Perm MD \$	Perm MD \$
POS MD \$	POS MD \$
Total MD \$	POS MD \$ + Perm MD \$
GP%	$\text{IMU} - (\text{Weekly Total MD \%} + \text{Shrink \%}) * (1 - \text{IMU})$
ADJ IMU	If Domestic (Retail - Cost * 0.86) / Retail Else IMU
ADJ GP%	$((\text{ADJ_IMU} * 100) - (\text{Weekly Total MD \%} + \text{Shrink \%})) * (1 - \text{ADJ} - \text{IMU}))$
GP\$	GP % * SLS AMT
GPS \$ % Diff AP TO NEED	$(\text{NEED GP \$} / \text{AP GP \$}) - 1) * 100$
BOH U	BOH U
BOH \$	BOH U * Current Retail

Table 7–3 (Cont.) Report Metrics

Metric	Calculation
EOH U	EOH U
EOH \$	EOH U * Current Retail
% ST U	Sales U (Cum.) / Receipt U (Cum.)
NORM CHK	$1 - (((EOH) / (Receipt\ U\ (Cum.)) * (1 / \# \text{ of Weeks}))) * 100$
FRESH STK	Sum 1...6 [Week - n Receipt U] / Current EOH
Store w/ Inv	# of Stores w/ BOH > 0
Store w/ Rec	# of Stores w/ Receipt U > 0
First Receipt Dt	First Receipt Date
# of Selling Weeks	Num_of_Selling_Wks
Min Out Date	Min (Out_Date)
Max Out Date	Max (Out_Date)
WOS	EOH / SALES_UNITS
POS Sales \$ % To Total	POS Sales \$ / Sales \$
MD Sales \$ % To Total	MD Sales \$ / Sales \$
Sales \$ % To Budget	Sales \$ / Dept Sales Budget \$
Sales Budget \$	Dept Sales Budget \$
Sales Budget \$ % Diff	$((Dept\ Sales\ \$ / Dept\ Sales\ Budgets\ \$) - 1) * 100$
Receipt \$ % to Budget	Receipt \$ / Dept Receipt Budget \$
Receipt Budget \$	Dept Receipt Budget \$
Receipt Budget \$ % Diff	$((Dept\ Receipt\ \$ / Dept\ Receipt\ Budget\ \$) - 1) * 100$
Budgeted IMU	Dept IMU Budget
Budgeted GP \$ (Gross Profit \$)	GP \$
Budgeted GP % (Gross Profit Pct)	GP %
Budgeted GP ROI	GPROI
Budgeted Perm MD \$	P2
Budgeted POS MD \$	P4
Item Status	Item_period_segments_tbl.pricing_status
POS BOH U	If Item Status = PR Then BOH U Else 0
POS BOH \$	If Item Status = PR Then BOH U * CurrPermPrice Else 0
POS EOH U	If Item Status = PR Then EOH U Else 0
POS EOH \$	If Item Status = PR Then EOH \$ Else 0
MD BOH U	If Item Status = MD Then BOH U Else 0
MD BOH \$	If Item Status = MD Then BOH \$ Else 0
MD EOH U	If Item Status = MD Then EOH U Else 0
MD EOH \$	If Item Status = MD Then EOH \$ Else 0
REG EOH U	EOH - POS EOH - MD EOH
REG BOH \$	BOH \$ - POS BOH \$ - MD BOH \$

Table 7–3 (Cont.) Report Metrics

Metric	Calculation
REG EOH \$	EOH \$ - POS EOH \$ - MD EOH \$
DC Receipt Units	DC Rcpt Units
DC Receipt \$	DC Rcpt Amt
Pack Desc	Pack desc
Number of Packs	Num_packs
Units Per Pack	Num_unit_per_pack
Total Rcpt Units	Num_packs * num_unit_per_pack
Total Retail	Total Rcpt Units * retail
Total Cost	Total Rcpt Units * cost
Item Size	Size
Quantities By Size	Inside units

Plan Updater

The chapter contains the following:

- “Introduction” on page 1
- “About Plan Updater” on page 1
- “Using Plan Updater” on page 2
- “Environment Variables” on page 2
- “plan.properties File” on page 2
- “planupdater.properties” on page 2
- “Plan Updater Script” on page 3

Introduction

The Plan Updater updates existing plans based on data changes that are not the result of user interaction with the Plan application. In effect, the Plan Updater makes sure that plan data and item data are up-to-date.

Data changes that the Plan Updater recognizes are the result of such activities as store shifts, pricing plan changes, FNN tasks, certain data feeds, the re-forecasting of items, and the re-calculation of item segments.

About Plan Updater

The Plan Updater runs as a batch process overnight. The batch process is controlled by the updater script, whose arguments can be used to configure the process.

When a plan is created, the setting in `plan.plan.library.planupdater.default` determines the default value (selected or not) of the check box in the Plan Library screen. This setting in the Plan Library is under the control of the end user, who can change the setting and so ultimately determine whether or not a particular plan subscribes to the Plan Updater.

Plan provides updates to RDM on a per plan basis. After the data in a plan is successfully updated, it is then migrated to RDM. The RDM update is the last update performed by the Plan Updater, if the Plan Updater has been configured to run the RDM refresh process.

The nightly updater process must occur after the nightly data load.

Using Plan Updater

This section provides technical details about using Plan Updater.

Environment Variables

The following environment variables must be set:

Table 8–1 UNIX Environment Variables for Plan Updater

Variable Name	Example
WEBLOGIC_HOME	export WEBLOGIC_HOME=/s000/bea/weblogic81/server
PLANUPDATER_HOME	export PLANUPDATER_HOME=/s000/install/PlanUpdater

Table 8–2 Windows Environment Variables for Plan Updater

Variable Name	Example
WEBLOGIC_HOME	set WEBLOGIC_HOME=D:\bea\weblogic81\server
PLANUPDATER_HOME	set PLANUPDATER_HOME=D:\install\PlanUpdater

plan.properties File

The `plan.plan.library.planupdater.default`, located in `plan.properties` file, determines the default setting for the Subscription flag. If this flag is set to 1, then a plan by default subscribes to updates. If this flag is set to 0, then a plan by default does not subscribe to updates. The setting in the Plan UI overrides this default setting.

For more information on the `plan.properties` files, see [Chapter 9, "Properties File Configuration"](#).

planupdater.properties

The following properties, found in `planupdater.properties` in the Plan Updater `conf` directory, can be configured:

Table 8–3 planupdater.properties

Property Name	Property Description	Property Value
<code>planupdater.maxthreads</code>	The maximum number of threads to process plan and item tasks.	Value: Integer Default = 10
<code>planupdater.planmaxseconds</code>	The maximum number of seconds to wait for all plan tasks for all selected plans to complete.	Value: Integer Default = 300
<code>planupdater.itemmaxseconds</code>	The maximum number of seconds to wait for all item tasks for all selected plans to complete.	Value: Integer Default = 120
<code>planupdater.testmode</code>	If set to true, then the log only records that a method was to be called for a particular plan or item. The plan or item is not updated.	Value: true or false Default = false
<code>planupdater.verbose</code>	Sets the verbosity level for log file output messages. If set to true, detailed messages are logged. If set to false, then default logging is provided.	Value: true or false Default = true

Plan Updater Script

The Plan Updater script is located in <install directory>/modules/tools/bin folder.

Operating System	Script
UNIX	planupdater.sh
Windows	planupdater.bat

The following command line parameters are available:

Table 8–4 Plan Updater Command Line Parameters

Parameter	Description
-operations	<p>Specify a list of tasks to be performed by Plan Updater. The operations parameter accepts the following task arguments. If none are listed, then all are assumed.</p> <p>planStore - Shifts stores into new store grades. New store grades are created from PLAN_STORE_GRADE_DELTAS. New stores are added from PLAN_LOCATION_DELTAS. Segments and forecasts are labelled "dirty" for non-submitted items. Budget changes are processed from PLAN_BUDGET_DELTAS. AP Receipt units/MIN/MAPQ for all items on LOCATION_PLANNED_ITEMS_TBL are updated. PLAN_LOCATIONS_TBL is updated for all items (which cleans out PLAN_LOCATION_DELTAS).</p> <p>planPrice - Checks for Pricing Plan changes that affect Plan.</p> <p>planAuto - Auto-populates a plan with basic items.</p> <p>planRdm - Sends a request to update RDM. This is an asynchronous call that does not prevent Plan from completing its task.</p> <p>itemForecast - Re-forecasts items in a plan if the calc params-changed flag for the related item set ID is marked "dirty." The item set can span multiple plans. (The Plan Library Plan Updater check box indicates whether or not an item set is included in the Plan Updater process.)</p> <p>itemSegment - Updates all segments for a particular item set ID.</p> <p>These tasks are executed in the following order:</p> <ol style="list-style-type: none"> 1. planPrice 2. planAuto 3. planStore 4. itemForecast 5. itemSegment 6. planRdm
-seize	Seize control of the plan if it is locked by another user.
-logbase	Define the log file directory for the current execution of Plan Updater.

Properties File Configuration

The chapter contains the following:

- “Introduction” on page 1
- “Calc Engine Properties” on page 2
- “Message Timeout Properties” on page 2
- “Auto-Population and FNN Properties” on page 3
- “Session Timeout Properties” on page 3
- “General Properties” on page 4
- “Client Plug-In Properties” on page 4
- “Application Plug-In Properties” on page 5
- “Background Event Subsystem Properties” on page 6
- “Forecast Properties” on page 7
- “Pack Optimization Properties” on page 8
- “Calc Parameters Properties” on page 8
- “Flow View Properties” on page 9
- “Login Properties” on page 9
- “Auxiliary URL Properties” on page 9
- “Cache Limit Sizes and Cache Management Properties” on page 10
- “Internal Developer Properties” on page 10
- “Grid Configuration Properties” on page 11
- “Metrics Key List” on page 11
- “Validation Properties” on page 12
- “Date Properties” on page 12

Introduction

This chapter contains a list of the properties found in the plan.properties file, located under <ConfigRoot>. this file can be configured directly. A copy of the file can be created and configured under a client subdirectory of <ConfigRoot>. This method is recommended. The properties defined in the properties file located in the client subdirectory override those found in the original file.

Note: If a property is commented out, it must be un-commented before the values can be modified.

Calc Engine Properties

This section describes the properties that are used to communicate with the Calc Engine.

Property Name	Property Description
messageframework.jndi.url	The url of the Calc Engine. Plan uses the url to locate the Calc Engine.
messageframework.context.factory	The JNDI context factory that is used to get at the JNDI name server hosting the JMS-administered objects for the Calc Engine.
messageframework.send.queue	The JMS Send Queue name for communicating with the Calc Engine.
messageframework.reply.queue	The JMS Reply Queue name for communicating with the Calc Engine.
messageframework.dump.messages	If this property is set to true, then the xml request/response messages sent to and from the Calc Engine are dumped into a file in the application server domain directory. The default value is false.
plan.force.basic.plc.for.forecast	If this property is set to true, it forces the Calc Engine to fit the forecast to Basic PLC regardless of the setting at the item level in the UI. If it is set to false, the forecast behavior is governed by the per-item PLC setting in the Like Item screen of the UI. The default value is false.
plan.force.basic.plc.for.types	This property defaults to a value of yes on the Like item screen Force Basic PLC fit option for the B purchase types. Multiple purchase types can be specified as a comma-separated list. If the value is empty, it defaults to no for all items.

Message Timeout Properties

This section describes the properties that define message timeouts. Timeouts are expressed in seconds.

Property Name	Property Description
plan.packopt.request.timeout	The number of seconds that Plan waits for an outstanding PO request. The default value is 3600 (1 hour).
plan.plansubmit.request.timeout	The number of seconds that Plan waits for a plan submit request. The default value is 300 (5 minutes).
plan.forecast.request.timeout	The number of seconds Plan waits for a forecast request. The default value is 1800 (30 minutes).
plan.current.forecast.timeout	The number of seconds Plan waits before the re-sending of a new forecast fails. The default value is 43200 (12 hours).

Auto-Population and FNN Properties

This section describes the auto-population properties and the FNN operations properties.

Property Name	Property Description
plan.suppress.auto.populate.fnn	A flag that dictates whether or not any FNN data (basic items or FNN forecast data) is imported (auto-populated). If this property is set to true, then plan.suppress.fnn.import and plan.enable.fnn.import.option are irrelevant.
plan.suppress.fnn.import	A flag that suppresses the FNN import phase of the auto-population process. If this property is set to true, then no FNN data is imported when the Plan is open. Instead, it is deferred until the Plan Updater next runs. The default value for this property is false. To prevent FNN data from being imported when the Plan is open, set the plan.suppress.fnn.segment.update property (described next) to true.
plan.suppress.fnn.segment.update	A flag that suppresses the segment data update that follows the FNN insert and update operations. The default value is false. If this property is set to true, then the segment is not updated, which can lead to improper results. Note: this property should only be used for development and testing.
plan.enable.fnn.import.option	Allows the user to determine whether the initial FNN data is imported in synchronous mode (in real time) or asynchronous mode (in the background). If the user chooses asynchronous mode, then the user can open the Plan in read-only mode or return to the Library page. The default value is true. If the property is set to false, then this feature is disabled. Note: If suppress.auto.populate is set to true, then this property has no effect.

Session Timeout Properties

This section describes the session timeout properties.

Property Name	Property Description
plan.enable.usersession.timer	Indicates how the user session expiration task is run. The default value is INTERVAL. Other values are OFF and NIGHTLY.
plan.statesvc.userSession.sweepHour	If plan.enable.usersession.timer is set to NIGHTLY, then this property specifies the time, using a 24-hour clock. The default value is 4.

Property Name	Property Description
plan.statesvc.userSession.sweepInterval	If plan.enable.usersession.timer is set to INTERVAL, then this property specifies the time interval for checking user sessions for expiration. The default value is 30. Note: This property value is set in Minutes.
plan.statesvc.userSession.expireInterval	The length of time a user session can survive without an active web session. The default value is 72 (3 days). Note: This property value is set in Days.

General Properties

This section describes the general application properties.

Property Name	Property Description
plan.item.number.length	The character length of an item (style), as specified by the client. Used to add leading zeros, as appropriate. The default value is 7.
plan.child.number.length	The character length of an item's child (color), as specified by the client. Used to add leading zeros, as appropriate. The default value is 2.
plan.storeset.all.id	The ID assigned to the ALL storeset. Note: Do Not Change.
plan.storeset.subset.all.id	The ID assigned to the ALL store subset. Note: Do Not Change.
plan.workflow.menu.file	Navigation/Workflow configuration file settings.
plan.receipt.budget.default	Receipt Budgets are reconciled against Store receipts or DC receipts. Valid values are Store or DC. Note: If the value is empty, then Store is assumed
plan.product.group.default.behavior	The default behavior of the product group (subclass). This value must correspond to the value returned by the BRM. If it does, then Plan's product group behavior is managed by the user. If it does not, then Plan's product group behavior is managed automatically, based on the BRM configuration and the underlying data.

Client Plug-In Properties

This section describes the client-specific plug-in classes.

Property Name	Property Description
plan.logic.assortmentplan.plugin.class Name	The Java class name of the client-specific AssortmentPlanPlugin module. This class must implement the com.profitlogic.buying4p.logic.plugins.AssortmentPlanPlugin interface.

Property Name	Property Description
plan.logic.planneditem.plugin.className	The Java class name of the client-specific PlannedItemPlugin module. This class must implement com.profitlogic.buying4p.logic.plugins.PlannedItemPlugin interface.
plan.logic.item.plugin.className	The Java class name of the client-specific ItemPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.ItemPlugin interface.
plan.logic.flow.param.plugin.className	The Java class name of the client-specific FlowParamPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.FlowParamPlugin interface.
plan.logic.pricing.plan.plugin.className	The Java class name of the client-specific PricingPlanPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.PricingPlanPlugin interface.
plan.logic.likeitem.plan.plugin.className	The Java class name of the client-specific LikeItemPlanPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.LikeItemPlanPlugin interface.
plan.logic.store.base.plugin.classname	The Java class name of the client-specific StoreBasePlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.StoreBasePlugin interface.
plan.logic.packconfig.plugin.className	The Java class name of the client-specific PackConfigPlanPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.PackConfigPlanPlugin interface.
plan.logic.packopt.plugin.className	The Java class name of the client-specific PackOptPlugin module. This class must implement the com.profitlogic.buying4p.logic.itemplugins.PackOptPlugin interface.
plan.logic.dcflowparam.plugin.className	The Java class name of the client-specific DCFlowParamPlugin module. This class must implement the com.profitlogic.buying4p.logic.plugins.DCFlowParamPlugin interface.

Application Plug-In Properties

This section describes other plug-able classes that should rarely, if ever, change.

Property Name

plan.logic.notification.className

Property Description

The Java class name of the Notification module. This class must implement the com.profitlogic.buying4p.logic.notification.Notifier interface.

Background Event Subsystem Properties

This section describes background event processing service properties. New types of background services can be added in a plug-able fashion. JMS point-to-point is used to send events to an MDB that drives the back-end processing.

Property Name

plan.backend.processing.queue.jndi.url

Property Description

The JNDI url for the back-end processing queue. The default value is un-set and commented out.

plan.backend.processing.queue.
context.factory

The JNDI context factory for the back-end processing queue.

plan.backend.processing.queue.
connection.factory

The JNDI connection factory for the back-end processing queue.

plan.backend.processing.queue

The JNDI queue name for the back-end processing queue.

plan.backend.num.forecast.event.listeners

The number of forecast event message listeners to register. The default value is 5.

plan.backend.num.auto.populate.event.
listeners

The number of autopopulate event message listeners to register. The default value is 3.

plan.backend.num.packopt.start.event.
listeners

The number of packopt start task event message listeners to register. The default value is 2.

plan.backend.num.packopt.run.event.
listeners

The number of packopt run task event message listeners to register. The default value is 10.

plan.backend.java.naming.security.
credentialsplan.backend.java.naming.security.
principal

When messages are sent from EventSubscriberMDB, credentials to the remove Geneva system must be passed.

plan.RunForecast.jndi_name

plan.RunForecast.home_class

plan.RunForecast.method_name

Defines the RunForecast back-end service. All three properties are needed so that the EventSubscriberMDB can reflect upon the associated session facade method.

plan.PackOptStart.jndi_name

plan.PackOptStart.home_class

plan.PackOptStart.method_name

Defines the service that starts pack optimization. All three properties are needed so that the EventSubscriberMDB can reflect upon the associated session facade method.

plan.PackOptStartItemDate.jndi_name

plan.PackOptStartItemDate.home_class

plan.PackOptStartItemDate.method_
name

Defines the service that starts pack optimization. All three properties are needed so that the EventSubscriberMDB can reflect upon the associated session facade method.

plan.PackOptStartItem.jndi_name

plan.PackOptStartItem.home_class

plan.PackOptStartItem.method_name

Defines the service that starts pack optimization. All three properties are needed so that the EventSubscriberMDB can reflect upon the associated session facade method.

Property Name	Property Description
plan.RunAutoPopulateFinish.jndi_name plan.RunAutoPopulateFinish.home_class plan.RunAutoPopulateFinish.method_name	Defines the service that performs auto populate. All three properties are needed so that the EventSubscriberMDB can reflect upon the associated session facade method.

Forecast Properties

This section describes the properties that control Calc Engine forecast behavior.

Property Name	Property Description
plan.facade.forecast.mode.synchronous	A flag used to switch between synchronous and asynchronous messages. The default value is true.
plan.forecastStatus.statusIds plan.forecastStatus.statusValues	Maps forecast status IDs to their display status. Both properties must have the same number of values and in the correct order. The status IDs are constants in com.profitlogic.buying4p.util.ItemPlanningConst. Note: The values are duplicated in GridResources.properties.
plan.forecast.updateSegments	This property is set to true if segments should update after a forecast.

Pack Optimization Properties

This section describes the pack optimization properties.

Property Name	Property Description
plan.packopt.minAlloc	If an allocation is less than the value of this property, then the allocation is ignored.
plan.packopt.zero.hiCost	The cost of assigning any receipts to a zero allocation.
plan.packopt.store.hiCost	The cost of exceeding a store's allocation.
plan.packopt.store.loCost	The cost of not reaching a store's allocation.
plan.packopt.store.inventory.hiCost	The cost of exceeding the presentation minimum.
plan.packopt.store.inventory.loCost	The cost of not reaching the presentation minimum.
plan.packopt.agg.pmin.hiCost	The cost of exceeding the aggregate minimum. Ignored.
plan.packopt.agg.pmin.loCost	The cost of not reaching the aggregate minimum.
plan.packopt.agg.cappingFactor	$\text{max} = \text{min} * \text{capping_factor}$
plan.packopt.agg.pmax.hiCost	The cost of exceeding the aggregate maximum.
plan.packopt.agg.pmax.loCost	The cost of not reaching the aggregate maximum.
plan.packopt.mode	The mode used to run the pack optimization. Values are: PREVIEW, AUTO_ACCEPT, and EITHER (which gives the user the choice). The default value is EITHER.
plan.packopt.packs.default.availability	A flag that specifies whether or not packs are available for the start-of-time. True indicates that packs are available. False indicates packs are not available.
plan.packopt.adaptive.pack.opt	Performs pack optimization on deliveries, one period at a time. After each pack optimization, the adaptive pack opt adjusts the pack opt request in the next period to account for any deficit or surplus from the previous period.

Calc Parameters Properties

This section describes the calc parameter properties.

Property Name	Property Description
plan.params.pricingPlan.storesetId	The client-specified storeset ID, used for choosing subsets for pricing events. Must match the storeset ID in the Planned_Promo_Ad_Des data feed.
plan.params.flowDates.basic.forecastLength	The number of months to run a forecast if the out-of-stock date is not specified.
plan.plannedItem.deliveryFrequencies	The client-specified list of delivery frequencies and schedules set in Flow Parameters for a planned item.
plan.plannedItem.deliveryFrequencyIds	
plan.plannedItem.deliveryFrequencyId. weekly	
plan.plannedItem.deliveryFrequency. PeriodWeeks	

Property Name	Property Description
plan.likeitem.style.store.weights.default	The default value for the store weight in the Like Item screen in the UI. The value can be set to the level sequence number in the hierarchy. The value must correspond to the value in the Forecast configuration.
plan.likeitem.color.store.weights.default	The default value for the store weight in the Like Item screen in the UI. The value can be set to the level sequence number in the hierarchy. The value must correspond to the value in the Forecast configuration.

Flow View Properties

This section describes the flow view properties.

Property Name	Property Description
plan.params.storeSetNames plan.params.all.storeset.name	The client-specific list of storeset names for Flow View. The first storeset ID should be the ALL subset. All the storeset names must be ADMIN storesets.
plan.flowView.default.storeset.name plan.flowView.default.subset.name	The default subset for the Flow View page that is initially displayed (ALL).
plan.flowView.default.receipt.types	The default of which receipts are show in Flow View page. The list is comma separated. Values are Store and DC.

Login Properties

This section describes the login properties.

Property Name	Property Description
plan.logout.destination	The client logout destination.
plan.login.action	The client login destination.
plan.login.external	The client external login destination.
plan.token.validate	The validate login token. Comment to disable.
plan.token.key	The login token key.
plan.token.lifespan	The login token life span in milliseconds.

Auxiliary URL Properties

This section describes auxiliary urls used by the application.

Property Name	Property Description
plan.reports.url	The url for the client reports html document. Typically, this is a link to the Merchant Desktop.
plan.help.expr	The client help expression (javascript).

Cache Limit Sizes and Cache Management Properties

This section describes properties that set limitations on caches.

Property Name	Property Description
plan.itemPlanAdapter.getWeeklyStoreSegments.fetchSize	Used to tune the database.
plan.memoryUsage.appAllowedMemoryUsagePercent	Memory usage permitted by the application before the caches are adjusted.
plan.itemCacheMgr.getCacheLimit.cacheLimit	The limit of the item cache.
plan.itemCacheMgr.getPercentToPurge.percentToPurge	The percentage of items purged when the memory used exceeds appAllowedMemoryUsagePercent.
plan.CalcParamsCache.getCacheLimit.cacheLimit	The limit of the Calc Param cache.
planCalcParamsCache.getPercentToPurge.percentToPurge	The percentage of Calc Param cache objects purged when the memory used exceeds appAllowedMemoryUsagePercent.

Internal Developer Properties

This section describes internal properties used only by developers.

Property Name	Property Description
plan.facade.forecast.curves	Flag that indicates whether forecasted data is obtained from curves or from ITEM_LOCATION_PERIODS_TBL. This property should not be changed.
plan.itemCacheMgr.suppress.cacheUse	Suppresses check ItemSegmentBean objects in the cache in order to measure performance curves vs. ITEM_LOCATION_PERIODS_TBL
plan.statesvc.trueSingleton.debug	The default is false. If set to true, TrueSingletons will not be. They will be only one per JVM.
plan.enable.common.singleton	If set to true, the new singleton implementation is used. If set to false, the old UserSessionMgrEJB implementation is used.
plan.suppress.bitem.forecast	Suppresses the B-item forecast message sent when a plan is first opened. To skip this step, un-comment and set value to true.
plan.suppress.auto.populate	Suppresses the auto-population of basic items when a plan is opened. To skip this step, un-comment and set value to true.
plan.bitem.forecast.duration.for.annualization	The number of weeks of sales forecast (for a basic item from the start of a plan period) that is required to make the item receipts eligible for annualization.
plan.flow.comp.storeproc	To compute the flow through a stored procedure, set to true.

Property Name	Property Description
plan.assortmentView.caching.for.need	If set to false, the Need-segment cache for the Assortment View is disabled.
plan.calcparams.cache.enabled	If set to false, the Calc Params cache is disabled.

Grid Configuration Properties

This section describes front-end grid configuration properties.

Property Name	Property Description
gridkeys	The grid configuration keys and files.
GridResources	The grid configuration resources file.
internalColumns customColumns	The column configuration files.

Metrics Key List

This section describes the metric list. The value of each metric is a comma-separated list of FQCNs. The FQCNs represent the derivation at each level (Weekly, Monthly, and Grand Total).

Note: The key of each metric (e.g., AveUnitRetail) should match the key names specified in any grid xml file that references the metric.

Property Name	Property Description
plan.logic.metrics	A comma-separated list of keys to metrics. Each key is another property that defines a comma-separated list of the classes that implement the metrics at the various levels.

Miscellaneous Properties

This section describes miscellaneous properties, including highest search level.

Property Name	Property Description
highest.plan.search.level	The highest search level at which the client ID is unique. If client IDs are not unique, then a search may return duplicates, which could cause errors. Note: Client IDs should be unique within the parent level, so choosing the immediate child of CHAIN is generally safe. Since the CHAIN level always exists, the default value is CHAIN.
plan.util.item.type.fashion plan.util.item.type.basic plan.util.item.type.VMI	The strings that a customer uses for the item types for Fashion, Basic, and VMI. The strings are found in Planned_Item_Types_Tbl. The strings must be unique. If an item type is not used, then assign any unused string to that property.
plan.util.item.type.default	The default type for the plan worklist table. The string should exist in Planned_Item_Types_Tbl.

Property Name	Property Description
plan.pricing.plan.sliding.window.enabled	Enables or disables the sliding window logic for the pricing plan.
plan.pricing.plan.sliding.window.start.offset.from.today	The offset from today's date, used to set the beginning of the 30-day sliding window for planned promotions. If plan.pricing.plan.sliding.window.enabled is set to false, then this property is ignored. Note: Internal use only. Should always be set to 0.
plan.plan.library.planupdater.default	The default for the Plan Library screen's PlanUpdater option. If this property is set to 0, then the PlanUpdater option in the Plan Library is turned off when a new plan is created. If this property is set to 1, then the PlanUpdater option in the Plan Library is turned on when a new plan is created.

Validation Properties

This section describes the validation properties.

Property Name	Property Description
plan.flow.dates.validate.oos.within.plan	Indicates whether or not carry-over is allowed. If carry-over is not allowed, then the flow date validation ensures that an item's out-of-stock date does not occur after the end of the plan.
plan.flow.dates.item.types.oos.not.required	A comma-separated list of the item types that can have a blank out-of-stock date.
plan.flow.dates.validate.against.actual.oos	Indicates whether or not the flow dates should be validated against the actual OOS date in the merchandise hierarchy.
plan.plan.worklist.save.validate.ap.quantities	Indicates whether or not the AP view quantities should be validated on Plan worklist save. If set to true, the AP view errors are displayed as warnings. If set to false, extra processing is avoided.

Date Properties

This section describes the date properties.

Property Name	Property Description
plan.use.calendar.dates	Determines whether the application displays calendar years or fiscal years. If set to true, the Info Header, New Plan, and Extended Time Periods in Item View and What-If use calendar years.

Forecast Properties

The section describes the forecast configuration properties.

Property Name	Property Description
<code>plan.forecast.config.variables</code>	Specifies the variable names to be used. Variables named here can be used by <code>plan.forecast.config.weights.option.##</code>
<i>Variable Name</i>	Defines the bean value associated with the variable.
<code>plan.forecast.config.weights.style.options</code>	Defines the options that should be enabled for the Style level drop-downs in the Like Item screen. The order of the drop-down is determined by the order of the options. Options should be in <code>##</code> format. Note that <code>0.#</code> is used for internal options and <code>1.#</code> is used for external options.
<code>plan.forecast.config.weights.color.options</code>	Defines the options that should be enabled for the Color level drop-downs in the Like Item screen. The order of the drop-down is determined by the order of the options. Options should be in <code>##</code> format. Note that <code>0.#</code> is used for internal options and <code>1.#</code> is used for external options.
<code>plan.forecast.config.weights.option.0.keys</code>	Defines internal option key. Currently not being used.
<code>plan.forecast.config.weights.option.0.8</code>	Defines the internal option value. Currently not being used.
<code>plan.forecast.config.weights.option.0.9</code>	Defines the internal option value. Currently not being used.
<code>plan.forecast.config.weights.option.1.keys</code>	Defines the keys used for the external options in a pipe delimited list. Variables used are defined in the variables property.
<code>plan.forecast.config.weights.option.1.2</code>	Defines the external option value. Key value pairs are passed to the Calc Engine. They consist of a combination of a specific value, such as merchandise hierarchy level, and variables, such as start date and end date. These values are used by <code>IR_GET_EXT_STORE_WEIGHTS</code> to calculate the external store weight passed to the Calc Engine.
<code>plan.forecast.config.weights.option.1.3</code>	Defines the external option value. See above for details.
<code>plan.forecast.config.weights.option.1.4</code>	Defines the external option value. See above for details.
<code>plan.forecast.config.weights.option.1.5</code>	Defines the external option value. See above for details.
<code>plan.forecast.config.weights.option.1.6</code>	Defines the external option value. See above for details.
<code>plan.forecast.config.weights.option.1.7</code>	Defines the external option value. See above for details.

The chapter contains the following:

- “Introduction” on page 1
- “Plan Default Metrics” on page 2

Introduction

Plan’s Configurable Metrics Framework allows a user to change and remove existing metrics as well as create new metrics in order to meet unique business needs. It is located in `metrics_intf.jar`, which is available on the Plan CD image.

The Framework includes developer documentation and a User’s Guide that provides a description of the Framework and details about changing or removing existing metrics and creating new metrics. The documentation is formatted as Javadoc documentation and is included inside the `metrics_intf.jar`. The documents can be accessed by extracting the `doc` folder from the JAR and then opening `doc/index.html`. From there, the User’s Guide can be accessed from the `com.profitlogic.plan.logic.metrics` package.

Plan comes with a set of metrics. The metrics and their defined derivations are listed at the end of this chapter. Most of the metrics in the Framework are displayed within Plan in the Item View screen and the What If screen only. However, some are hidden by default and are used only as internal dependencies with other metrics.

It is assumed that anyone attempting to modify metrics or create new metrics is familiar with Java and can write a class implementation.

Any new metrics must be placed under the metrics directory under `<ConfigRoot>` in a jar file (for example, `<ConfigRoot>/plan/metrics/custom_metrics.jar`). The metrics directory may not exist by default, so you may have to create it.

Since some metrics have dependencies with other metrics, it is important to consider all the ramifications of any changes:

- Dependencies can only exist between metrics defined at the same level (weekly, monthly, or grand total) or between a metric at a given level and a metric at a lower level.
- Dependencies cannot be defined that create a cyclical dependency.
- All the classes defined in the dependency must be implemented.
- Dependencies do not need to be defined between a metric and the four input values, `SalesU`, `Sales$`, `RcptU`, and `Rcpt$`.

Any changes to metrics can potentially impact several files: plan.properties and the appropriate Grid xml files (which in turn may require changes to the GridResources.properties file).

Metrics have internal and external names. The internal names are hard-coded and are used to resolve dependencies and to perform metric lookups. The numbers 0 - 999 are reserved by Oracle for its core metrics, so any custom metrics should be assigned a unique internal name using numbers beginning with 1,000. External names are specified both in plan.properties (which defines what metrics are available) and in the relevant Grid xml file and are configurable.

The format to use for metrics in the Grid xml files is Metrics.<External Metric name>. The External Metric Name must be the same in the Grid xml file and plan.properties.

Plan can replace the default plan.properties file during upgrades, so users should create another file in the client directory in order to override metrics.

The following rules apply to creating new metrics:

- Class implementations must implement the Metric interface or the WeightedMetric interface. The abstract base classes, AbstractMetric and AbstractRollup, which implement these classes, are provided.
- Implementations for the same metric must return the same internal name.
- Implementations for different metrics must return different internal names.
- Implementations must declare any dependencies on other metrics using the Metric.getDependencies(com.profitlogic.plan.logic.metrics.MetricsCalculator) method.

To access the complete documentation on getting started with Configurable Metrics, refer to the developer documentation mentioned at the beginning of this chapter.

Plan Default Metrics

The following table lists in alphabetical order the standard Plan metrics and how they are calculated.

Note: LTD = Life-to-Date. This value is the cumulative running total since the beginning of the total time period.

Table 10–1 Plan Standard Metrics

Display Name	Column Key	Description	Weekly Calculation	Monthly/Grand Total Calculation
Sales AUR	AveUnitRetail	Average unit retail for a time period	SalesDollars/ SalesUnits	SalesDollars/ SalesUnits
AUC	AveUnitCost	Average unit cost for a time period	SalesDollarsAtCost/ SalesUnits	SalesDollarsAtCost/ SalesUnits
Not displayed by default	AverageInventoryDollars	Average inventory dollars (at retail)	None	(SUM [WeeklyBohDollars] + MonthlyEohDollars) / (NumSellingWeeks + 1)
Not displayed by default	AverageInventoryDollarsAtCost	Average inventory \$ at Cost	None	(SUM [WeeklyBohDollarsAtCost] + MonthlyEohDollarsAtCost) / (NumSellingWeeks + 1)

Table 10–1 (Cont.) Plan Standard Metrics

Display Name	Column Key	Description	Weekly Calculation	Monthly/Grand Total Calculation
Not displayed by default	AverageInventoryUnits	Average inventory units	None	$(\text{SUM} [\text{WeeklyBohUnits}] + \text{MonthlyEohUnits}) / (\text{NumSellingWeeks} + 1)$
BOH \$	BohDollars	Beginning on-hand dollars for a time period	PrevWeekEohDollars	First Week Of Month EohDollars
BOH U	BohUnits	Beginning on-hand units for a time period	PrevWeekEohUnits	First Week of Month EohUnits/First Month of Grand Total EohUnits
EOH \$	EohDollars	Ending on-hand dollars for a time period	$\text{EohUnits} * \text{CurrPermPrice}$	$\text{EohUnits} * \text{CurrPermPrice}$
EOH U	EohUnits	Ending on-hand units for a time period	$\text{InventoryUnits} - \text{SalesUnits}$	Last Week of Month EohUnits = Monthly / Last Month of Grand Total EohUnits = Grand Total
WOS	WeeksOfSupply Inventory Note: This combines metrics WeeksOfSupply + ExcessInventory (listed here collectively as WOS)	Forward weeks of supply	Current BohUnits / SUM [Weekly SalesUnits by week] where SUM [Weekly SalesUnits by week] < BohUnits. Remainder is dropped. If not enough sales weeks to reach constraint, then WOS metric will append a + for excess inventory.	Monthly WOS = First week of month. Grand Total WOS = First Month of Display.
GP%	GrossProfitPct	Gross profit percentage by time period	$(\text{IMU} * 100) - \text{MD} \% * (1 - \text{IMU}) - (\text{Merch Charge} \% + \text{Shrink} \% + \text{Freight} \% * (1 - \text{IMU}))$	$(\text{SUM} [\text{WeeklyGrossProfit Dollars}] / \text{MonthlySalesDollars}) * 100$
GP\$	GrossProfit Dollars	Gross profit dollars by time period	$[\text{WeeklyGrossProfitPct} * \text{WeeklySalesDollars}] / 100$	SUM [WeeklyGrossProfit Dollars]
GPROI	Gproi	Gross profit return on investment	None	$\text{SUM} [\text{WeeklyGrossProfitDollars}] / \text{AverageInventoryDollarsAtCost}$
IMU	InitialMarkup	Initial markup	$(\text{Retail} - \text{Cost}) / \text{Retail}$	$(\text{Retail} - \text{Cost}) / \text{Retail}$
Pricing Status	PricingStatus	Markdown or promotion indicator (MD, PR, REG)	If CurrPermPrice <> Initial Retail then MD. If Weekly Promo exists, then PR. Else REG.	None

Table 10–1 (Cont.) Plan Standard Metrics

Display Name	Column Key	Description	Weekly Calculation	Monthly/Grand Total Calculation
MD Sales%	PermMarkdownSalesDollarsPct	Ratio of clearance sales \$ to total sales \$	PermMarkdownSalesUnits / SalesUnits	PermMarkdownSalesUnits / SalesUnits
MD%	MarkdownPct	Ratio of markdown \$ to total sales \$ at retail	MD\$/SalesDollarsAtRetail	MarkdownDollars/SalesDollarsAtRetail
MD Sales \$	PermMarkdownSalesDollars	Sales \$ of clearance items	If PricingStatus = MD, then SalesDollars. Else 0.	SUM [WeeklyPermMarkdownSalesDollars]
MD Sales U	PermMarkdownSalesUnits	Sales units of clearance items	If PricingStatus = MD, then SalesUnits. Else 0.	SUM [WeeklyPermMarkdownSalesUnits]
Perm MD\$	PermMarkdownDollars	Permanent markdown \$ for a time period triggered by change in total inventory valuation due to a permanent markdown.	BohUnits * (last week of CurrPermPrice - this week CurrPermPrice) + ReceiptUnits * (Retail - CurrPermPrice)	SUM [WeeklyPermMarkdownDollars]
POS MD\$	PosMarkdownDollars	POS (temporary) markdown \$ component of weekly sales	Weekly SalesUnits * (CurrPermPrice - AveUnitRetail)	SUM [WeeklyPosMarkdown]
POS Sales %	PosMarkdownSalesDollarsPct	Ratio of promotional sales \$ to total sales \$	PosMarkdownSalesUnits / SalesUnits * 100	SUM [WeeklyPosMarkdownSalesUnits] / SUM [WeeklySalesUnits]
POS Sales \$	PosMarkdownSalesDollars	Sales \$ of promotional items	If PricingStatus = PR, then SalesDollars. Else 0.	SUM [WeeklyPosMarkdownSalesDollars]
POS Sales U	PosMarkdownSalesUnits	Sales units of promotional items	If PricingStatus = PR, then SalesUnits. Else 0.	SUM [WeeklyPosMarkdownSalesUnits]
Receipt \$	ReceiptDollars	Total receipt dollars by time period	This is an input value	SUM [WeeklyReceiptDollars]
%Receipt \$	ReceiptDollarsPct	Percentage of receipt dollars for a time period contribution to the total receipts for the item.	ReceiptDollars / Total ReceiptDollars	SUM [WeeklyReceiptDollars] / Total ReceiptDollars
Receipt U	ReceiptUnits	Total receipt units by time period	This is an input value	SUM [WeeklyReceiptUnits]
%Receipt U	ReceiptUnitsPct	Percentage of receipt units for a time period contribution to the total receipts for the item. This field is editable for the AP segment and allows the user to alter the receipt flow.	ReceiptUnits / Total ReceiptUnits	SUM [WeeklyStoreReceiptUnits] / Total ReceiptUnits

Table 10–1 (Cont.) Plan Standard Metrics

Display Name	Column Key	Description	Weekly Calculation	Monthly/Grand Total Calculation
Sales \$	SalesDollars	Total sales dollars by time period	$(\text{SalesUnits} * \text{AveUnitRetail}) * (1 - \text{AveUnitRetail Adjustment} / 100)$ Hard-coded input value	SUM [Weekly SalesDollars]
%Sales \$	SalesDollarsPct	Percentage of sales dollars for a time period contribution to the total sales dollars for the item.	$\text{SalesDollars} / \text{Total SalesDollars}$	$\text{SUM [Weekly SalesDollars]} / \text{Total SalesDollars}$
Sales U	SalesUnits	Total Sales units by time period	This is an input value	SUM [Weekly SalesUnits]
%Sales U	SalesUnitsPct	Percentage of sales units for a time period contribution to the total sales units for the item.	$\text{SalesUnits} / \text{Total SalesUnits}$	$\text{SUM [Weekly SalesUnits]} / \text{Total SalesUnits}$
%ST \$	SellThruPctDollars	% Sell through in dollars	$\text{LTD_SalesDollars} / \text{LTD_ReceiptDollars}$	$\text{LTD_SalesDollars} / \text{LTD_ReceiptDollars}$
%ST U	SellThruPctUnits	% Sell through in units	$\text{LTD_SalesUnits} / \text{LTD_ReceiptUnits}$	$\text{LTD_SalesUnits} / (\text{LTD_ReceiptUnits})$
Total MD\$	MarkdownDollars	Total markdown \$, either promo or permanent markdown	$\text{PosMarkdown} + \text{PermMarkdown Dollars}$	$\text{PosMarkdown} + \text{PermMarkdownDollars}$
TO \$	TurnoverDollars	Turnover dollars	None	$\text{SalesDollars} / \text{AverageInventory Dollars}$
TO U	TurnoverUnits	Turnover units	None	$\text{SalesUnits} / \text{AverageInventoryUnits}$
Not displayed by default	NumSellingWeeks	Weeks in item's life cycle	Week included if within item life cycle	SUM [Weekly NumSellingWeeks]
Not displayed by default	BohDollarsAtCost	BOH \$ at item cost	$\text{BohUnits} * \text{InitialCost}$	SUM [Weekly BohDollarsAtCost]
Not displayed by default	EohDollarsAtCost	EOH \$ at item cost	$\text{EohUnits} * \text{InitialCost}$	SUM [Weekly EohDollarsAtCost]
Not displayed by default	SalesDollarsAtCost	Sales \$ at item cost	$\text{SalesUnits} * \text{InitialCost}$	SUM [Weekly SalesDollarsAtCost]
Not displayed by default	SalesDollarsAtRetail	Sales \$ at initial retail	$\text{SalesUnits} * \text{InitialRetail}$	SUM [SalesDollarsAtRetail]
Not displayed by default	InventoryUnits	Inventory available during time period	$\text{BohUnits} + \text{ReceiptUnits}$	$\text{BohUnits} + \text{ReceiptUnits}$

Table 10–1 (Cont.) Plan Standard Metrics

Display Name	Column Key	Description	Weekly Calculation	Monthly/Grand Total Calculation
Not displayed by default	CurrPermPrice	Permanent retail price at beginning of time period (no promo effect) (Weighted by inventory)	CurrPermPrice at beginning of week (from Pricing Plan)	CurrPermPrice for first week of month
Not displayed by default	InitialRetail	Initial retail of item (Weighted by inventory)	Retail set from Plan Worklist	Retail set from Plan Worklist
Not displayed by default	InitialCost	Initial cost of item (Weighted by inventory)	Cost set from Plan Worklist	Cost set from Plan Worklist