Oracle® Retail Plan

Configuration Guide

Release 12.3

November 2009

ORACLE®

Oracle® Retail Plan Configuration Guide, 12.3

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

Contents

4 Database Configuration

5 Configurable Data Attributes

6 Merchant Desktop and Retail Data Mart Configuration

9   Front End Metrics

Index

Preface

Audience

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

Related Documents

For more information, see the following documents in the Oracle Retail Plan 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*

Supplemental Documentation on MetaLink

The following technical white paper is available on the MetaLink Web site:

MetaLink Note 737759.1: Oracle Retail Password Security Management Guide

Oracle Retail Plan and Place applications now include a Password Security Management module that helps you generate and store encrypted passwords used in the application. This enables you to meet the password encryption security policies or laws mandated for your business.

The white paper introduces you to the Password Security Management module and the methodology adopted to encrypt the passwords. It also includes information that will help you perform administrative or recovery tasks efficiently.

Customer Support

- https://metalink.oracle.com

When contacting Customer Support, please provide:

- Product version and program/module name

- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to recreate
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

If you are installing the application for the first time, you install either a base release (for example, 12.0) or a later patch release (for example, 12.2). If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Patch documentation can contain critical information related to the base release and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site (with the exception of the Data Model which is only available with the release packaged code):

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

This chapter contains the following:

- "About the Plan Configuration Guide" on page 1-1
- "What's in This Book" on page 1-1

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. A list of the chapters in the Plan Configuration Guide.
- Chapter 2 – User Management. Used to create, modify, and inactivate user accounts.
- Chapter 3 – Business Rule Manager. Used to customize business rule settings, that is, the constraints used by Price to determine markdowns and forecasts.
- Chapter 4 – Database Configuration. Various post-installation required configurations.
- Chapter 5 – Configurable Data Attributes. Used to specify custom data that can be viewed through the Plan UI.
- Chapter 6 – 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 7 – Plan Updater. Updates existing plans based on data changes.
- Chapter 8 – Properties File. Contains a list of the properties found in the plan.properties file, which are configurable.
- Chapter 9 – Front End Metrics. Describes the metrics used in the Plan application. It also provides an overview of the configurable metrics framework.

2

User Management

This chapter contains the following sections:

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. The user can submit selected items in a Plan. Requires PLAN_READ and PLAN_WRITE.

- PLAN_DELETE. The user can delete an existing Plan.

- PLAN_CREATE. The user can create a new Plan.

- PLAN_READ. The user 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. The user can publish plans to RDM in advance of the nightly batch process.

- CALC_PARAM_WRITE. The user 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. The user 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. This action requires PLAN_READ and CALC_PARAM_READ. It is used to 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. This role contains the entire set of Plan default actions.

- PLAN_BIZADMIN. This role contains the entire set of Plan default actions and BRM default actions.

- PLANNER. This role contains all Plan default actions except Submit and Publish.

- PLAN_VIEWER. This role 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, 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 permits 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 permits 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 permits 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, refer to the Oracle Retail Plan Administration 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 *<Plan_Installation>/modules/tools/conf* directory. The following table lists the XML files along with the relevant schema and database table names:

Table 2–1 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 *<Plan_Installation>/modules/tools/bin/plansecurityload.sh*. The script uses 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 *<Plan_Installation>/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 completes the load process 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 (for example, *plan_user_set.xml*) to load.
app-server-home	The application server home directory (for example, *c:/bea8/weblogic81/server*).
um-script-dir	The directory where the UM loader script is located (for example, */modules/tools/bin/bulkloader.sh*)
jndi-url	The jndi url for the application server. This defaults to *t3://localhost:7001*
init-ctx-factory	The initial context factory class for the server. This defaults to weblogic.jndi.WLInitialContextFactory

User Management Security

In order to ensure the security of the application, User Management provides the following security features:

- The AUTOCOMPLETE attribute in 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" xmls: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
<?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.
```

3

Business Rule Manager

This chapter contains the following sections:

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 documentation set.

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.

Two Plan business rules are optional, ANNUAL_BASICS_PLAN_MIN and OOS_ VALID_DAYS_OF_MONTH.

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 12 default business rules accessible through the BRM.

Table 3–1   Default Plan Business Rules

Business Rule Name and UI Display Name	Business Rule Description	Default Value	Lowest MH Level Allowed	Lowest LH Level Allowed	Set for Attri-butes? Y/N
AUR_ADJUSTMENT_PCT AUR Adjustment Pct	The average unit retail discount percent. Used, for example, in the display of new pricing plans.	0.025	Color	Chain	Y (MH only)
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	Planning Level (the MH level at which Plans are created)	Chain	N
SAFETY_STOCK Safety Stock	The additional time, in days, to stock an item so that it is not out-of-stock. This setting has an effect on the Receipt Flow calculation. This setting can be set at the STORE level in the location hierarchy.	14	Color	Store	Y
TOLERANCE_PCT Tolerance Pct	The acceptable variance, expressed as a percent, from the Store Financial Plan. This setting is used by the Fit to Plan action in the Assortment View. The AP quantities are increased or decreased in order to match the Financial Plan within the value for this setting.	0.1	Planning Level (the MH level at which Plans are created)	Chain	N
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	Style	Chain	Y (MH only)
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	Planning Level (the MH level at which Plans are created)	Chain	N

Table 3–1   (Cont.) Default Plan Business Rules

Business Rule Name and UI Display Name	Business Rule Description	Default Value	Lowest MH Level Allowed	Lowest LH Level Allowed	Set for Attri- butes? Y/N
STORE_QTY_LOW_ THRESHOLD Store Qty Low Threshold Attribute	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	Planning Level (the MH level at which Plans are created)	Chain	N
STORE_QTY_HIGH_ THRESHOLD Store Qty High Threshold Attribute	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	Planning Level (the MH level at which Plans are created)	Chain	N
PLAN_WINDOW_BEGIN_ OFFSET In-Season Plan Start Offset	The number of weeks before the plan start date to include when sales are checked for in-season items.	4	Department	Chain	N
PLAN_WINDOW_END_ OFFSET In-Season Plan End Offset	The number of weeks after the plan end date to include when sales are checked for in-season items.	4	Department	Chain	N
ASV_LOW_THRESHOLD Assortment View Low Threshold	Used to indicate the first threshold (expressed as a percentage) at which the performance of an item is highlighted. Yellow indicates Under-performance and Green indicates Over-performance.	20			N
ASV_HIGH_THRESHOLD Assortment View High Threshold	Used to indicate the second threshold (expressed as a percentage) at which the performance of an item is highlighted. Red indicates Under-performance and Green indicates Over-performance.	50			N

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 that 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> is the root directory of the application configuration files.
- <rule_definitions> is the name of the xml file that contains the rule definitions.
- <host> is the application server host
- <port> is 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 that 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 of 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. LoadBRMAttributeValues 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 3-8.

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 3–2   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

Table 3–2   (Cont.) Business Rule Instances Standard Interface Specification

Attribute	Attribute Description	Data Type	Maximum Length	Nullable Y/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 3–3   Business Rule Manager Properties

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

4

Database Configuration

The chapter contains the following:

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 .

- 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

For more information on the delphi.properties, see .

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 4-1.

Pricing Event Types

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

Table 4–1   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 4–2   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 4–3   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 4–4   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 4–5   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 4–6   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_SQC	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 4–7   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 4–7   (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_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. The Merchandise ID.

- Location_ID. The Location ID.

- Attribute_Mask. This is 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_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.

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_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_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. The Merchandise ID.
- Merchandise_Level_Desc. the Merchandise level description.
- AS_Merchandise_ID. The As Merchandise ID.
- AS_Merchandise_Level_Desc. The AS merchandise level description.
- Attribute_Value. This is 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_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. The Merchandise ID.
- Location_ID. The Location ID.
- Promo_Start_Dt. The beginning of a promotion.
- Promo_End_Dt. The end of a 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. The Merchandise ID.

- Location_ID. The Location ID.

- Attribute_Mask. This is used for constructing attribute values

Customization: By default, the IR_PROMO_CORP_ATTRIBUTS 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_REFORECAST_PARAMETER_VW

Forecasts are flagged as invalid by Plan:

- when stores are added or deleted

- when a plan changes because of a pricing event

- when the actual data from the weekly data load updates the sales data for an item that is not at the end of its lifecycle

Items with invalid forecasts are re-forecast during the batch process.

The IR_REFORECAST_PARAMETER_VW view is used to prioritize items for re-forecasting. After the daily batch load, Plan Updater is run. The Updater picks up items that have been flagged for re-forecasting or items that it could not get to during previous runs because of prioritization. The batch processing window may not be large enough to handle all the active items, so this view is used to indicate whether or not an item should be re-forecast and what the priority is for the item.

Plans are re-forecast in descending priority order. The average priority order for all items in a given plan is calculated. This average value is used to determine the plan priority order for re-forecasting.

The priority logic for items is as follows:

- Non-basic items (fashion and key) with a sell-through value of 3% - 50% have a priority of 1.

- Non-basic items (fashion and key) with a sell-through value of 51% - 75% have a priority of 2.

- Basic items have a priority of 3.

- Fashion item with a sell-through value that is not in the P1 or P2 range have a priority of 4.

This view has the following columns:

- Sell_Thru_Pcnt. The total actual sales / total receipts (actual and future), expressed as a percentage.

- P1_LBound. The lower boundary for Priority 1, expressed as a percentage.

- P1_UBound. The upper boundary for Priority 1, expressed as a percentage.

- P2_LBound. The lower boundary of Priority 2, expressed as a percentage.

- P2_UBound. The upper boundary of Priority 2, expressed as a percentage.

The ITEM_SETS_TBL has a Forecast_Priority column that contains the value calculated by the view.

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. The Merchandise ID.

- Merchandise_Level_Desc. The Merchandise level description.

- Location_ID. The Location ID.

- Location_Level_Desc. The Location level description.

- AS_Merchandise_ID. The AS Merchandise ID.

- AS_Merchandise_Level_Desc. The AS Merchandise level description.

- AS_Location_ID. The AS Location ID.

- AS_Location_Level_Desc. The 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.

5

Configurable Data Attributes

This chapter contains the following:

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.

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 5–1   CDA Data Type

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

Table 5–1   (Cont.) CDA Data Type

Data Attribute Type	Data Type
Date	NUMBER
Currency	VARCHAR
Currency	NUMBER (2 columns)

The following tables supports extension by the CDA Administrative Utility:

Table 5–2   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

6

Merchant Desktop and Retail Data Mart Configuration

The chapter contains the following:

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 that in both cases you should 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 <rmi-install-dir>/rmiStart.bat and <rmi-install-dir>/rmiStop.bat 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 6–1   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 6–2   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 ensures 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 this process 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. (However, this is not case sensitive.) 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. On Microstrategy Desktop, this setting is found in the Job Prioritization tab on the Database Instances window (appears when you right-click on your project and select the Prioritization option).

    > **Note:** To ensure optimal performance, Oracle recommends that you set up a value of 15 for the *High*, *Medium*, and *Low* fields in the *Number of connections by priority* area (on the Job Prioritization tab).

- 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

The following RDM metrics are available for use in reports. The metrics are divided into the following categories:

- Plan Metrics. Miscellaneous metrics.

- Plan Metrics\ACT Metrics. Actual History metrics.

- Plan Metrics\AP Metrics. Assortment Planning metrics.

- Plan Metrics\Budget Metrics. Budget metrics.

- Plan Metrics\Count Metrics. Distinct or total number of a particular item appearing across records.

- Plan Metrics\DC metrics. Distribution Center metrics include receipt inventory costs.

- Plan Metrics\Need Metrics. Bottom-up item forecasting metrics.

- Plan Metrics\OPT Metrics. Optimized History metrics.

- Pack Optimization Metrics. Stored at the pack/store/week level.

Many of these categories are sub-divided into Color and Style metrics. Further sub-divisions can include:

- FinView Metrics. Aggregated metrics such as sums, totals, and sub-totals.

- ItemByTime Metrics. Sales, receipt, and inventory metrics over time.

- GP LY Metrics. Gross profit performance metrics to which a Last Year transformation is applied. Such metrics are appropriate for Optimized History.

- Receipt $ and Units Metrics. Receipt units and dollar amounts.

- Sales $ and Units Metrics. Sales amounts, sales units, and sell-through ratios.

- Inventory, Performance, and Cost Metrics. Beginning inventory units, ending inventory units, dollar amounts, performance ratios.

The following two tables list the metrics:

Table 6–3, " Metrics and Metric Folder Path Names" lists the complete path name for each folder and lists the metrics that are found in each folder. Each metric within a folder is assigned a number that identifies it as a member of the folder (for example 1-a is assigned to the first metric in the first folder).

Table 6–4, " Metrics, Metric Descriptions, and Metric Calculations" lists all the metrics in alphabetical order and includes a description of the metric and the metric calculation. The same number that is assigned to the metric in Table 6–3 is also listed in Table 6–4 so that the number can be used as a cross reference between the two tables.

For detailed descriptions, including the list of metrics in a specific report, for all Plan reports, consult the application User Guide.

Table 6–3   Metrics and Metric Folder Path Names

Folder	Metric	No.
\\Public Objects\Metrics\Plan Metrics		
	Max Out Date	1-a
	Max Out Date - extension	1-b
	Min Out Date	1-c

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	Min Out Date - extension	1-d
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\		
	ACT AUC	2-a
	ACT Average Selling Retail	2-b
	ACT Average Selling Retail (Last Year)	2-c
	ACT BOH $ (Last Year)	2-d
	ACT BOH Units (Last Year)	2-e
	ACT BOH Units (Sum Subtotal) (Last Year)	2-f
	ACT Current Retail by EOH	2-g
	ACT Current Retail by Sls	2-h
	ACT Current Retail by Sls (Last Year)	2-i
	ACT GP $	2-j
	ACT GP $ Hist (Last Year)	2-k
	ACT GP %	2-l
	ACT GP % (Last Year)	2-m
	ACT GP % Hist (Last Year)	2-n
	ACT IMU	2-o
	ACT MD Sls $	2-p
	ACT MD Sls Units	2-q
	ACT Perm MD $	2-r
	ACT POS MD $	2-s
	ACT POS Sls $	2-t
	ACT POS Sls Units	2-u
	ACT Reg Sls $	2-v
	ACT Reg Sls Units	2-w
	ACT Total MD $	2-x
	ACT Total MD $ (Last Year)	2-y
	ACT Total MD %	2-z
	ACT WOS	2-aa
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT FinView Metrics\		
	ACT BOH Units (Sum Subtotal)	3-a
	ACT EOH Units (Sum Subtotal)	3-b
	ACT Sls Units (Cum.) FV	3-c
	ACT ST % (Cum.)	3-d
	ACT ST % (Cum.)	3-e
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT ItemByTime Metrics\		
	ACT EOH $	4-a

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	ACT EOH $ (Last Year)	4-b
	ACT EOH Units (Last Subtotal)	4-c
	ACT EOH Units (Last Subtotal) (Last Year)	4-d
	ACT Sls $ (Cum.)	4-e
	ACT Sls $ (Cum.) (Last Year)	4-f
	ACT Sls Units (Cum.)	4-g
	ACT Sls Units (Cum.) (Last Year)	4-h
	ACT ST % (Cum.)	4-i
	ACT ST % (Cum.)	4-j
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT Rcpt $		
	ACT Rcpt $	5-a
	ACT Rcpt $ (Last Year)	5-b
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT Rcpt Units		
	ACT Rcpt Units	6-a
	ACT Rcpt Units (Last Year)	6-b
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT Sales $		
	ACT Sls $	7-a
	ACT Sls $(Last Year)	7-b
\\Public Objects\Metrics\Plan Metrics\ACT Metrics\ACT Sales Units		
	ACT Sls Units	8-a
	ACT Sls Units (Last Year)	8-b
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)		
	AP Color - ADJ GP %	9-a
	AP Color - ADJ IMU	9-b
	AP Color - ASR	9-c
	AP Color - AUC	9-d
	AP Color - BOH	9-e
	AP Color - BOH $	9-f
	AP Color - Current Retail by Sls (Domestic)	9-g
	AP Color- DC EOH Amt	9-h
	AP Color - DC EOH Units	9-i
	AP Color - GM $	9-j
	AP Color - GP %	9-k
	AP Color - IMU	9-l
	AP Color - In Transit Inv Amt	9-m
	AP Color - In Transit Inv Units	9-n
	AP Color - Initial Retail By Sls	9-o

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	AP Color - Item Cost by Sls (Domestic)	9-p
	AP Color - MD BOH	9-q
	AP Color - MD BOH $	9-r
	AP Color - MD Sls $	9-s
	AP Color - MD Sls Units	9-t
	AP Color - Perm MD $	9-u
	AP Color - POS BOH	9-v
	AP Color - POS BOH $	9-w
	AP Color - POS MD $	9-x
	AP Color - POS Sls $	9-y
	AP Color - POS Sls Units	9-z
	AP Color - Reg BOH	9-aa
	AP Color - Reg BOH $	9-bb
	AP Color - Reg Sls $	9-cc
	AP Color - Reg Sls Units	9-dd
	AP Color - Total MD $	9-ee
	AP Color - Total MD %	9-ff
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Color Assortment Review Metrics		
	AP Color - # of Stores with Inv	10-a
	AP Color Rcpt Units / # of Stores	10-b
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Color FinView Metrics		
	AP Color BOH Units (Sum Subtotal)	11-a
	AP Color EOH Units (Sum Subtotal)	11-b
	AP Color Sls Units (Cum.) FV	11-c
	AP Color ST % (Cum.)	11-d
	AP Color ST % (Cum.)	11-e
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Color ItemByTime Metrics		
	AP Color EOH $ (LTD)	12-a
	AP Color EOH Units (LTD)	12-b
	AP Color Rcpt $ (Cum.)	12-c
	AP Color Rcpt Units (Cum.)	12-d
	AP Color Sls $ (Cum.)	12-e
	AP Color Sls Units (Cum.)	12-f
	AP Color ST % (Cum.)	12-g
	AP Color ST % (Cum.)	12-h

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Rcpt Sales\		
	AP Color Rcpt $	13-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Rcpt Units\		
	AP Color Rcpt Units	14-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Sales\		
	AP Color Sls $	15-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Color MH Level)\AP Units\		
	AP Color Sls Units	16-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\		
	AP Style - ADJ GP %	17-a
	AP Style - ADJ IMU	17-b
	AP Style - ASR	17-c
	AP Style - AUC	17-d
	AP Style - BOH $	17-e
	AP Style - Current Retail by Sls (Domestic)	17-f
	AP Style - DC EOH Amt	17-g
	Ap Style - DC EOH Units	17-h
	AP Style - GM $	17-i
	AP Style - GP %	17-j
	AP Style - IMU	17-k
	AP Style - In Transit Inv Amt	17-l
	AP Style - In Transit Inv Units	17-m
	AP Style - Initial Retail By Sls	17-n
	AP Style - Item Cost by Sls (Domestic)	17-o
	AP Style - MD BOH	17-p
	AP Style - MD BOH $	17-q
	AP Style - MD Sls $	17-r
	AP Style - MD Sls Units	17-s
	AP Style - Perm MD $	17-t
	AP Style - Perm MD %	17-u
	AP Style - POS BOH	17-v
	AP Style - POS BOH $	17-w
	AP Style - POS MD $	17-x
	AP Style - POS MD %	17-y
	AP Style - POS Sls $	17-z
	AP Style - POS Sls Units	17-aa
	AP Style - Reg BOH	17-bb

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	AP Style - Reg BOH $	17-cc
	AP Style - Reg Sls $	17-dd
	AP Style - Reg Sls Units	17-ee
	AP Style - Total MD $	17-ff
	AP Style - Total MD %	17-gg
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Rcpt Sales		
	AP Style Rcpt $	18-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Rcpt Units		
	AP Style Rcpt Units	19-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Sales		
	AP Style Sls $	20-a
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Style Assortment Review Metrics		
	AP Style - # of Stores with Inv	21-a
	AP Style Rcpt Units / # of Stores	21-b
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Style FinView Metrics		
	AP Style BOH Units (Sum Subtotal)	22-a
	AP Style EOH Units (Sum Subtotal)	22-b
	AP Style Sls Units (Cum.) FV	22-c
	AP Style ST % (Cum.)	22-d
	AP Style ST % (Cum.)	22-e
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Style ItemByTime Metrics		
	AP Style EOH $ (LTD)	23-a
	AP Style EOH Units (LTD)	23-b
	AP Style Rcpt $ (Cum.)	23-c
	AP Style Rcpt Units (Cum.)	23-d
	AP Style Sls $ (Cum.)	23-e
	AP Style Sls Units (Cum.)	23-f
	AP Style ST % (Cum.)	23-g
	AP Style ST % (Cum.)	23-h
\\Public Objects\Metrics\Plan Metrics\AP Metrics\AP Metrics (Style MH Level)\AP Units		
	AP Style Sls Units	24-a
\\Public Objects\Metrics\Plan Metrics\Budget Metrics		
	Budgeted Perm MD %	25-a
	Budgeted POS MD %	25-b
	Budgeted Total MD %	25-c

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
\\Public Objects\Metrics\Plan Metrics\Budget Metrics\Company Budget Metrics		
	Budgeted Charge Pct	26-a
	Budgeted Freight Pct	26-b
	Budgeted Perm MD $	26-c
	Budgeted POS MD $	26-d
	Budgeted Shrink Pct	26-e
	Budgeted Total MD $	26-f
\\Public Objects\Metrics\Plan Metrics\Budget Metrics\Store Budget Metrics\Budget Rcpt Sales		
	Store Budget Rcpt $	27-a
\\Public Objects\Metrics\Plan Metrics\Budget Metrics\Store Budget Metrics\Budget Sales		
	Store Budget Sls $	28-a
\\Public Objects\Metrics\Plan Metrics\Count Metrics		
	# of Colors	29-a
	# of Colors (Last Year)	29-b
	# of Styles	29-c
	# of Styles (Last Year)	29-d
	# of Weeks	29-e
	AP Color - # of Colors with Inv	29-f
	AP Color - # of Styles with Inv	29-g
	AP Color - # of Weeks	29-h
	AP Style - # of Style with Inv	29-i
	AP Style - # of Weeks	29-j
	LSP Color - # of Colors with Inv	29-k
	LSP Color - # of Styles with Inv	29-l
	LSP Color - # of Weeks	29-m
\\Public Objects\Metrics\Plan Metrics\DC Metrics\DC Metrics (Color MH Level)		
	DC Rcpt $	30-a
	DC Rcpt Units	30-b
\\Public Objects\Metrics\Plan Metrics\DC Metrics\DC Metrics (Style MH Level)		
	DC Rcpt $	31-a
	DC Rcpt Units	31-b
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)		
	LSP Color - ADJ CG %	32-a
	LSP Color - ADJ IMU	32-b
	LSP Color - ASR	32-c
	LSP Color - AUC	32-d
	LSP Color - BOH	32-e

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	LSP Color - BOH $	32-f
	LSP Color - Current Retail by Sls (Domestic)	32-g
	LSP Color - DC EOH Amt	32-h
	LSP Color DC EOH Units	32-i
	LSP Color - GM $	32-j
	LSP Color - GP $	32-k
	LSP Color - IMU	32-l
	LSP Color - In Transit Inv Amt	32-m
	LSP Color - In Transit Inv Units	32-n
	LSP Color - Initial Retail by SlsS	32-o
	LSP Color - Item Cost by Sls (Domestic)	32-p
	LSP Color - MD BOH	32-q
	LSP Color - MD BOH $	32-r
	LSP Color - MD Sls $	32-s
	LSP Color - MD Sls Units	32-t
	LSP Color - Perm MD $	32-u
	LSP Color - POS BOH	32-v
	LSP Color - POS BOH $	32-w
	LSP Color - POS MD $	32-x
	LSP Color - POS Sls $	32-y
	LSP Color - POS Sls Units $	32-z
	LSP Color - Reg BOH	32-aa
	LSP Color - Reg BOH $	32-bb
	LSP Color - Reg Sls $	32-cc
	LSP Color - Reg Sls Units	32-dd
	LSP Color - Total MD $	32-ee
	LSP Color - Total MD %	32-ff
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Color Assortment Review Metrics		
	LSP Color - # of Stores with Inv	33-a
	LSP Color Rcpt Units/# of Stores	33-b
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Color FinView Metrics		
	LSP Color BOH Units (Sum Subtotal)	34-a
	LSP Color EOH Units (Sum Subtotal)	34-b
	LSP Color Sls Units (Cum.) FV	34-c
	LSP Color ST % (Cum.)	34-d
	LSP Color ST % (Cum.)	34-e

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Color ItemByTime Metrics		
	LSP Color EOH $ (LTD)	35-a
	LSP Color EOH Units (LTD)	35-b
	LSP Color Rcpt $ (Cum.)	35-c
	LSP Color Rcpt Units (Cum.)	35-d
	LSP Color Sls $ (Cum.)	35-e
	LSP Color Sls Units (Cum.)	35-f
	LSP Color ST % (Cum.)	35-g
	LSP Color ST % (Cum.)	35-h
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Rcpt Sales		
	LSP Color Rcpt $	36-a
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Rcpt Units		
	LSP Color Rcpt Units	37-a
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Sales		
	LSP Color Sls $	38-a
\\Public Objects\Metrics\Plan Metrics\LSP Metrics\LSP Metrics (Color MH Level)\LSP Units		
	LSP Color Sls Units	39-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)		
	Need Color - ASR	40-a
	Need Color - BOH $	40-b
	Need Color - DC EOH Amt	40-c
	Need Color - DC EOH Units	40-d
	Need Color - GM $	40-e
	Need Color - GP %	40-f
	Need Color - IMU	40-g
	Need Color - In Transit Inv Amt	40-h
	Need Color - In Transit Inv Units	40-i
	Need Color - Perm MD $	40-j
	Need Color - POS MD $	40-k
	Need Color - Total MD $	40-l
	Need Color - Total MD %	40-m
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Color FinView Metrics		
	Need Color BOH Units (Sum Subtotal)	41-a
	Need Color EOH Units (Sum Subtotal)	41-b
	Need Color Sls Units (Cum.) FV	41-c
	Need Color ST % (Cum.)	41-d

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	Need Color ST % (Cum.)	41-e
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Color ItemByTime Metrics\		
	Need Color - WOS	42-a
	Need Color EOH Units (LTD)	42-b
	Need Color Rcpt $ (Cum.)	42-c
	Need Color Rcpt Units (Cum.)	42-d
	Need Color Sls $ (Cum.)	42-e
	Need Color Sls Units (Cum.)	42-f
	Need Color ST % (Cum.)	42-g
	Need Color ST % (Cum.)	42-h
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Rcpt Sales\		
	Need Color Rcpt $	43-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Rcpt Units\		
	Need Color Rcpt Units	44-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Sales\		
	Need Color Sls $	45-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Color MH Level)\Need Units\		
	Need Color Sls Units	46-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\		
	Need Style - ASR	47-a
	Need Style - BOH	47-b
	Need Style - BOH $	47-c
	Need Style - DC EOH Amt	47-d
	Need Style - DC EOH Units	47-e
	Need Style - GM $	47-f
	Need Style - GP %	47-g
	Need Style - IMU	47-h
	Need Style - In Transit Inv Amt	47-i
	Need Style - In Transit Inv Units	47-j
	Need Style - Perm MD $	47-k
	Need Style - POS MD $	47-l
	Need Style - Total MD $	47-m
	Need Style - Total MD %	47-n
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Rcpt Sales\		
	Need Style Rcpt $	48-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Rcpt Units\		

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
	Need Style Rcpt Units	49-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Sales		
	Need Style Sls $	50-a
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Style FinView Metrics		
	Need Style BOH Units (Sum Subtotal)	51-a
	Need Style EOH Units (Sum Subtotal)	51-b
	Need Style Sls Units (Cum.) FV	51-c
	Need Style ST % (Cum.)	51-d
	Need Style ST % (Cum.)	51-e
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Style ItemByTime Metrics		
	Need Style - WOS	52-a
	Need Style EOH Units (LTD)	52-b
	Need Style Rcpt $ (Cum.)	52-c
	Need Style Rcpt Units (Cum.)	52-d
	Need Style Sls $ (Cum.)	52-e
	Need Style Sls Units (Cum.)	52-f
	Need Style ST % (Cum.)	52-g
	Need Style ST % (Cum.)	52-h
\\Public Objects\Metrics\Plan Metrics\Need Metrics\Need Metrics (Style MH Level)\Need Units		
	Need Style Sls Units	53-a
\\Public Objects\Metrics\Plan Metrics\OPT Metrics		
	OPT Average Selling Retail	54-a
	OPT Average Selling Retail (Last Year)	54-b
	OPT BOH $	54-c
	OPT BOH $ (Last Year)	54-d
	OPT GP $	54-e
	OPT GP $ (Last Year)	54-f
	OPT GP %	54-g
	OPT GP % (Last Year)	54-h
	OPT IMU	54-i
	OPT Perm MD $	54-j
	OPT POS MD $	54-k
	OPT Total MD $	54-l
	OPT Total MD $ (Last Year)	54-m
	OPT Total MD %	54-n

Table 6–3   (Cont.) Metrics and Metric Folder Path Names

Folder	Metric	No.
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT FinView Metrics		
	OPT BOH Units (Sum Subtotal)	55-a
	OPT BOH Units (Sum Subtotal) (Last Year)	55-b
	OPT EOH Units (Sum Subtotal)	55-c
	OPT Sls Units (Cum.) FV	55-d
	OPT ST % (Cum.)	55-e
	OPT ST % (Cum.)	55-f
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT ItemByTime Metrics		
	OPT EOH Units (Last Subtotal)	56-a
	OPT Sls $ (Cum.)	56-b
	OPT Sls Units (Cum.)	56-c
	OPT ST % (Cum.)	56-d
	OPT ST % (Cum.)	56-e
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT Rcpt Sales		
	OPT Rcpt $	57-a
	OPT Rcpt $ (Last Year)	57-b
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT Rcpt Units		
	OPT Rcpt Units	58-a
	OPT Rcpt Units (Last Year)	58-b
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT Sales		
	OPT Sls $	59-a
	OPT Sls $ (Last Year)	59-b
\\Public Objects\Metrics\Plan Metrics\OPT Metrics\OPT Units		
	OPT Sls Units	60-a
	OPT Sls Units (Last Year)	60-b
\\Public Objects\Metrics\Plan Metrics\Pack Opt Metrics		
	Pack Opt Average Cost	61-a
	Pack Opt Average Retail Price	61-b
	Pack Opt IMU%	61-c
	Pack Opt Inside Units	61-d
	Pack Opt Item Setup Cost	61-e
	Pack Opt Item Setup Retail	61-f
	Pack Opt Num Packs	61-g
	Pack Opt Total Cost	61-h
	Pack Opt Total Receipt Units	61-i
	Pack Opt Total Retail	61-j

The following table lists all the metrics in alphabetical order and includes a description of the metric and the metric calculation. The same number that is assigned to the metric in Table 6–3 is also listed in this table so that the number can be used as a cross reference between the two tables.

Table 6–4   Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
29-a	# of Colors	Distinct number of colors.	Count<Distinct>
29-b	# of Colors (Last Year)	Distinct number of colors.	Count<Distinct>
29-c	# of Styles	Distinct number of styles.	Count<Distinct>
29-d	# of Styles (Last Year)	Distinct number of styles.	Count<Distinct>
29-e	# of Weeks	Max number of weeks an item is on sale.	Max([Actual History Num Selling Weeks])
2-a	ACT AUC	Weekly WAC from sales file. For aggregations, weighted average based on EOH.	Sum(([Actual History Wtd Item Cost By Sls] * [Actual History Sls Units]))~ / Sum([Actual History Sls Units])~
2-b	ACT Average Selling Retail	Actual History average sale retail price.	[ACT Sls $] / [ACT Sls Units]
2-c	ACT Average Selling Retail (Last Year)	Actual History average sale retail price.	Sum([Actual History Sls Amt])~[Week Period - Last Year]| / Sum([Actual History Sls Units])~[Week Period - Last Year]|
2-d	ACT BOH $ (Last Year)	Actual History beginning on hand inventory dollar amount.	Sum(([Actual History Wtd Cur Retail By Boh] * [Actual History BOH]))
2-e	ACT BOH Units (Last Year)	Actual Historybeginning on hand inventory units.	Sum([Actual History BOH])
3-a	ACT BOH Units (Sum Subtotal)	Actual History beginning on hand inventory units.	Sum([Actual History BOH])
2-f	ACT BOH Units (Sum Subtotal) (Last Year)	Actual beginning on hand inventory units.	Sum([Actual History BOH])
2-g	ACT Current Retail by EOH	Current Retail Price weighted by sales.	Sum(([Actual History Wtd Cur Retail By Eoh] * [Actual History EOH]))~|[Week Period] / Sum([Actual History EOH])~|[Week Period]
2-h	ACT Current Retail by Sls	Actual History current retail price weighted by sale units.	Sum(([Actual History Wtd Cur Retail By Sls] * [Actual History Sls Units]))~ / Sum([Actual History Sls Units])~
2-i	ACT Current Retail by Sls (Last Year)	Actual current retail price.	Sum(([Actual History Wtd Cur Retail By Sls] * [Actual History Sls Units]))~[Week Period - Last Year]| / Sum([Actual History Sls Units])~[Week Period - Last Year]|
4-a	ACT EOH $	Actual History end on hand inventory dollar amount.	[ACT EOH Units (Last Subtotal)] * [ACT Current Retail by EOH]
4-b	ACT EOH $ (Last Year)	Actual History end on hand inventory dollar amount.	Sum(([Actual History Wtd Cur Retail By Eoh] * [Actual History EOH]))
4-c	ACT EOH Units (Last Subtotal)	Actual History end on hand inventory units.	Sum([Actual History EOH])
4-d	ACT EOH Units (Last Subtotal) (Last Year)	Actual History end on hand inventory units.	Sum([Actual History EOH])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
3-b	ACT EOH Units (Sum Subtotal)	Actual History end on hand inventory units.	Sum([Actual History EOH])
2-j	ACT GP $	Actual History gross profit dollar amount.	Sum([Actual History Gross Profit Amt]) {~}
2-k	ACT GP $ Hist (Last Year)	Actual gross profit dollar amount.	Sum([Actual History Gross Profit Amt]) {~} | [Week Period - Last Year] |
2-l	ACT GP %	Actual History gross profit % margin.	[ACT IMU] - ([ACT Total MD %] * (1 - [ACT IMU]))
2-m	ACT GP % (Last Year)	Actual History gross profit % margin.	((Sum(((NullToZero([Actual History Wtd Init Retail By Sls]) - NullToZero([Actual History Wtd Item Cost By Sls])) * [Actual History Sls Units]))~[Week Period - Last Year]| * Sum(([Actual History Wtd Init Retail By Sls] * [Actual History Sls Units]))~[Week Period - Last Year]|) - (Sum((([Actual History Wtd Item Cost By Sls] * [Actual History Sls Units]))~~[Week Period - Last Year]| * Sum((NullToZero([Actual History Pos Md Amt]) ~ NullToZero([Actual History Perm Md Amt])))~[Week Period - Last Year]|)) / Power(Sum(([Actual History Wtd Init Retail By Sls] * [Actual History Sls Units]))~[Week Period - Last Year]|,2)
2-n	ACT GP % Hist (Last Year)	Actual gross profit percent.	Sum([Actual History Gross Profit Amt])~[Week Period - Last Year]| / Sum([Actual History Sls Amt])~[Week Period - Last Year]|
2-o	ACT IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum((([Actual History Wtd Init Retail By Sls] * [Actual History Sls Units]) - ([Actual History Wtd Item Cost By Sls] * [Actual History Sls Units])))~ / Sum(([Actual History Wtd Init Retail By Sls] * [Actual History Sls Units]))~
2-p	ACT MD Sls $	Dollar amount associated with MD Sales Units {If Derived for Optimized History: If Retail < Initial Retail, then = Sales $, else 0}.	Sum([Actual History Md Sales Amt])
2-q	ACT MD Sls Units	Any unit sold where Current Inventory Price is < Original Retail price {If Derived for Optimized History: If Retail < Initial Retail, then = Sales U, else 0}.	Sum([Actual History Md Sales Units])
2-r	ACT Perm MD $	Actual History permanent markdown dollars.	Sum([Actual History Perm Md Amt])
2-s	ACT POS MD $	Actual History point of sale markdown dollars.	Sum([Actual History Pos Md Amt])
2-t	ACT POS Sls $	Dollar amount associated with POS Sales Units.	Sum([Actual History Pos Sls Amt])
2-u	ACT POS Sls Units	Any unit sold where Sales Price < Original Retail Price.	Sum([Actual History Pos Sls Units])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
5-a	ACT Rcpt $	Actual History receipt dollars.	Sum([Actual History Rcpt Amt])
5-b	ACT Rcpt $ (Last Year)	Actual History receipt dollars.	Sum([Actual History Rcpt Amt])
6-a	ACT Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([Actual History Rcpt Units])
6-b	ACT Rcpt Units (Last Year)	Store receipts from Distros and Transfers (new data feed).	Sum([Actual History Rcpt Units])
2-v	ACT Reg Sls $	Actual History sales dollars.	([ACT Sls $] - [ACT POS Sls $]) - [ACT MD Sls $]
2-w	ACT Reg Sls Units	Actual History sales units.	([ACT Sls Units] - [ACT POS Sls Units]) - [ACT MD Sls Units]
7-a	ACT Sls $	Actual History sales dollars.	Sum([Actual History Sls Amt])
4-e	ACT Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([ACT Sls $])
4-f	ACT Sls $ (Cum.) (Last Year)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>(Sum([Actual History Sls Amt])~[Week Period - Last Year]|)
7-b	ACT Sls $(Last Year)	Actual History sales dollars.	Sum([Actual History Sls Amt])
8-a	ACT Sls Units	Actual History sales units.	Sum([Actual History Sls Units])
4-g	ACT Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([ACT Sls Units])
4-h	ACT Sls Units (Cum.) (Last Year)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>(Sum([Actual History Sls Units])~[Week Period - Last Year]|)
3-c	ACT Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	Sum([Actual History Sls Units])
8-b	ACT Sls Units (Last Year)	Actual History sales units.	Sum([Actual History Sls Units])
3-d	ACT ST % (Cum.)	Actual History sell-through percent.	[ACT Sls Units (Cum.) FV] / ([ACT Sls Units (Cum.) FV] ~ [ACT EOH Units (Sum Subtotal)])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
4-i	ACT ST % (Cum.)	Actual sell through rate.	[ACT Sls Units (Cum.)] / ([ACT Sls Units (Cum.)] ~ [ACT EOH Units (Last Subtotal)])
4-j	ACT ST % (Cum.)	Sell through percent.	[ACT Sls Units (Cum.)] / [ACT Rcpt Units (Cum.)]
3-e	ACT ST % (Cum.)	Sell through percent - using FV metrics.	[ACT Sls Units (Cum.) FV] / [ACT Rcpt Units (Cum.) FV]
2-x	ACT Total MD $	Actual History total markdown dollars.	[ACT POS MD $] ~ [ACT Perm MD $]
2-y	ACT Total MD $ (Last Year)	Actual History total markdown dollars.	Sum((NullToZero([Actual History Pos Md Amt]) ~ NullToZero([Actual History Perm Md Amt])))
2-z	ACT Total MD %	Markdown percent.	([ACT Total MD $] / Sum(([Actual History Wtd Cur Retail By Sls] * [Actual History Sls Units])) {~})
2-aa	ACT WOS	Basic weeks of supply using inventory divided by last week of sales. One Plus and RW use 4 week average.	Sum([Actual History EOH])~ | [Week Period] / Sum([Actual History Sls Units])~ | [Week Period]
29-f	AP Color - # of Colors with Inv	Number of colors available with inventory.	Count<Distinct>
10-a	AP Color - # of Stores with Inv	Number of stores with inventory on hand.	Count<Distinct>
29-g	AP Color - # of Styles with Inv	Number of colors available with inventory.	Count<Distinct>
29-h	AP Color - # of Weeks	Number of weeks merchandise is on sale.	Max([AP Color Num Selling Weeks])
9-a	AP Color - ADJ GP %	Adjusted gross profit % margin by budgeted shrink percentage.	([AP Color - ADJ IMU] - ([AP Color - Total MD %] * (1 - [AP Color - ADJ IMU]))) - ((([Budgeted Charge Pct] ~ [Budgeted Shrink Pct]) ~ ([Budgeted Freight Pct] * (1 - [AP Color - ADJ IMU])))
9-b	AP Color - ADJ IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	(([AP Color - Current Retail by Sls(Domestic)] - [AP Color - Item Cost by Sls(Domestic)]) * 0.86) / [AP Color - Current Retail by Sls(Domestic)]
9-c	AP Color - ASR	Average Selling Retail price.	[AP Color Sls $] / [AP Color Sls Units]
9-d	AP Color - AUC	Weekly WAC from sales file. For aggregations, weighted average based on EOH is used.	Sum(([AP Color Sls Units] * [AP Color Wtd Item Cost By Sls]))~ / Sum([AP Color Sls Units])~
9-e	AP Color - BOH	Beginning on hand inventory units.	Sum([AP Color BOH])
9-f	AP Color - BOH $	Beginning on hand inventory dollar amount.	Sum(([AP Color Wtd Cur Retail By Boh] * [AP Color BOH]))
9-g	AP Color - Current Retail by Sls (Domestic)	Current Retail Price weighted by sales.	Sum(([AP Color Wtd Cur Retail By Sls] * [AP Color Sls Units]))~<Orig Ind = D> / Sum([AP Color Sls Units])~<Orig Ind = D>

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
9-h	AP Color - DC EOH Amt	DC EOH amount across all DCs. Calculated as DC EOH units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([AP Color Dc Eoh Amt]) {~}
9-i	AP Color - DC EOH Units	DC EOH units across all DCs.	Sum([AP Color Dc Eoh Units]) {~}
9-j	AP Color - GM $	Gross Margin dollar amount.	"ApplySimple(""Case when #0 > 0 then #1 Else #2 End"", [AP Color - Actual Flag], [AP Color - GM $ mapped], ([AP Color - GP %] * [AP Color Sls $]))"
9-k	AP Color - GP %	Gross Profit percent.	([AP Color - IMU] - ([AP Color - Total MD %] * (1 - [AP Color - IMU]))) - ((([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ ([Budgeted Freight Pct] * (1 - [AP Color - IMU])))
9-l	AP Color - IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum(((([AP Color Wtd Init Retail By Sls] * [AP Color Sls Units]) - ([AP Color Wtd Item Cost By Sls] * [AP Color Sls Units])))~ / Sum((([AP Color Wtd Init Retail By Sls] * [AP Color Sls Units]))~
9-m	AP Color - In Transit Inv Amt	In Transit Inventory Amount for all DCs. Calculated as In transit units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([AP Color In Transit Inv Amt]) {~}
9-n	AP Color - In Transit Inv Units	In transit total units for all DCs.	Sum([AP Color In Transit Inv Units]) {~}
9-o	AP Color - Initial Retail By Sls	AP initial retail price weighted by sales.	Sum((([AP Color Wtd Init Retail By Sls] * [AP Color Sls Units]))~ / Sum([AP Color Sls Units])~
9-p	AP Color - Item Cost by Sls (Domestic)	Item cost weighted by sales.	Sum((([AP Color Wtd Item Cost By Sls] * [AP Color Sls Units]))~<Orig Ind = D> / Sum([AP Color Sls Units])~<Orig Ind = D>
9-q	AP Color - MD BOH	Markdown beginning on hand inventory.	Sum([AP Color Mkdn Boh])
9-r	AP Color - MD BOH $	Markdown beginning on hand dollar amount.	Sum((([AP Color Wtd Cur Retail By Boh] * [AP Color Mkdn Boh]))
9-s	AP Color - MD Sls $	Dollar amount associated with MD Sales Units {If Derived for Optimized History: If Retail < Initial Retail, then = Sales $, else 0}.	Sum([AP Color Mkdn Sls Amt])
9-t	AP Color - MD Sls Units	Any unit sold where Current InventoryPrice is < Original Retail price {If Derived for Optimized History: If Retail < Initial Retail, then = Sales U, else 0}.	Sum([AP Color Mkdn Sls Units])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
9-u	AP Color - Perm MD $	Permanent markdown dollars.	Sum([AP Color Perm Md Amt])
9-v	AP Color - POS BOH	Point of sale beginning on hand inventory units.	Sum([AP Color Pos Boh])
9-w	AP Color - POS BOH $	Point of sale beginning on hand inventory dollar amount.	Sum(([AP Color Wtd Init Retail By Boh] * [AP Color Pos Boh]))
9-x	AP Color - POS MD $	Point of sale markdown dollars.	Sum([AP Color Pos Md Amt])
9-y	AP Color - POS Sls $	Dollar amount associated with POS Sales Units.	Sum([AP Color Pos Sls Amt])
9-z	AP Color - POS Sls Units	Any unit sold where Sales Price < Original Retail Price.	Sum([AP Color Pos Sls Units])
9-aa	AP Color - Reg BOH	Beginning on hand inventory units.	([AP Color BOH Units (Sum Subtotal)] - [AP Color - MD BOH]) - [AP Color - POS BOH]
9-bb	AP Color - Reg BOH $	Beginning on hand inventory dollar amount.	([AP Color - BOH $] - [AP Color - MD BOH $]) - [AP Color - POS BOH $]
9-cc	AP Color - Reg Sls $	Sales dollar amount.	([AP Color Sls $] - [AP Color - MD Sls $]) - [AP Color - POS Sls $]
9-dd	AP Color - Reg Sls Units	Sales units.	([AP Color Sls Units] - [AP Color - MD Sls Units]) - [AP Color - POS Sls Units]
9-ee	AP Color - Total MD $	Total markdown dollars.	[AP Color - Perm MD $] ~ [AP Color - POS MD $]
9-ff	AP Color - Total MD %	Markdown percent.	([AP Color - Total MD $] / Sum(([AP Color Wtd Cur Retail By Sls] * [AP Color Sls Units])) {~})
11-a	AP Color BOH Units (Sum Subtotal)	Beginning on hand inventory units.	Sum([AP Color BOH])
12-a	AP Color EOH $ (LTD)	Life to date end on hand dollar amount.	[AP Color Rcpt $ (Cum.)] - [AP Color Sls $ (Cum.)]
12-b	AP Color EOH Units (LTD)	Life to date end on hand inventory units.	[AP Color Rcpt Units (Cum.)] - [AP Color Sls Units (Cum.)]
11-b	AP Color EOH Units (Sum Subtotal)	End on hand inventory units.	Sum([AP Color EOH])
13-a	AP Color Rcpt $	Receipt dollars.	Sum([AP Color Rcpt Amt])
12-c	AP Color Rcpt $ (Cum.)	Cumulative Sum of Receipt dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Color Rcpt $])
14-a	AP Color Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([AP Color Rcpt Units])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
12-d	AP Color Rcpt Units (Cum.)	Cumulative Sum of metric from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Color Rcpt Units])
10-b	AP Color Rcpt Units / # of Stores	Receipts units per store.	[AP Color Rcpt Units] / [AP Color - # of Stores with Inv]
15-a	AP Color Sls $	Sales dollars.	Sum([AP Color Sls Amt])
12-e	AP Color Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Color Sls $])
16-a	AP Color Sls Units	Sales units	Sum([AP Color Sls Units])
12-f	AP Color Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Color Sls Units])
11-c	AP Color Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	Sum([AP Color Sls Units])
12-g	AP Color ST % (Cum.)	Sell through percent at color level (cumulative).	[AP Color Sls Units (Cum.)] / [AP Color Rcpt Units (Cum.)]
11-d	AP Color ST % (Cum.)	Sell through percent at color level (sum of subtotal).	[AP Color Sls Units (Cum.) FV] / ([AP Color Sls Units (Cum.) FV] ~ [AP Color EOH Units (Sum Subtotal)])
12-h	AP Color ST % (Cum.)	Sell through percent at color level (cumulative).	[AP Color Sls Units (Cum.)] / [AP Color Rcpt Units (Cum.)]
11-e	AP Color ST % (Cum.)	Sell through percent.	[AP Color Sls Units (Cum.) FV] / [AP Color Rcpt Units (Cum.) FV]
29-i	AP Style - # of Style with Inv	Number of distinct styles.	Count<Distinct>
21-a	AP Style - # of Stores with Inv	Number of stores with inventory on hand.	Count<Distinct>
29-j	AP Style - # of Weeks	Number of weeks merchandise is on sale.	Max([AP Style Num Selling Weeks])
17-a	AP Style - ADJ GP %	Gross profit percent adjusted by budget shrink percentage.	([AP Style - ADJ IMU] - ([AP Style - Total MD %] * (1 - [AP Style - ADJ IMU]))) - ((([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ [Budgeted Freight Pct] * (1 - [AP Style - ADJ IMU])))
17-b	AP Style - ADJ IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	(([AP Style - Current Retail by Sls(Domestic)] - [AP Style - Item Cost by Sls(Domestic)]) * 0.86) / [AP Style - Current Retail by Sls(Domestic)]

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
17-c	AP Style - ASR	Average Selling Retail price.	[AP Style Sls $] / [AP Style Sls Units]
17-d	AP Style - AUC	Weekly WAC from sales file. For aggregations, weighted average based on EOH is used.	Sum(([AP Style Sls Units] * [AP Style Wtd Item Cost By Sls]))~ / Sum([AP Style Sls Units])~
17-e	AP Style - BOH $	Beginning on hand dollars.	Sum(([AP Style Wtd Cur Retail By Boh] * [AP Style BOH]))
17-f	AP Style - Current Retail by Sls (Domestic)	Current Retail Price weighted by sales.	Sum(([AP Style Wtd Cur Retail By Sls] * [AP Style Sls Units]))~<Orig Ind = D> / Sum([AP Style Sls Units])~<Orig Ind = D>
17-g	AP Style - DC EOH Amt	DC EOH amount across all DC. Calculated as DC EOH units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([AP Style Dc Eoh Amt]) {~}
17-h	AP Style - DC EOH Units	DC EOH units across all DCs.	Sum([AP Style Dc Eoh Units]) {~}
17-i	AP Style - GM $	Gross Margin dollar amount.	ApplySimple("Case when #0 > 0 then #1 Else #2 End", [AP Style - Actual Flag], [AP Style - GM $ mapped], ([AP Style - GP %] * [AP Style Sls $]))
17-j	AP Style - GP %	Gross profit percent.	([AP Style - IMU] - ([AP Style - Total MD %] * (1 - [AP Style - IMU]))) - (([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ ([Budgeted Freight Pct] * (1 - [AP Style - IMU])))
17-k	AP Style - IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum((([AP Style Wtd Init Retail By Sls] * [AP Style Sls Units]) - ([AP Style Wtd Item Cost By Sls] * [AP Style Sls Units])))~ / Sum(([AP Style Wtd Init Retail By Sls] * [AP Style Sls Units]))~
17-l	AP Style - In Transit Inv Amt	In Transit Inventory Amount for all DCs. Calculated as In transit units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([AP Style In Transit Inv Amt]) {~}
17-m	AP Style - In Transit Inv Units	In transit total units for all DCs.	Sum([AP Style In Transit Inv Units]) {~}
17-n	AP Style - Initial Retail By Sls	AP initial retail price weighted by sales.	Sum(([AP Style Wtd Init Retail By Sls] * [AP Style Sls Units]))~ / Sum([AP Style Sls Units])~
17-o	AP Style - Item Cost by Sls (Domestic)	Item cost weighted by sales.	Sum(([AP Style Wtd Item Cost By Sls] * [AP Style Sls Units]))~<Orig Ind = D> / Sum([AP Style Sls Units])~<Orig Ind = D>
17-p	AP Style - MD BOH	Beginning on hand of markdown inventory units.	Sum([AP Style Mkdn Boh])
17-q	AP Style - MD BOH $	Beginning on hand of markdown inventory dollar amount.	Sum(([AP Style Wtd Cur Retail By Boh] * [AP Style Mkdn Boh]))

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
17-r	AP Style - MD Sls $	Dollar amount associated with MD Sales Units {If Derived for Optimized History: If Retail < Initial Retail, then = Sales $, else 0}.	Sum([AP Style Mkdn Sls Amt])
17-s	AP Style - MD Sls Units	Any unit sold where Current Inventory Price is < Original Retail price {If Derived for Optimized History: If Retail < Initial Retail, then = Sales U, else 0}.	Sum([AP Style Mkdn Sls Units])
17-t	AP Style - Perm MD $	Permanent markdown dollars at style level.	Sum([AP Style Perm Md Amt])
17-u	AP Style - Perm MD %	Permanent markdown percentage at style level.	[AP Style - Perm MD $] / [AP Style Sls $]
17-v	AP Style - POS BOH	Point of sale beginning on hand inventory units.	Sum([AP Style Pos Boh])
17-w	AP Style - POS BOH $	Point of sale beginning on hand inventory dollar amount.	Sum(([AP Style Wtd Cur Retail By Boh] * [AP Style Pos Boh]))
17-x	AP Style - POS MD $	Point of sale markdown dollars.	Sum([AP Style Pos Md Amt])
17-y	AP Style - POS MD %	Point of sale markdown percent.	[AP Style - POS MD $] / [AP Style Sls $]
17-z	AP Style - POS Sls $	Dollar amount associated with POS Sales Units.	Sum([AP Style Pos Sls Amt])
17-aa	AP Style - POS Sls Units	Any unit sold where Sales Price < Original Retail Price.	Sum([AP Style Pos Sls Units])
17-bb	AP Style - Reg BOH	Beginning on hand inventory units.	([AP Style BOH Units (Sum Subtotal)] - [AP Style - MD BOH]) - [AP Style - POS BOH]
17-cc	AP Style - Reg BOH $	Beginning on hand inventory dollar amount.	([AP Style - BOH $] - [AP Style - MD BOH $]) - [AP Style - POS BOH $]
17-dd	AP Style - Reg Sls $	Sales dollars.	([AP Style Sls $] - [AP Style - MD Sls $]) - [AP Style - POS Sls $]
17-ee	AP Style - Reg Sls Units	Sales units.	([AP Style Sls Units] - [AP Style - MD Sls Units]) - [AP Style - POS Sls Units]
17-ff	AP Style - Total MD $	Total markdown dollars.	[AP Style - Perm MD $] ~ [AP Style - POS MD $]
17-gg	AP Style - Total MD %	Total markdown percent.	([AP Style - Total MD $] / Sum(([AP Style Wtd Cur Retail By Sls] * [AP Style Sls Units])) {~})
22-a	AP Style BOH Units (Sum Subtotal)	Beginning on hand units (sum of subtotals).	Sum([AP Style BOH])
23-a	AP Style EOH $ (LTD)	Life to date end on hand inventory dollar amount.	[AP Style Rcpt $ (Cum.)] - [AP Style Sls $ (Cum.)]
23-b	AP Style EOH Units (LTD)	Life to date end on hand inventory units.	[AP Style Rcpt Units (Cum.)] - [AP Style Sls Units (Cum.)]

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
22-b	AP Style EOH Units (Sum Subtotal)	End on hand inventory units.	Sum([AP Style EOH])
18-a	AP Style Rcpt $	Receipt dollars.	Sum([AP Style Rcpt Amt])
23-c	AP Style Rcpt $ (Cum.)	Cumulative Sum of Receipt dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Style Rcpt $])
19-a	AP Style Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([AP Style Rcpt Units])
23-d	AP Style Rcpt Units (Cum.)	Cumulative Sum of metric from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Style Rcpt Units])
21-b	AP Style Rcpt Units / # of Stores	Store receipts per store from Distros and Transfers (new data feed).	[AP Style Rcpt Units] / [AP Style - # of Stores with Inv]
20-a	AP Style Sls $	Sales dollars.	Sum([AP Style Sls Amt])
23-e	AP Style Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Style Sls $])
24-a	AP Style Sls Units	Sales units.	Sum([AP Style Sls Units])
23-f	AP Style Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([AP Style Sls Units])
22-c	AP Style Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	Sum([AP Style Sls Units])
23-g	AP Style ST % (Cum.)	Allocation plan sell through percent.	[AP Style Sls Units (Cum.)] / [AP Style Rcpt Units (Cum.)]
22-d	AP Style ST % (Cum.)	Sell through percent at style level.	[AP Style Sls Units (Cum.) FV] / ([AP Style Sls Units (Cum.) FV] ~ [AP Style EOH Units (Sum Subtotal)])
23-h	AP Style ST % (Cum.)	Sell through percent.	[AP Style Sls Units (Cum.)] / [AP Style Rcpt Units (Cum.)]
22-e	AP Style ST % (Cum.)	Sell through percent - using FV metrics.	[AP Style Sls Units (Cum.) FV] / [AP Style Rcpt Units (Cum.) FV]
26-a	Budgeted Charge Pct	Maximum budgeted charge percent.	Max([Company Budget Charge Pct])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
26-b	Budgeted Freight Pct	Maximum budgeted freight charge percent.	Max([Company Budget Freight Pct])
26-c	Budgeted Perm MD $	Permanent markdown dollars.	Sum([Company Budget P4 Mkdn Amt])
25-a	Budgeted Perm MD %	Permanent markdown percent.	[Budgeted Perm MD $] / [Store Budget Sls $]
26-d	Budgeted POS MD $	Point of sale markdown dollars.	Sum([Company Budget P2 Mkdn Amt])
25-b	Budgeted POS MD %	Point of sale markdown percent.	[Budgeted POS MD $] / [Store Budget Sls $]
26-e	Budgeted Shrink Pct	Budgeted shrink percentage.	Max([Company Budget Shrink Pct])
26-f	Budgeted Total MD $	Budgeted total markdown dollars.	[Budgeted Perm MD $] ~ [Budgeted POS MD $]
25-c	Budgeted Total MD %	Budgeted total markdown percent.	[Budgeted Total MD $] / [Store Budget Sls $]
30-a	DC Rcpt $	Distribution center receipt dollars.	Sum([DC Style Rcpt Amt])
31-a	DC Rcpt $	Store receipt dollars from Distros and Transfers (new data feed).	Sum([DC Color Rcpt Amt])
30-b	DC Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([DC Style Rcpt Units])
31-b	DC Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([DC Color Rcpt Units])
29-k	LSP Color - # of Colors with Inv	Number of colors available with inventory.	Count<Distinct>
33-a	LSP Color - # of Stores with Inv	Number of stores with inventory on hand.	Count<Distinct>
29-l	LSP Color - # of Styles with Inv	Number of colors available with inventory.	Count<Distinct>
29-m	LSP Color - # of Weeks	Number of weeks merchandise is on sale.	Max([LSP Color Num Selling Weeks])
32-a	LSP Color - ADJ GP %	Adjusted gross profit % margin by budgeted shrink percentage.	([LSP Color - ADJ IMU] - ([LSP Color - Total MD %] * (1 - [LSP Color - ADJ IMU]))) - (([Budgeted Charge Pct] ~ [Budgeted Shrink Pct]) ~ ([Budgeted Freight Pct] * (1 - [LSP Color - ADJ IMU])))
32-b	LSP Color - ADJ IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	(([LSP Color - Current Retail by Sls(Domestic)] - [LSP Color - Item Cost by Sls(Domestic)]) * 0.86) / [LSP Color - Current Retail by Sls(Domestic)]
32-c	LSP Color - ASR	Average Selling Retail price.	[LSP Color Sls $] / [LSP Color Sls Units]
32-d	LSP Color - AUC	Weekly WAC from sales file, for aggregations, weighted average based on EOH is used .	Sum(([LSP Color Sls Units] * [LSP Color Wtd Item Cost By Sls]))~ / Sum([LSP Color Sls Units])~

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
32-e	LSP Color - BOH	Beginning on hand inventory units.	Sum([LSP Color BOH])
32-f	LSP Color - BOH $	Beginning on hand inventory dollar amount.	Sum(([LSP Color Wtd Cur Retail By Boh] * [LSP Color BOH]))
32-g	LSP Color - Current Retail by Sls (Domestic)	Current Retail Price weighted by sales.	Sum(([LSP Color Wtd Cur Retail By Sls] * [LSP Color Sls Units]))~<Orig Ind = D> / Sum([LSP Color Sls Units])~<Orig Ind = D>
32-h	LSP Color - DC EOH Amt	DC EOH amount across all DC. Calculated as DC EOH units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([LSP Color Dc Eoh Amt]) {~}
32-i	LSP Color - DC EOH Units	DC EOH units across all DCs.	Sum([LSP Color Dc Eoh Units]) {~}
32-j	LSP Color - GM $	Gross Margin dollar amount.	ApplySimple("Case when #0 > 0 then #1 Else #2 End", [LSP Color - Actual Flag], [LSP Color - GM $ mapped], ([LSP Color - GP %] * [LSP Color Sls $]))
32-k	LSP Color - GP %	Gross Profit percent.	([LSP Color - IMU] - ([LSP Color - Total MD %] * (1 - [LSP Color - IMU]))) - ((([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ ([Budgeted Freight Pct] * (1 - [LSP Color - IMU])))
32-l	LSP Color - IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum((([LSP Color Wtd Init Retail By Sls] * [LSP Color Sls Units]) - ([LSP Color Wtd Item Cost By Sls] * [LSP Color Sls Units])))~ / Sum((([LSP Color Wtd Init Retail By Sls] * [LSP Color Sls Units]))~
32-m	LSP Color - In Transit Inv Amt	In Transit Inventory Amount for all DCs. Calculated as In transit units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([LSP Color In Transit Inv Amt]) {~}
32-n	LSP Color - In Transit Inv Units	In transit total units for all DCs.	Sum([LSP Color In Transit Inv Units]) {~}
32-o	LSP Color - Initial Retail By Sls	LSP initial retail price weighted by sales	Sum(([LSP Color Wtd Init Retail By Sls] * [LSP Color Sls Units]))~ / Sum([LSP Color Sls Units])~
32-p	LSP Color - Item Cost by Sls (Domestic)	Item cost weighted by sales.	Sum(([LSP Color Wtd Item Cost By Sls] * [LSP Color Sls Units]))~<Orig Ind = D> / Sum([LSP Color Sls Units])~<Orig Ind = D>
32-q	LSP Color - MD BOH	Markdown beginning on hand inventory.	Sum([LSP Color Mkdn Boh])
32-r	LSP Color - MD BOH $	Markdown beginning on hand dollar amount.	Sum(([LSP Color Wtd Cur Retail By Boh] * [LSP Color Mkdn Boh]))

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
32-s	LSP Color - MD Sls $	Dollar amount associated with MD Sales Units {If Derived for Optimized History: If Retail < Initial Retail, then = Sales $, else 0}.	Sum([LSP Color Mkdn Sls Amt])
32-t	LSP Color - MD Sls Units	Any unit sold where Current Inventory Price is < Original Retail price {If Derived for Optimized History: If Retail < Initial Retail, then = Sales U, else 0}.	Sum([LSP Color Mkdn Sls Units])
32-u	LSP Color - Perm MD $	Permanent markdown dollars.	Sum([LSP Color Perm Md Amt])
32-v	LSP Color - POS BOH	Point of sale beginning on hand inventory units.	Sum([LSP Color Pos Boh])
32-w	LSP Color - POS BOH $	Point of sale beginning on hand inventory dollar amount.	Sum(([LSP Color Wtd Init Retail By Boh] * [LSP Color Pos Boh]))
32-x	LSP Color - POS MD $	Point of sale markdown dollars.	Sum([LSP Color Pos Md Amt])
32-y	LSP Color - POS Sls $	Dollar amount associated with POS Sales Units.	Sum([LSP Color Pos Sls Amt])
32-z	LSP Color - POS Sls Units	Any unit sold where Sales Price < Original Retail Price.	Sum([LSP Color Pos Sls Units])
32-aa	LSP Color - Reg BOH	Beginning on hand inventory units.	([LSP Color BOH Units (Sum Subtotal)] - [LSP Color - MD BOH]) - [LSP Color - POS BOH]
32-bb	LSP Color - Reg BOH $	Beginning on hand inventory dollar amount.	([LSP Color - BOH $] - [LSP Color - MD BOH $]) - [LSP Color - POS BOH $]
32-cc	LSP Color - Reg Sls $	Sales dollar amount.	([LSP Color Sls $] - [LSP Color - MD Sls $]) - [LSP Color - POS Sls $]
32-dd	LSP Color - Reg Sls Units	Sales units.	([LSP Color Sls Units] - [LSP Color - MD Sls Units]) - [LSP Color - POS Sls Units]
32-ee	LSP Color - Total MD $	Total markdown dollars.	[LSP Color - Perm MD $] ~ [LSP Color - POS MD $]
32-ff	LSP Color - Total MD %	markdown percent.	([LSP Color - Total MD $] / Sum(([LSP Color Wtd Cur Retail By Sls] * [LSP Color Sls Units])) {~})
34-a	LSP Color BOH Units (Sum Subtotal)	Beginning on hand inventory units.	Sum([LSP Color BOH])
35-a	LSP Color EOH $ (LTD)	Life to date end on hand dollar amount.	[LSP Color Rcpt $ (Cum.)] - [LSP Color Sls $ (Cum.)]
35-b	LSP Color EOH Units (LTD)	Life to date end on hand inventory units.	[LSP Color Rcpt Units (Cum.)] - [LSP Color Sls Units (Cum.)]
34-b	LSP Color EOH Units (Sum Subtotal)	End on hand inventory units.	Sum([LSP Color EOH])
36-a	LSP Color Rcpt $	Receipt dollars.	Sum([LSP Color Rcpt Amt])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
35-c	LSP Color Rcpt $ (Cum.)	Cumulative Sum of Receipt dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	RunningSum<SortBy = [Week Period]>([LSP Color Rcpt $])
37-a	LSP Color Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([LSP Color Rcpt Units])
35-d	LSP Color Rcpt Units (Cum.)	Cumulative Sum of metric from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	RunningSum<SortBy = [Week Period]>([LSP Color Rcpt Units])
33-b	LSP Color Rcpt Units / # of Stores	Receipts units per store.	[LSP Color Rcpt Units] / [LSP Color - # of Stores with Inv]
38-a	LSP Color Sls $	Sales dollars.	Sum([LSP Color Sls Amt])
35-e	LSP Color Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	RunningSum<SortBy = [Week Period]>([LSP Color Sls $])
39-a	LSP Color Sls Units	Sales units.	Sum([LSP Color Sls Units])
35-f	LSP Color Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	RunningSum<SortBy = [Week Period]>([LSP Color Sls Units])
34-c	LSP Color Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	Sum([LSP Color Sls Units])
35-g	LSP Color ST % (Cum.)	Sell through percent at color level (cumulative).	[LSP Color Sls Units (Cum.)] / [LSP Color Rcpt Units (Cum.)]
34-e	LSP Color ST % (Cum.)	Sell through percent at color level (sum of subtotal).	[LSP Color Sls Units (Cum.) FV] / ([LSP Color Sls Units (Cum.) FV] ~ [LSP Color EOH Units (Sum Subtotal)])
35-h	LSP Color ST % (Cum.)	Sell through percent at color level (cumulative).	[LSP Color Sls Units (Cum.)] / [LSP Color Rcpt Units (Cum.)]
34-d	LSP Color ST % (Cum.)	Sell through percent.	[LSP Color Sls Units (Cum.) FV] / [LSP Color Rcpt Units (Cum.) FV]
1-a	Max Out Date	Latest out date.	Max([Oos Date])
1-b	Max Out Date - extension	The latest outdate when looking at a group of items with different out dates.	Max([Oos Date - not extended])
1-c	Min Out Date	Earliest out date.	Min([Oos Date])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
1-d	Min Out Date - extension	Earliest out date.	Min([Oos Date - not extended])
40-a	Need Color - ASR	Average Selling Retail price.	[Need Color Sls $] / [Need Color Sls Units]
40-b	Need Color - BOH $	Beginning on hand dollars.	Sum(([Need Color Wtd Cur Retail By Boh] * [Need Color BOH]))
40-c	Need Color - DC EOH Amt	DC EOH amount across all DC. Calculated as DC EOH units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([Need Color Dc Eoh Amt]) {~}
40-d	Need Color - DC EOH Units	DC EOH units across all DCs.	Sum([Need Color Dc Eoh Units]) {~}
40-e	Need Color - GM $	Gross margin dollars.	ApplySimple("Case when #0 > 0 then #1 Else #2 End", [Need Color - Actual Flag], [Need Color - GM $ mapped], ([Need Color - GP %] * [Need Color Sls $]))
40-f	Need Color - GP %	Gross profit percentage at color level.	([Need Color - IMU] - ([Need Color - Total MD %] * (1 - [Need Color - IMU]))) - ((([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ [Budgeted Freight Pct] * (1 - [Need Color - IMU])))
40-g	Need Color - IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum(((([Need Color Wtd Init Retail By Sls] * [Need Color Sls Units]) - ([Need Color Wtd Item Cost By Sls] * [Need Color Sls Units]))~ / Sum(([Need Color Wtd Init Retail By Sls] * [Need Color Sls Units]))~
40-h	Need Color - In Transit Inv Amt	In Transit Inventory Amount for all DCs. Calculated as In transit units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([Need Color In Transit Inv Amt]) {~}
40-i	Need Color - In Transit Inv Units	In transit total units for all DCs.	Sum([Need Color In Transit Inv Units]) {~}
40-j	Need Color - Perm MD $	Permanent markdown dollars.	Sum([Need Color Perm Md Amt])
40-k	Need Color - POS MD $	Point of sale markdown dollars.	Sum([Need Color Pos Md Amt])
40-l	Need Color - Total MD $	Total markdown dollars.	[Need Color - Perm MD $] ~ [Need Color - POS MD $]
40-m	Need Color - Total MD %	Markdown percent.	([Need Color - Total MD $] / Sum(([Need Color Wtd Cur Retail By Sls] * [Need Color Sls Units])) {~})
42-a	Need Color - WOS	Basic weeks of supply using inventory divided by last week of sales. One Plus and RW use 4 week average.	[Need Color EOH Units (LTD)] / [Need Color Sls Units]
41-a	Need Color BOH Units (Sum Subtotal)	Beginning on hand inventory units.	Sum([Need Color BOH])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
42-b	Need Color EOH Units (LTD)	End on hand inventory units.	[Need Color Rcpt Units (Cum.)] - [Need Color Sls Units (Cum.)]
41-b	Need Color EOH Units (Sum Subtotal)	End on hand inventory units.	Sum([Need Color EOH])
43-a	Need Color Rcpt $	Receipt dollars.	Sum([Need Color Rcpt Amt])
42-c	Need Color Rcpt $ (Cum.)	Cumulative Sum of Receipt dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Color Rcpt $])
44-a	Need Color Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([Need Color Rcpt Units])
42-d	Need Color Rcpt Units (Cum.)	Cumulative Sum of metric from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Color Rcpt Units])
45-a	Need Color Sls $	Sales dollars.	Sum([Need Color Sls Amt])
42-e	Need Color Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Color Sls $])
46-a	Need Color Sls Units	Sales units	Sum([Need Color Sls Units])
42-f	Need Color Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Color Sls Units])
41-c	Need Color Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	Sum([Need Color Sls Units])
41-d	Need Color ST % (Cum.)	Sell through percent.	[Need Color Sls Units (Cum.) FV] / ([Need Color Sls Units (Cum.) FV] ~ [Need Color EOH Units (Sum Subtotal)])
42-g	Need Color ST % (Cum.)	Need color sell through percent.	[Need Color Sls Units (Cum.)] / [Need Color Rcpt Units (Cum.)]
42-h	Need Color ST % (Cum.)	Sell through percent.	[Need Color Sls Units (Cum.)] / [Need Color Rcpt Units (Cum.)]
41-e	Need Color ST % (Cum.)	Sell through percent - using FV metrics.	[Need Color Sls Units (Cum.) FV] / [Need Color Rcpt Units (Cum.) FV]
47-a	Need Style - ASR	Average Selling Retail price.	[Need Style Sls $] / [Need Style Sls Units]

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
47-b	Need Style - BOH	Beginning on hand inventory - used for MS filter only.	Sum([Need Style BOH])
47-c	Need Style - BOH $	Beginning on hand dollars.	Sum(([Need Style Wtd Cur Retail By Boh] * [Need Style BOH]))
47-d	Need Style - DC EOH Amt	DC EOH amount across all DC. Calculated as DC EOH units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([Need Style Dc Eoh Amt]) {~}
47-e	Need Style - DC EOH Units	DC EOH units across all DCs.	Sum([Need Style Dc Eoh Units]) {~}
47-f	Need Style - GM $	Gross margin dollars.	ApplySimple("Case when #0 > 0 then #1 Else #2 End", [Need Style - Actual Flag], [Need Style - GM $ mapped], ([Need Style - GP %] * [Need Style Sls $]))
47-g	Need Style - GP %	Gross percent percentage at style level.	([Need Style - IMU] - ([Need Style - Total MD %] * (1 - [Need Style - IMU]))) - ((([Budgeted Shrink Pct] ~ [Budgeted Charge Pct]) ~ ([Budgeted Freight Pct] * (1 - [Need Style - IMU])))
47-h	Need Style - IMU	Initial Markup (or the markup on the goods when they first come in). This is common across retailers (Initial Retail - Initial cost) / Initial Retail.	Sum(((([Need Style Wtd Init Retail By Sls] * [Need Style Sls Units]) - ([Need Style Wtd Item Cost By Sls] * [Need Style Sls Units]))~ / Sum((([Need Style Wtd Init Retail By Sls] * [Need Style Sls Units]))~
47-i	Need Style - In Transit Inv Amt	In Transit Inventory Amount for all DCs. Calculated as In transit units * current permanent price or initial setup retail price if permanent price is not available for the period.	Sum([Need Style In Transit Inv Amt]) {~}
47-j	Need Style - In Transit Inv Units	In transit total units for all DCs.	Sum([Need Style In Transit Inv Units]) {~}
47-k	Need Style - Perm MD $	Permanent markdown dollars.	Sum([Need Style Perm Md Amt])
47-l	Need Style - POS MD $	Point of sale markdown dollars at style level.	Sum([Need Style Pos Md Amt])
47-m	Need Style - Total MD $	Total markdown dollars.	[Need Style - Perm MD $] ~ [Need Style - POS MD $]
47-n	Need Style - Total MD %	Total markdown percent.	([Need Style - Total MD $] / Sum((([Need Style Wtd Cur Retail By Sls] * [Need Style Sls Units])) {~})
52-a	Need Style - WOS	Basic weeks of supply using inventory divided by last week of sales. One Plus and RW use 4 week average.	[Need Style EOH Units (LTD)] / [Need Style Sls Units]
51-a	Need Style BOH Units (Sum Subtotal)	Beginning on hand inventory units.	Sum([Need Style BOH])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
52-b	Need Style EOH Units (LTD)	Life to date ending on hand inventory units .	[Need Style Rcpt Units (Cum.)] - [Need Style Sls Units (Cum.)]
51-b	Need Style EOH Units (Sum Subtotal)	End on hand inventory units.	Sum([Need Style EOH])
48-a	Need Style Rcpt $	Receipt dollars.	Sum([Need Style Rcpt Amt])
52-c	Need Style Rcpt $ (Cum.)	Cumulative Sum of Receipt dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Style Rcpt $])
49-a	Need Style Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([Need Style Rcpt Units])
52-d	Need Style Rcpt Units (Cum.)	Cumulative Sum of metric from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Style Rcpt Units])
50-a	Need Style Sls $	Sales dollars.	Sum([Need Style Sls Amt])
52-e	Need Style Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Style Sls $])
53-a	Need Style Sls Units	Sales units.	Sum([Need Style Sls Units])
52-f	Need Style Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([Need Style Sls Units])
51-c	Need Style Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used).	Sum([Need Style Sls Units])
51-d	Need Style ST % (Cum.)	Sell through percent cumulative at sum subtotal level.	[Need Style Sls Units (Cum.) FV] / ([Need Style Sls Units (Cum.) FV] ~ [Need Style EOH Units (Sum Subtotal)])
52-g	Need Style ST % (Cum.)	Sell through percent cumulative at cumulative level.	[Need Style Sls Units (Cum.)] / [Need Style Rcpt Units (Cum.)]
52-h	Need Style ST % (Cum.)	Sell through percent.	[Need Style Sls Units (Cum.)] / [Need Style Rcpt Units (Cum.)]
51-e	Need Style ST % (Cum.)	Sell through percent - using FV metrics.	[Need Style Sls Units (Cum.) FV] / [Need Style Rcpt Units (Cum.) FV]
54-a	OPT Average Selling Retail	Optimized history average selling retail price.	[OPT Sls $] / [OPT Sls Units]

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
54-b	OPT Average Selling Retail (Last Year)	Optimized history average selling retail price this period last year.	Sum([Optimized History Sls Amt])~[Week Period - Last Year]│ / Sum([Optimized History Sls Units])~[Week Period - Last Year]│
54-c	OPT BOH $	Optimized history beginning on hand inventory dollars.	Sum(([Optimized History Wtd Cur Retail By Boh] * [Optimized History BOH]))
54-d	OPT BOH $ (Last Year)	Optimized history beginning on hand inventory dollars - last year.	Sum(([Optimized History Wtd Cur Retail By Boh] * [Optimized History BOH]))
55-a	OPT BOH Units (Sum Subtotal)	Optimized history beginning on hand inventory units.	Sum([Optimized History BOH])
55-b	OPT BOH Units (Sum Subtotal) (Last Year)	Optimized history beginning on hand inventory units this period last year.	Sum([Optimized History BOH])
56-a	OPT EOH Units (Last Subtotal)	End on hand inventory units.	Sum([Optimized History EOH])
55-c	OPT EOH Units (Sum Subtotal)	Optimized history ending on hand inventory units.	Sum([Optimized History EOH])
54-e	OPT GP $	Optimized history gross profit dollars.	[OPT GP %] * [OPT Sls $]
54-f	OPT GP $ (Last Year)	Optimized history gross profit dollars this period last year.	Sum([Optimized History Sls Amt])~[Week Period - Last Year]│ * (((Sum(((NullToZero([Optimized History Wtd Init Retail By Sls]) - NullToZero([Optimized History Wtd Item Cost By Sls])) * [Optimized History Sls Units]))~[Week Period - Last Year]│ * Sum(([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]))~[Week Period - Last Year]│) - (Sum((([Optimized History Wtd Item Cost By Sls] * [Optimized History Sls Units]))~[Week Period - Last Year]│ * Sum((NullToZero([Optimized History Promo Md Amt]) ~ NullToZero([Optimized History Perm Md Amt])))~[Week Period - Last Year]│)) / Power(Sum(([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]))~[Week Period - Last Year]│,2))
54-g	OPT GP %	Optimized history gross profit percent.	[OPT IMU] - ([OPT Total MD %] * (1 - [OPT IMU]))

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
54-h	OPT GP % (Last Year)	Optimized history percent this period last year.	((Sum((((NullToZero([Optimized History Wtd Init Retail By Sls]) - NullToZero([Optimized History Wtd Item Cost By Sls])) * [Optimized History Sls Units]))~~[Week Period - Last Year]| * Sum(([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]))~[Week Period - Last Year]|) - (Sum((([Optimized History Wtd Item Cost By Sls] * [Optimized History Sls Units]))~~[Week Period - Last Year]| * Sum((NullToZero([Optimized History Promo Md Amt]) ~ NullToZero([Optimized History Perm Md Amt])))~~[Week Period - Last Year]|)) / Power(Sum((([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]))~~[Week Period - Last Year]|,2)
54-i	OPT IMU	Initial Retail / Cost is Weighted Average Cost from sales file.	Sum(((([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]) - ([Optimized History Wtd Item Cost By Sls] * [Optimized History Sls Units])))~ / Sum(([Optimized History Wtd Init Retail By Sls] * [Optimized History Sls Units]))~
54-j	OPT Perm MD $	Optimized history permanent markdown dollars.	Sum([Optimized History Perm Md Amt])
54-k	OPT POS MD $	Optimized history point of sale markdown dollars.	Sum([Optimized History Promo Md Amt])
57-a	OPT Rcpt $	Optimized history receipt dollars.	Sum([Optimized History Rcpt Amt])
57-b	OPT Rcpt $ (Last Year)	Optimized history receipt dollars this period last year.	Sum([Optimized History Rcpt Amt])
58-a	OPT Rcpt Units	Store receipts from Distros and Transfers (new data feed).	Sum([Optimized History Rcpt Units])
58-b	OPT Rcpt Units (Last Year)	Store receipts from Distros and Transfers (new data feed).	Sum([Optimized History Rcpt Units])
59-a	OPT Sls $	Optimized history sales dollars.	Sum([Optimized History Sls Amt])
56-b	OPT Sls $ (Cum.)	Cumulative Sum of Sales dollars from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([OPT Sls $])
59-b	OPT Sls $ (Last Year)	Optimized history sales dollars this period last year.	Sum([Optimized History Sls Amt])
60-a	OPT Sls Units	Optimized history sales units.	Sum([Optimized History Sls Units])

Table 6–4   (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
56-c	OPT Sls Units (Cum.)	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	RunningSum<SortBy = [Week Period]>([OPT Sls Units])
55-d	OPT Sls Units (Cum.) FV	Cumulative Sum of Sales units from beginning of time until week to date. (Will effectively be the same as LTD if no time filter is used.)	Sum([Optimized History Sls Units])
60-b	OPT Sls Units (Last Year)	Optimized history sales units this period last year.	Sum([Optimized History Sls Units])
55-e	OPT ST % (Cum.)	Optimized history sell through percent cumulative.	[OPT Sls Units (Cum.) FV] / ([OPT Sls Units (Cum.) FV] ~ [OPT EOH Units (Sum Subtotal)])
56-d	OPT ST % (Cum.)	Optimized history sell through percent from subtotal.	[OPT Sls Units (Cum.)] / ([OPT Sls Units (Cum.)] ~ [OPT EOH Units (Last Subtotal)])
56-e	OPT ST % (Cum.)	Sell through percent.	[OPT Sls Units (Cum.)] / [OPT Rcpt Units (Cum.)]
55-f	OPT ST % (Cum.)	Sell through percent - using FV metrics.	[OPT Sls Units (Cum.) FV] / [OPT Rcpt Units (Cum.) FV]
54-l	OPT Total MD $	Optimized history total markdown dollars.	[OPT POS MD $] ~ [OPT Perm MD $]
54-m	OPT Total MD $ (Last Year)	Optimized history total markdown dollars this period last year.	((Sum([Optimized History Promo Md Amt]) {~} + Sum([Optimized History Perm Md Amt]) {~}) / Sum(([Optimized History Wtd Cur Retail By Sls] * [Optimized History Sls Units])) {~})
54-n	OPT Total MD %	Optimized history markdown percent.	((Sum([Optimized History Promo Md Amt]) {~} + Sum([Optimized History Perm Md Amt]) {~}) / Sum(([Optimized History Wtd Cur Retail By Sls] * [Optimized History Sls Units])) {~})
61-a	Pack Opt Average Cost	Optimized average cost for a pack.	Avg([Pack Opt Item Setup Cost])
61-b	Pack Opt Average Retail Price	Optimized average retail price for a pack.	Avg([Pack Opt Item Setup Retail])
61-c	Pack Opt IMU%	Initial Markup (or the markup on the goods when they first come in). This is common across retailers (Initial Retail - Initial cost) / Initial Retail.	Sum(((([Pack Opt Item Setup Retail] - [Pack Opt Item Setup Cost]) * ([Pack Opt Num Packs] * [Pack Opt Inside Units])))~ / Sum((([Pack Opt Item Setup Retail] * ([Pack Opt Num Packs] * [Pack Opt Inside Units])))~
61-d	Pack Opt Inside Units	Optimized inside units for a pack.	Avg([Pack Opt Inside Units])
61-e	Pack Opt Item Setup Cost	Item setup cost.	Avg([Pack Opt Item Setup Cost])
61-f	Pack Opt Item Setup Retail	Item setup retail price.	Avg([Pack Opt Item Setup Retail])
61-g	Pack Opt Num Packs	Optimized number of packs.	Sum([Pack Opt Num Packs])

Table 6–4 (Cont.) Metrics, Metric Descriptions, and Metric Calculations

No.	Metric Name	Metric Description	Metric Calculation
61-h	Pack Opt Total Cost	Optimized total cost.	[Pack Opt Total Receipt Units] * [Pack Opt Item Setup Cost]
61-i	Pack Opt Total Receipt Units	Optimized total receipt units for a pack.	Sum(([Pack Opt Num Packs] * [Pack Opt Inside Units]))
61-j	Pack Opt Total Retail	Optimized retail price for a pack.	[Pack Opt Total Receipt Units] * [Pack Opt Item Setup Retail]
27-a	Store Budget Rcpt $	Store budget receipt dollars.	Sum([Store Budget Rcpt Amt])
28-a	Store Budget Sls $	Store budget sales dollars.	Sum([Store Budget Sales Amt])

7

Plan Updater

The chapter contains the following:

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 7–1   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

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 8, "Properties File Configuration".

planupdater.properties

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

Table 7–2   planupdater.properties

Property Name	Property Description	Property Value
planupdater.maxthreads	The maximum number of threads to process plan and item tasks.	Value: Integer Default = 10
planupdater.planmaxseconds	The maximum number of seconds to wait for all plan tasks for all selected plans to complete.	Value: Integer Default = 300
planupdater.itemmaxseconds	The maximum number of seconds to wait for all item tasks for all selected plans to complete.	Value: Integer Default = 120
planupdater.testmode	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
planupdater.verbose	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 (planupdater.sh) is located in <install directory>/modules/tools/bin folder.

The following command line parameters are available:

Table 7–3   Plan Updater Command Line Parameters

Parameter	Description
-operations	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.
	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).
	planPrice. Checks for Pricing Plan changes that affect Plan.
	planAuto. Auto-populates a plan with basic items.
	planRdm. Sends a request to update RDM. This is an asynchronous call that does not prevent Plan from completing its task.
	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.)
	itemSegment. Updates all segments for a particular item set ID.
	These tasks are executed in the following order: 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.

Configuration Settings

The following settings are required for scaling Plan Updater.

Increase the WebLogic default thread count. The default thread count on each Plan server should be set to approximately 35 (5 provides an extra margin). The thread count on each CE server should be 40.

To set the default thread count, complete the following steps:

1. Click the server name under servers on the admin console.

2. Click Show in Advanced Options at the bottom of the right frame.

3. Click Configure Execute Queues.

4. Select weblogic.kernal.Default. Change the thread count parameter to the appropriate number. The value should be greater than or equal to 25, which is the default value.

5. After setting the appropriate thread count on all the servers in the cluster, restart all servers.

Increase the number of Plan Updater threads. The Plan Updater thread count need to be increased to a number proportional to the WebLogic default thread count. Use the following formula to define this number: Plan Updater threads = n * (threadcnt - 5), where *n* is the number of Plan FE servers. (For example, if you set the WebLogic default thread count to 35, then, for 3 servers, the Plan Updater threads should be 3 * (35 -5) = 90.)

To increase the thread count, complete the following steps:

1. Identify the server being used to start the Plan Updater run.

2. Navigate to the <install_base> directory.

3. Open the file <install_base>/modules/tools/conf/planupdater.properties

4. Change the value of planupdater.maxthreads to 90.

Change the CE MDB Setting. The number of MDBs on each CE server should be set to 25.

Complete the following steps on the admin server:

1. **cd** *<install_base>/modules/Engine/lib*

2. **cp** *delphi.ear delphi.orig*

3. **mkdir** *temp*

4. **cp** *delphi.ear temp*

5. **cd** *temp*

6. **vi.jar -xvf** *delphi.ear*

7. **mkdir** *temp1*

8. **mv** *DelphiListerMDB.jar temp1*

9. **cd** *temp1*

10. **jar -xfv** *DelphiListenerMDB.jar*

11. **vi** *META-INF/weblogic-ejb-jar.xml*

12. Change the <max-beans-in-free-pool> and <initial-beans-in-free-pool> to 25.

    ```
    <pool>
    <max-beans-in-free-pool>25</max-beans-in-free-pool>
    <initial-beans-in-free-pool>25</initial-beans-in-free-pool>
    </pool>
    ```

13. **rm** *DelphiListenerMDB.jar*

14. **jar -cmf** *META-INF/MANIFEST.MF DelphiListenerMDB.jar* *

15. **mv** *DelphiListenerMDB.jar ../*

16. **rm -rf** *temp1*

17. **rm** *delphi.ear*

18. **jar -cmf** *META-INF/MANIFEST.MF delphi.ear* *

19. **mv** *delphi.ear ../*

20. From the WebLogic console on the admin server, delete the application delphi.

21. Deploy delphi.ear from <install_base>/modules/Engine/lib/ and target it to the CE cluster. Make sure that "Copy this application onto every target for me" is selected.

22. Verify that the deployment is complete by clicking the DelphiListenerMDB.jar under Applications. Click descriptors on the right frame and select weblogic-ejb-jar.xml to view your changes.

JDBC Connection Pool Settings. Refer to the following table for the correct range of values for the connection pool settings.

Table 7–4   JDBC Connection Pool Settings

Connection Pool	Minimum Value	Maximum Value
AnalyticalConnectionPool	1	50
AuditConnectionPool	1	50
AuditConnectionPool	1	10
BusinessConnectionPool	1	50
CommonConnectionPool	1	100
ForecastConnectionPool	1	50
HistoricalConnectionPool	1	50
RDMConnectionPool	1	35
RuntimeConnectionPool	1	10

To change the connection pool settings, complete the following steps:

1. Click JDBC in the admin console.

2. Click the name of the connection pool whose settings need to be changed.

3. Choose the Connections tab.

4. Set the Initial Capacity and Maximum Capacity text fields to the appropriate values. Refer to Table 7–4, " JDBC Connection Pool Settings".

5. You do not need to re-start the servers after changing these properties.

Schema Process Settings. The schema process setting on OAK SCHEMA needs to be set to a number proportional to the Plan Updater threads. To determine the process setting, use the formula (n * 3) + 100, where *n* is the number of Plan Updater threads. (For example, if there are 100 threads, then set the parameter to 400.)

To set the process setting, complete the following steps:

1. Locate init<SCHEMANAME>.ora file on the database server.

2. Set the property Processes = 400.

3. Bounce the database server.

Create the Storeset for the Plan Updater user.

To create a Storeset for the Plan Updater user, complete the following steps:

1. Login to the application as the Plan Updater user.

2. Select Storeset Management.

3. Select Action > New Storeset and click Apply.

4. Click New Storeset.

5. Select Action > NewSubset and click Apply.

6. Select the Stores tab.

7. Move a random store to the NewSubset List.

8. Click Done.

9. Log out of the application.

8

Properties File Configuration

The chapter contains the following:

■ "Delphi.properties File" on page 8-16

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.

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. It must be defined for each deployment.
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.

Property Name	Property Description
plan.default.plc.for.forecast	Use this property to set the default option for the *PLC Selection* drop-down list (on the *Like Item* tab). You can specify one of the following:

- regularPLC

- basicPLC

- tabularPLC

- nonshiftedTabularPLC

The default value is regularPLC. When you specify a value for this parameter, the PLC Selection drop-down list gets enabled on the Like Items tab.

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 if the original forecast fails. The default value is 43200 (12 hours).
plan.forecast.refresh.interval	The number of milliseconds Plan waits before refreshing the engine status response. The default value is 30000.
plan.lock.status.refresh.interval	The number of milliseconds Plan waits before refreshing the plan lock status response. The default value is 45000.

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.

Property Name	Property Description
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 runs again. The default value for this property is false. To prevent FNN data from being imported when the Plan is open, set the plan.supress.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. 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. If supress.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.
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. 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). 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.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.
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

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.AssortmentPlan Plugin interface.
plan.logic.planneditem.plugin.className	The Java class name of the client-specific PlannedItemPlugin module. This class must implement com.profitlogic.buying4p.logic.plugins.PlannedItem Plugin 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.FlowParam Plugin interface.

Property Name	Property Description
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.class Name	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.class Name	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	Property Description
plan.logic.notification.className	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	Property Description
plan.backend.processing.queue. connection.factory	The JNDI connection factory for the back-end processing queue.

Property Name	Property Description
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. credentials plan.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.RunForecasts.jndi_name plan.RunForecasts.home_class plan.RunForecasts.method_name	Defines the RunForecast back-end service for multiple items. 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.
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.

Calc Engine Forecast Properties

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

Property Name	Property Description
plan.user.forecast.parallel	Forecasts for multiple items can be triggered in three ways: 1. A user selects multiple styles in the Plan Worklist screen and runs Request Forecast on the items. 2. A user copies and pastes Buy Parameters for multiple items and then select Save. 3. A user edits Buy parameters for multiple items, then selects Save and then Edit. This property determines whether the forecasts are processed in sequence (a value of *false*) or in parallel (a value of *true* - the default). The parallel mode uses multiple threads and can cause the application to respond slowly to other user requests. The sequential mode takes more time to complete, but is more reliable and distributes system resources more equitably.
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. These 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	max = min * capping_factor
plan.packopt.agg.pmax.hiCost	The cost of exceeding the aggregate maximum.

Property Name	Property Description
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. Designed to overcome the problems that parallel pack opt has with stores that are receiving small deliveries.

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.forecast Length	The number of months to run a forecast if the out-of-stock date is not specified.
plan.plannedItem.deliveryFrequencies plan.plannedItem.deliveryFrequencyIds plan.plannedItem.deliveryFrequencyId. weekly plan.plannedItem.deliveryFrequency PeriodWeeks	The client-specified list of delivery frequencies and schedules set in Flow Parameters for a planned item.
plan.likeitem.style.store.weights.default plan.likeitem.color.store.weights.default	The defaults for the store weight in the Like Item grid. The value can be set to the level sequence number in the hierarchy. Style can only be assigned a value of 8. Color can only be assigned a value of 8 or 9.
plan.likeitem.hierarchy.fallbackLevel	The minimum level of the hierarchy that the user must enter to return a Like Item search.

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).

Property Name	Property Description
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).

Fetch Size Properties

This section describes database fetch size tuning.

Property Name	Property Description
plan.itemPlanAdapter.getweeklyStoreSegments.fetchsize	The database fetch size.

Cache Limit Sizes and Cache Management Properties

This section describes properties that set limitations on caches.

Property Name	Property Description
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.

Property Name	Property Description
planCalcParamsCache.getPercentTo Purge.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. These properties control the way forecasted data gets into the application.

Property Name	Property Description
plan.itemCacheMgr.suppress.cacheUse	Suppresses check ItemSegmentBean objects in the cache in order to measure performance curves vs. ITEM_LOCATION_PERIODS_TBL
reuse.style.calcdata.for.colors	Switch to make calc data retrieval for colors largely reuse values from calc data retrieved for earlier parent style.
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.

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).

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.

Search Level Properties

This section describes the search level properties.

Property Name	Property Description
highest.plan.search.level	The highest search level above planning level at which the client ID is unique across that level. If the client ID is not unique, then there may be duplicates in the search lists. If duplicates exist, then a user may choose the wrong entries. Client IDs should be unique within the parent level, so users can safely choose the immediate child of CHAIN. The default should be CHAIN, since this level always exists in a customer configuration.
lowest.plan.search level	The lowest search level is similar to the highest search level, with one exception: It sets the lower bounds of the search. This property takes the same type of values as highest.plan.search.level.

Strings in Planned_Item_Types_Tbl Properties

This section describes the strings for the item types.

Property Name	Property Description
plan.util.item.type.fashion plan.util.item.type.basic plan.util.item.type.VMI	The planned_item_types_tbl contains strings for Fashion, Basic, and VMI item types. These strings are referenced by the code and so should be distinct. A string must be assigned for each item type, even if a customer does not use a particular item type.
plan.util.item.type.default	The default type for the plan worklist table. This string should be one of the strings in planned_item_types_tbl.

Sliding Window Logic Properties

This section describes the sliding window logic properties.

Property Name	Property Description
plan.pricing.plan.sliding.window.enabled	This property is used to enable and disable the sliding window logic for the pricing plan. Values are true and false.
plan.pricing.plan.sliding.window.days	The number of days to use for the sliding window. The default value is 30. If the value for plan.pricing.plan.sliding.window.enabled is false, then this property is ignored.
plan.pricing.plan.sliding.window.start. offset.from.today	The offset from the current date to use for the beginning of the planned promotions sliding window. this property is for Internal Use only and should always be set to 0. If the value for plan.pricing.plan.sliding.window.enabled is false, then this property is ignored.

Plan Updater Properties

This section describes the plan updater properties.

Property Name	Property Description
plan.plan.library.planupdater.default	The default value for the PlanUpdater option found in the Plan Library screen. If the value of this propertiy is 0, then the PlanUpdater is turned off when a new plan is created. If the value of this property is 1, then the PlanUpdater 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.against. actual.oos	Indicates whether or not the flow dates should be validated against the actual OOS date in the merchandise hierarchy.
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	Indicates the types of items that can have a blank out-of-stock date. Can be blank. Basic items can have a blank out-of-stock date.
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 in the format MM/dd/yyyy. Used for testing only.

Forecast Configuration Properties

The section describes the forecast configuration properties.

Property Name	Property Description
plan.forecast.config.variables	Specifies the variable names to be used. Variables named here can be used by plan.forecast.config.weights.option.#.#
Variable Name	Defines the bean value associated with the variable.
lookup.startDate lookup.EndDate	User-defined variables specifiying the property path of the forecast Interpretor. Path must follow actual bean properties of the Forecast Interpretor.
plan.forecast.config.weights.style.options	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 #.# format. Note that 0.# is used for internal options and 1.# is used for external options.
plan.forecast.config.weights.color.options	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 #.# format. Note that 0.# is used for internal options and 1.# is used for external options.
plan.forecast.config.weights.option.0.keys	Defines internal option key. Currently not being used.
plan.forecast.config.weights.option.0.8	Defines the internal option value. Currently not being used.
plan.forecast.config.weights.option.0.9	Defines the internal option value. Currently not being used.
plan.forecast.config.weights.option.1.keys	Defines the keys used for the external options in a pipe delimited list. Variables used are defined in the variables property.
plan.forecast.config.weights.option.1.2	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 IR_GET_EXT_ STORE_WEIGHTS to calculate the external store weight passed to the Calc Engine.

Property Name	**Property Description**
plan.forecast.config.weights.option.1.3	Defines the external option value. See above for details.
plan.forecast.config.weights.option.1.4	Defines the external option value. See above for details.
plan.forecast.config.weights.option.1.5	Defines the external option value. See above for details.
plan.forecast.config.weights.option.1.6	Defines the external option value. See above for details.
plan.forecast.config.weights.option.1.7	Defines the external option value. See above for details.
plan.constrainedForecast.enabled	Set this parameter to "true" to enable the constrained forecasting in Plan. When this parameter is set to "true", the Need Type Owner field appears on the Like Item tab. When set to "false", the Need Type Owner field will not appear on the Like Item tab and users will no longer have the option to generate a constrained forecasts.
plan.max.constrained.sales.units	Use this parameter to set the maximum constrained sales units in the Store Quantities pop-up screen. The value for this parameter defaults to 10.
plan.highlight.stores.color	The color used to highlight the stores with Constrained Sales Units. The value for this parameter defaults to LightYellow.

Delphi.properties File

The delphi.properties file includes parameters that help set up the forecasting component (Delphi) of the Calc Engine specific to your business. Most of the parameters in this file have reasonable defaults set for the software. The parameters that do not have default values set, are included in the **Required Deployment** and **Required Model** sections. When configuring the application, ensure that the values in this section are set.

This file is located at the following location in the Place Installation folder:

```
<Plan_Installation>\config\Engine\
```

The following table lists the parameters in the delphi.properties file:

Property Name	Property Description
Required Deployment Settings	
engine.agorai.lib	Location, on the Calc Engine installation base, where the Agorai component (Optimization Engine) is installed.
delphi.rmi.port	Use this parameter to specify the port used by the RMI server in an interactive mode.
optimize.status.tbl	Use this parameter to specify the name of the database table name that stores the optimization status for each item and collection. The value defaults to *item_status_tbl*.
driver	Use this parameter to specify the Java CLASS driver that is used to communicate with the database. The value defaults to *jdbc.driver.OracleDriver*.
url	Use this parameter to specify the URL to connect to the database.
user	Use this parameter to specify the user name to connect to the database.
password	Use this parameter to specify the password (associated with the user name) to connect to the database.
Optional Batch Configuration	
batch.write.size	Number of items or collections to be included in a batch for write operations to the database.
chunk.tryLimit	Maximum number of times that any item will be tried for optimization, before evaluating that the item cannot be processed. Set the limit according to the optimal policy based on the simulations, and the value must be greater than 1.

Property Name	Property Description
chunk.sizes	Sequence of chunk sizes that are used to group items with retries. The value defaults to 1000;100;1.
	The sequence of chunk sizes relates to the progressive discovery of which items are causing chunks to fail. The optimal policy is to try large chunks first, then split the chunks into smaller pieces when retries are needed.
	When specifying the values, consider the following:
	■ Decreasing sequence of positive integers
	■ Length of the sequence equals the *chunk.trylimit*.
	■ Last number must be 1.
	■ Members must be separated by semi-colon.
	Choosing the chunk size is the key to good performance and the large chunk size must balance two factors, random memory access per worker and process limitations.
worker.lifetime	The time duration (in minutes) the processor must be allowed to run before it determines an infinite loop and terminates.
	The value defaults to 30 and must be more than 1 minute.
chunk.active	The time duration (in minutes) a chunk, after being terminated, can be reclaimed.
Required Model	
Lifecycle estimation parameters	
delphi.TLandingAlgorithmName	Use this parameter to specify the TLanding algorithm to be used. You can specify one of the following values:
	■ HMM – is the default value and indicates the use of Hidden Markov Model.
	■ SellThru – chooses the week in which the cumulative sales (from the start of history) as a percentage of all historical sales reaches the threshold defined by the *delphi.lifecycle.sellThruPercent* parameter.
	■ SimplifiedPLAN – the legacy algorithm used in previous versions.
delphi.lifecycle.useCorrectedSales	Use this parameter (Set to *True*) to specify the use of the corrected sales (de-seasoned, de-promoted, de-priced, and so on) in the TLanding algorithm calculations.
	The value defaults to *False* and indicates the use of unit sales.
	This parameter is used when the *delphi.TLandingAlgorithmName* parameter is set to *HMM*.
delphi.lifecycle.sellThruPercent	Use this parameter to specify the threshold (in percentage) for picking the tlanding date. The value defaults to 1.0.
	This parameter is used when the *delphi.TLandingAlgorithmName* parameter is set to *SellThru.*
delphi.lifecycle.typicalSeasonLength	Use this parameter to specify the typical season length (in weeks). The value defaults to 46.
delphi.lifecycle.minimumDataLength	Use this parameter to specify the minimum data length, as a fraction of typical season lenghts for all pre-season like items. An error gets logged in the Calc Engine response message, when the data available for pre-season like item is less than the product of the minimum data length and typical season length.
	The value defaults to 0.05.

Property Name	Property Description
delphi.lifecycle.nonPositiveSalesLimit	Use this parameter to set the non-positive sales limit, as a fraction of available data points. An error gets logged in the Calc Engine response message, when the number of data points with non-positive sales divided by the number of total data points is greater than the parameter value. The default value is 1 and indicates that the Calc Engine does not impose a requirement by default.
delphi.lifecycle.enableTrivialLifecycle	Use this parameter to enable the Calc Engine to apply a trivial logic and find lifecycles.
	When set to *True*, the Calc Engine takes the first date that has positive sales and positive inventory as tLanding, and tLanding + *delphi.lifecycle.typicalSeasonLength* as cycle end. The default value for this parameter is False and indicates the use of normal approach.
History loading parameters	
delphi.data.scantHistoryWarningLimit	For pre-season like items, indicates that a warning is generated if the number of weeks of history is lesser than the weeks specified in this parameter.
	To disable the warning, set the limit to -1.
delphi.data.scantSalesWarningLimit	For pre-season like items, indicates that a warning is generated if the sum of the unit sales in the first or last 52 weeks of sales for a pre-season like item are less than amount specified in this parameter.
	To disable the warning, set the limit to -1.
delphi.data.unbalancedSalesWarningLimit	For pre-season like items, indicates that a warning is generated if the ratio of sales units in the first & last years of history for a preseason like item is greater than the limit specified in this parameter.
	To disable the warning, set the limit to -1.
User lifecycle override parameters	
delphi.lifecycle.override. scantHistoryErrorLimit	For pre-season like items, indicates that an error is generated if the historical data available within the user-specified date range (lifecycle) are shorter than the number of weeks specified in this parameter.
delphi.lifecycle.override. scantHistoryWarningLimit	For pre-season like items, indicates that a warning is generated if the historical data available within the user-specified date range (lifecycle) are shorter than the number of weeks specified in this parameter.
	To disable the warning, set the limit to -1.
delphi.lifecycle.override. prolongedHistoryErrorLimit	For pre-season like items, indicates that an error is generated if the historical data available within the user-specified date range are longer than the number of weeks specified in this parameter.
	To disable the error, set the limit to -1.
Optional Deployment	
delphi.generate.i18n.keys	Use this parameter (set to *True*) to enable the use of internationalization information for faults, warnings, and information messages. The value defaults to *False* for backward compatibility.
delphi.user.language delphi.user.country	Use these parameters to specify the language and country settings for the application. The delphi.user.language parameter defaults to 'en' (English), and the delphi.user.country parameter defaults to 'US' (United States of America).

Property Name	Property Description
engine.record.messageCapture	Use this parameter (set to True) to enable message capture.
	When you enable this feature, you must also set the message capture directory in the *engine.record.directory* parameter.
engine.record.directory	Use this parameter to specify the location where the message captures get stored. When left blank, messages get stored in the WebLogic domain directory.
engine.record.internals	Use this parameter (set to *True*) to enable logging of internal engine messages.
	To use this feature, you must enable the message capture feature (engine.record.messageCapture)
engine.info.messages	Use this parameter (set to *True*) to generate informational messages in the response.
delphi.log4j.properties	Use this parameter to specify the file (delphi.log4j.properties) that contains properties controlling the Delphi logging behavior.
engine.connection.factory	The JMS Connection Factory look-up name.
engine.connection.factory	The JMS Connection Factory look-up name.
engine.CE.Response.Queue	The JMS Reply Queue name.
engine.Datasource.Runtime.Data	The JNDI name of the engine datasource to access connections to the database.
engine.data.access.jar	Use this parameter to specify the path to the JAR file that includes the set of data access classes used for implementing the standard CE Data Abstraction Layer (DAL) at runtime.
engine.data.access.factory	Use this parameter to specify the fully qualified name of the JAVA class which implements the Dal Factory Interface. See the Javadoc API documentation for the CE DAL for more information on this interface.
engine.allocation.supplyStartingInventory	Use this parameter (set to True) to include the information on the starting inventory in the result of an allocation request.
	The starting inventory is the initial inventory level of each SKU at each store at the start of the coverage period.
	To exclude this additional information, set the value to false.
engine.forecast.supplyActualLifecycles	Use this parameter (set to True) to include the pre-season lifecycles (used by Calc Engine) in the result of a forecast or forecast based allocation request.
	To exclude this information, set the value to False.
RMI Connection Pools	
engine.data.access.context	Use this parameter to specify the fully qualified name of the JAVA class that provides the data access context.
rmi.analytical.url	URL of the analytical connection pool.
rmi.analytical.user	The user name to connect to the analytical connection pool.
rmi.analytical.password	The password (associated with the user name) to connect to the analytical connection pool.
rmi.business.url	URL of the business connection pool.
rmi.business.user	The user name to connect to the business connection pool.
rmi.business.password	The password (associated with the user name) to connect to the business connection pool.
rmi.forecast.url	URL of the forecast connection pool.

Property Name	Property Description
rmi.forecast.user	The user name to connect to the forecast connection pool.
rmi.forecast.password	The password (associated with the user name) to connect to the forecast connection pool.
rmi.historical.url	URL of the historical connection pool.
rmi.historical.user	The user name to connect to the historical connection pool.
rmi.historical.password	The password (associated with the user name) to connect to the historical connection pool.
rmi.runtime.url	URL of the runtime connection pool.
rmi.runtime.user	The user name to connect to the runtime connection pool.
rmi.runtime.password	The password (associated with the user name) to connect to the runtime connection pool.
Optional Model	
delphi.newsizeprofiles	Use this parameter (set to True) to specify the use of the table based size profiles.
	To use the stored procedure based (older) size profiles, set the value to False.
delphi.updateZeroStoreWeights	Use this parameter (set to True) to specify the use of the budget-based weights for stores that get history-based weights of zero.
	To use the weights based on history alone, set the value to False.
delphi.useExternalStoreWeightsAsFallback	Use this parameter (set to True) for the Calc Engine to use the relative external store weight (in place of store budget) as a fallback when the item has no sales history.
	The value defaults to False for backward compatibility.
	When you set the value to True, ensure that external store weights exist in the database.
delphi.enableLikeStores	Use this parameter (set to True) to enable the use of like stores.
	If the like stores are not specified in the input XML files, this property is ignored.
delphi.minWeeksStoreWeightData	Use this parameter to set the minimum number of weeks of historical data required to calculate the pre-season store weights. In case the weeks qualified for store-weight data are lesser than the weeks specified in this parameter, the budget-based weights are used.
delphi.data.duplicateBoundaryRequestBehavior	Use this parameter to set the control behavior for the duplicate boundary requests (multiple like items). Set one of the following values:
	■ match - (default) all bounds must match
	■ error - generate an error if there is more one like item
	■ min - consider the earliest date
	■ max - consider the latest date
delphi.absolutePromotionalScaling	Set the parameter value to True to use absolute promotional scaling. The value defaults to False and indicates the use of relative scaling.
delphi.usePromotionalMediaTypes	Set the parameter value to True to use promotions of all media types.

Property Name	Property Description
delphi.inventoryEffectModel	Use this parameter to apply an inventory effect that attempts to correct the historical sales.
	Set one of the following values:
	■ none - (default) no inventory effect
	■ piecewiseLinear
	■ piecewiseLinearByStoreCount
	■ exponentialByStoreCount
	■ sigmoidByStoreCount
	The inventory effect is a number that represents the fraction of sales lost due to insufficient inventory in the store. The Calc Engine corrects for this by determining how much the historical sales were lowered by such an effect and inflating them to compensate.
delphi.inseasonUpdate.plcScale.exponentialSmoothing.enable	Use this parameter (set to True) to enable the exponential smoothing.
delphi.inseasonUpdate.plcScale.exponentialSmoothing.rate.preseason	Use these exponential smoothing parameters to set up the exponential smoothing rate applied on pre-season and in-season data. You can also specify the data window (in weeks) to be considered to be used.
delphi.inseasonUpdate.plcScale.exponentialSmoothing.notouch	
	These parameters can be used only when the exponential smoothing feature is enabled.
delphi.inseasonUpdate.plcScale.exponentialSmoothing.rate.inseason	
delphi.inseasonUpdate.plcScale.exponentialSmoothing.datawindow	**Note:** The default values for these parameters ensure optimal performance. Oracle recommends that these default values remain unchanged.
delphi.plc.tabulated.enable	Use this parameter (set to True) to specify that the application uses the Tabulated PLC feature.
	The Tabulated PLC feature allows the Calc Engine to use the Like Item's selling shape over time as the Product Lifecycle Curve (PLC) for the forecast. This feature enables the allocator additional ability to alter the shape and scale of the item's forecast.
delphi.plc.tabulated.movingAvg	Use this parameter to specify the weighing coefficients (an array of real numbers) for the moving average window that smoothens the corrected sales to get tabulated PLC.
	The valid values must be positive, and the size of the array must be an odd number. The value right in the middle of the array is the weight applied to the current date point.
	The value defaults to {0.1, 1, 0.1}.
delphi.plc.tabulated.extrapolation.count	Number of the valid data points used to compute the average as the extrapolations of the tabulated PLC. The value must be a positive integer.
	The default value is 2. For optimal performance, ensure that the value for this parameter is set to 3.
delphi.plcfit.RValue	Indicates the R value threshold to select a fashion fit.
delphi.plcfit.minTimeToPeak	Indicates the minimum time-to-peak (in weeks) the Calc Engine must try when fitting fashion PLC.
	Valid values can be between 1 and 9.
delphi.plcfit.baseLevel.min	Use these parameters to control the base levels during a fashion PLC fit. The Calc Engine selects the best values among the range that is generated based on the min, max, and step specified in these parameters.
delphi.plcfit.baseLevel.max	
delphi.plcfit.baseLevel.step	
	Valid values can be between 0 and 1.

Property Name	Property Description
delphi.plcfit.minFashionWeeks	Use this parameter for the Calc Engine to force a basic PLC fit, if after data cleansing, the preSeason like item has fewer weeks of data than the number of weeks specified through this parameter.
	Valid values can be between 4 and 51. For optimal performance, ensure that the value for this parameter is set to 6.
delphi.plcfit.preseasonLifecycleSpan	Use this parameter to control the pre-season lifecycle span calculations. Valid values are "old" and "new".
	The value defaults to "old" and indicates the use of t1 and t2 calculations.
delphi.adjust.inseason.plc.alignment	Use this parameter to set the inseason PLC alignment behavior based on the inseason like item, inseason lifecycle, and stores sales plan.
	Set the value to *false* to specify that the sales plan start date be always used. When set to *true*, the inseason lifecycle start date is used for the forecasted item that acts as the inseason like item for itself and the inseason lifecycle start date is later than the sales plan start date.
	When the inseason like item is the forecasted item itself and the inseason lifecycle start date found by calc engine is later than store sales plan start date, it is recommended that the inseason updated PLC is aligned with the item's inseason lifecycle start than sales plan start date.
delphi.forecastedItemID.likeItemID.comparable	Use this parameter to enable the comparison of the forecasted item ID and like item ID. The same IDs will indicate that the like item is the forecasted item itself.
	The value of the parameter defaults to *false* indicating that such a conclusion cannot be arrived based on the IDs.

IR Performance Tuning Strategies

strategy.activitydata	Use these parameters to tune the performance of the IRs by changing the strategies.
strategy.businesspolicy	
strategy.distribution	Valid strategy definitions are single, list, and template.
strategy.forcedmarkdowns	
strategy.itemdates	
strategy.itemparameters	
strategy.itemprices	
strategy.markdowncalendar	
strategy.modelvalues	
strategy.pastticketprices	
strategy.pendingmarkdowns	
strategy.plannedpromos	
strategy.priceladder	

Test Configuration (Set only while testing the software.)

delphi.disableHistoricPromotions	Use this parameter (set to True) to disable the removing the promotion lifts from the historical data.
delphi.disableFuturePromotions	Use this parameter (set to True) to disable the sales lifts of future promotions by forcing them to have a lift of 1.0.

Property Name	Property Description
delphi.disableFuturePriceEffects	Use this parameter (set to True) to set the price elasticity exponent to zero.
delphi.disableMerchandiseDistribution	Use this parameter (set to True) to set the merchandise distributions to 1.0.
delphi.UseCache	Use this parameter to specify the use of the Pine cache. Valid values are "disabled" and "required".
	Set the value to "disabled" to disable the use of the Pine cache.
engine.data.access.statsFile	Use this parameter to specify the name of the file that will include the data access layer (DAL) performance statistics.
	The statistics are updated after each processed message.
CSV DAL Record/Playback	
engine.csvdal.useMessageNameAsSubDir	Use this parameter to specify that the data recorded for each message is recorded in a separate subdirectory of the recording directory and the subdirectory uses the message name as its name. The value defaults to True.
	The CSV DAL record/playback method supports merging the data from multiple messages into a single collection of csv files. To use this mode, set the value to False.
CsvDal recording parameters	
engine.csvdal.recordEnable	Indicates that the CSV DAL recording is stored in the directory specified in the *engine.csvdal.recordTo* parameter.
engine.csvdal.recordTo	Location where the Direct CSV DAL recording gets stored. This directory is created if it does not exist.
engine.csvdal.clobber	Use this parameter (set to True) to delete any existing data files in the CSV DAL recording directory, before the recording begins.
	When set to False, the existing files are updated with the data from the current recording session.
CsvDal playback parameters	
engine.csvdal.playbackFrom	Indicates the CSV DAL playback directory.
Automated Failure Generation	
engine.failure.lowerBound	Indicates the range of valid item identifiers.
engine.failure.upperBound	
engine.failure.rate	Indicate the fraction (not the percentage) of the items within the *lowerBound* and *upperBound* that will be failed.
engine.failure.logFileName	Indicates the name of a file that will contain all item IDs of items that fail.

9

Front End Metrics

This chapter contains the following:

Assortment View Metrics

The Assortment View screen displays planned receipts of merchandise at the item and color level by volume groups and subsets for the designated plan period. The display also provides top-down financial budget data at the total and volume group/subset levels for the designated plan period and shows how the assortment plan compares to budget constraints. See the *Plan User Guide* for more information on the Assortment View screen.

Table 9–1   Assortment View Metrics

Column Group				
Display Name	**Metric**	**Description**	**Method**[1]	**Calculation**
Store Totals - Receipts				
Total Units	TotalReceiptUnits	Total receipt units.	C	Total Store Receipt U (Act U + Receipt U)
Total $	TotalReceiptsDollars	Total receipt value of the merchandise in dollars.	C	Total Store Receipt U * Initial Retail
% Receipt U	ReceiptUnitsTotalPct	Receipt unit percent contribution to plan. EDITABLE To adjust the flow over time of the receipts. True for sales also.	C	Style_Total_Receipt U/ Plan_Total_Receipt U * 100
% Receipt $	ReceiptDollarsTotalPct	Receipt dollars percent contribution to plan.	C	Style_Total_Receipt $/ Plan_Total_Receipt $ * 100
Act $	PastReceiptDollars	Actual receipts values in dollars for a given item per StoreSet/All.	C	Act U * Initial Retail
Act U	PastReceiptUnits	Actual receipt units for a given item per StoreSet/All.	F	Actual Store Receipts U
Receipt $	Total ReceiptDollars	Planned receipt dollars for a given item per StoreSet.	C	Receipt U * Initial Retail
Receipt U	ReceiptUnits	Planned receipt units for a given item per StoreSet/All.	C	Planned Future ReceiptUnits

Table 9–1   (Cont.) Assortment View Metrics

Column Group				
Display Name	**Metric**	**Description**	**Method[1]**	**Calculation**
Store Totals - Sales				
Total U	TotalSalesUnits	Total of all sales units for a given item per StoreSet/All.	C	Act Sales Units + Future Sales Units.
Total $	TotalSalesDollars	Total of all sales dollars for a given item per StoreSet/All.	C	Act Sales Dollars + Future Sales Dollars.
Act $	PastSalesDollars	Actual sales in dollars for a given item per StoreSet/All.	F	Actual Sales $, Fed
Sales $	TotalSalesDollars	Planned sales in dollars for a given item per StoreSet/All.	C	Sum of Weekly Sales $ for all future weeks in current Plan
Act U	PastSalesUnits	Actual sales units for a given item per StoreSet/All.	F	Actual Sales, Fed
Sales U	TotalSalesUnits	All planned sales units for a given item per StoreSet/All.	C	Planned Future SalesUnits
Store Totals - Inventory				
BOH U	BOHUnits	Beginning on hand units of the plan period for a given item per StoreSet/All.	F	For items carry-over from Prev-Plan, BOH = EOH units of Previous Plan (if any) For others, BOH = 0
BOH $	BOHDollars	Beginning on hand dollars of the plan period for a given item per StoreSet/All.	C	BOHUnits * Current Retail
EOH U	EOHUnits	Ending on hand units of the plan period for a given item per StoreSet/All.	C	- Pre-season Units = BOH + Receipt Units – Sales Units - In-season Units = OH + Receipt Units – Sales Units Plan until OH overtakes last week at which the front-end will used the OH amount as EOH.
EOH $	EOHDollars	Ending on hand dollars of the plan period for a given item per StoreSet/All.	C	EOH * Current Retail
OH U	OHUnits	On Hand units of last week of Actuals for a given item per StoreSet/All.	F	In-season only = EOH units of last week of actuals
OH $	OHDollars	On hand dollars of last week of actuals for a given item per StoreSet/All.	F	In-season only = EOH dollars of last week of actuals

[1] Key: F = Feed. C = Calculation. N = None.

Item View Metrics

The Item View screen displays an item or group of items and their metrics by week. It provides visibility into planned and actual sales, receipt, inventory, gross margin, and key metrics by week and month. See the *Plan User Guide* for more information on the Item View screen.

Table 9–2   Item View Metrics

Column Group				Weekly Calculation
Display Name	Metric	Description	Method[1]	Monthly or Grand Total Calculation
Store Receipts				
Receipt U	RECEIPT_UNITS	Total receipt units for time period.	F	Store Receipts (units) in this week In AP & Need, actuals are updated LSP : Always from last submitted SUM[Weekly RECEIPT_UNITS]
% Receipt U	RECEIPT_UNITS_PCT	Percent of receipt units for a time period contribution to the total receipts for the item. EDITABLE FOR AP-segment; allows the user to alter the receipt flow.	C	Receipt U (for the week) / Plan_Total_ Receipts_Units * 100 SUM[Weekly Store RECEIPT_UNITS] / Plan_Total_Receipts * 100
Receipt $	RECEIPT_$	Total receipt dollars for time period.	F	RECEIPT_UNITS * Initial Retail SUM[Weekly RECEIPT_$]
% Receipt $	RECEIPT_$_PCT	Percent of receipt dollars for a time period contribution to the total receipts for the item.	C	RECEIPT_$/ Plan_Total_Receipts_$s * 100 SUM[Weekly Store RECEIPT_$] / Plan_Total_Receipts_$s * 100
DC Receipts				
DC Receipt U	DC_RECEIPT_UNITS	Total DC receipt units by time period.	NA	DC_RECEIPT_UNITS for the week In AP & Need, comes from confirmed receipts LSP : Always from last submitted SUM[Weekly DC_RECEIPT_UNITS]
DC RECEIPT $	DC_RECEIPT_$	Total DC receipt dollars by time period.	NA	DC Receipt U * InitialRetail SUM[Weekly DC_RECEIPT_$]
DC EOH U	Metrics.DcEohUnits	Number of units in the Distribution Center (DC) at the end of a time period.		DC_EOH (Units) at the end of current week DC EOH Units in the Last Week of the Month.
DC EOH $	Metrics.DcEohDollars	Total value in dollars of DC End-On-Hand.		DC EOH U * CurrentPermPrice$ DC EOH Dollars in the Last Week of the Month.
In Transit U	Metrics.InTransitUnits	Total units in transit during this time period. Units shipped from DC and yet to be received at stores		Calculated based on Store Receipt Date and Transit Time for each store. In Transit U for the last week of the month
In Transit $	Metrics.InTransitDollars	Total value in dollars of the in transit inventory.		In Transit U * CurrentPermPrice In Transit $ for the last week of the month
Sales				
Sales U	SALES_UNITS	Total sales units by time period.	F	SALES in units in current week In AP & Need, actuals are updated LSP : Always from last submitted Sum of all Weekly Sales (units) in this month

Table 9–2   (Cont.) Item View Metrics

Column Group				Weekly Calculation
Display Name	Metric	Description	Method[1]	Monthly or Grand Total Calculation
% Sales U	SALES_UNITS_PCT	Percent of sales units for a time period contribution to the total sales units for the item EDITABLE FOR AP-segment; allows the user to alter the sales flow.	C	Sales U / Total SALES_UNITS Sum of all Weekly Sales / Total SALES_UNITS * 100
Sales $	SALES_$	Total sales dollars by time period.	F	(SALES_UNITS * AUR) * (1-AUR Adjustment/100) Sum of all Weekly Sales $ in this month
% Sales $	SALES_$_PCT	Percent of sales dollars for a time period contribution to the total sales $ for the item.	C	Sales $ / Total SALES_UNITS Sum of all Weekly Sales $/ Total SALES_$ * 100
BOH				
BOH U	BOH_UNITS	Beginning on hand units for a time period.	C	Sum of BOH Store Inventory (Units) at the start of this week Calculated in Load Script BOH_UNITS in the First Week of the month
BOH $	BOH_$	Beginning on hand dollars for a time period.	C	BOH U * CurrentPermPrice Calculated in Load Script BOH_$ in the First Week of the month
EOH				
EOH U	EOH_UNITS	Ending on hand units for a time period.	F	Sum of EOH Store Inventory (Units) at the end of this week In AP & Need, Actuals is updated LSP : Always from last submitted EOH_UNITS in the Last Week of the month
EOH $	EOH_$	Ending on hand dollars for a time period.	C	EOH U * CurrentPermPrice EOH_$ in the Last Week of the month
Markdown				
PERM MD $	PERM_MD_$	Permanent markdown $ for a time period triggered by change in total inventory valuation due to a permanent markdown.	F	BOH_Units * (CurrPermPrice-in-LastWeek - CurrPermPrice) + Receipt U * (RETAIL - CurrPermPrice) Sum of Perm MD $ for all weeks in this month
POS MD $	POS_MD_$	POS (temporary) markdown $ component of weekly sales.	F	Sales U * (CurrPermPrice - AUR) Sum of POS MD $ for all weeks in this month
TOTLA MD $	TOTAL_MD_DLR	Total markdown $, either promo or permanent markdown	C	POS MD $ + PERM MD $ Sum of Total MD $ for all weeks in this month
MD %	Metrics.MarkdownPct	Markdown percent.	C	MD$/Sales $ @ Retail * 100 Month MD$/Month Sales $ @ Retail * 100

Table 9–2   (Cont.) Item View Metrics

Column Group				Weekly Calculation
Display Name	Metric	Description	Method[1]	Monthly or Grand Total Calculation
MD IND	Metrics.PricingCodesString	Displays multiple values for a given week (MD, POS, TRF), and is driven off of the pricing plan. It behaves the same way for planned and actual weeks.	NA	Shows whether there is a MD or POS-event or Traffic event effective during the current week NA
POS/Perm MD Sales				
POS Sales U	PR_SALES_UNITS	Sales units of items on promotion.	F	IF Item is on promotion, POS Sales U = Sales U for the week Sum of all weeks of the month
POS Sales $	PR_SALES_$	Sales $ of promotional items.	F	IF Item is on promotion, POS Sales $ = Sales $ for the week Sum of all weeks of the month
POS Sales %	PR_Sales_$_PCT	Ratio of promotional sales $ to total sales $.	C	For the week, PR_SALES_UNITS / SALES_UNITS * 100 For the month, PR_SALES_UNITS / SALES_UNITS * 100
MD Sales U	MD_SALES_UNITS	Sales units of clearance items.	F	IF Item is on markdown, MD Sales U = Sales U for the week Sum of all weeks of the month
MD Sales $	MD_SALES_$	Sales $ of clearance items.	F	IF Item is on markdown, MD $ = Sales $ for the week Sum of all weeks of the month
MD Sales %	MD_SALES_%			For the week, MD_SALES_UNITS / SALES_UNITS * 100 For the month, MD_SALES_UNITS / SALES_UNITS * 100
Avg Unit				
Sales AUR	AUR	Average unit retail for a time period.	C	Sales $ / Sales U Sum (Sales $ in the month) / Sum (Sales U in the month)
AUC	AUC	Average unit cost for a time period.	C	SalesDollarsAtCost/SalesUnits SalesDollarsAtCost/SalesUnits
Gross Profit				
GP %	GP_PCT	Gross profit percentage by time period	C	IMU * (1- (MD % + Merch Charge % + Shrink % + Freight %)) (SUM [WeeklyGrossProfitDollars] / MonthlySalesDollars) * 100
GP $	GP_$	Gross profit dollars by time period.	F	[GP% * Sales $] / 100 Sum of Weekly GP $ in the month
GPROI	GPROI	Gross profit return on investment.	C	NA SUM [WeeklyGrossProfitDollars] / AverageInventoryDollarsAtCost
Sell Through				
% ST $	ST_$_PCT	% sell through in dollars.	C	LTD_SALES_$ / (LTD_RECEIPT_$) % ST $ of last week in the month
% ST U	ST_UNITS_PCT	% sell through in units.	C	LTD_SALES_UNITS / (LTD_RECEIPT_UNITS) % ST U of last week in the month

Table 9–2   (Cont.) Item View Metrics

| Column Group | | | | Weekly Calculation |
Display Name	Metric	Description	Method[1]	Monthly or Grand Total Calculation
TO $	TO_$	Turn over dollars.	C	NA SALES_$ / AVG_INV_$_RTL
TO U	TO_UNITS	Turn over units.	C	NA SALES_UNITS / AVG INV U
Num Stores				
FWOS	WOS	Forward weeks of supply.	C	Weekly WOS = Current BOH_UNITS / SUM[Weekly SALES_UNITS by week] where SUM[Weekly SALES_UNITS by week] < BOH_UNITS Remainder is dropped. If not enough sales weeks to reach constraint, then WOS metric will append a '+' Monthly WOS = First week of Month Grand Total WOS = First Month of Display
ST w/Inv	ST_WITH_INV	# of stores with inventory for a time period.	C	# of Stores w/ EOH_UNITS > 0 MAX[Weekly ST_WITH_INV]
ST w/Rec	ST_WITH_REC	# of stores with receipts for a time period.	C	# of Stores w/ RECEIPT_UNITS > 0 in the week MAX[Weekly ST_WITH_REC]

[1] Key: F = Feed. C = Calculation. N = None.

Flow View Metrics

The Flow View tab displays the actual data for the active items and the planned data for the inactive items. It displays the AP receipt quantities flow by week throughout the relevant plan period. You can use the screen to compare receipt plans to the top-down financial plan. The data presented in the Flow View tab is read only, but you can update the data by adjusting each item's calculation parameters or the overall planned receipt quantities in the Assortment View screen. See the *Plan User Guide* for more information on the Flow View tab.

Table 9–3   Flow View Metrics

| Column Group | | | Weekly Calculation |
Display Name	Metric	Description	Monthly or Grand Total Calculation
Store Receipts			
Receipt U	RECEIPT_UNITS	Total receipt units for time period.	Planned / actual Store Receipts (units) in this week SUM[Weekly RECEIPT_UNITS]
Receipt $	RECEIPT_$	Total receipt dollars for time period.	RECEIPT_UNITS * InitialRetail SUM[Weekly RECEIPT_$]
DC Receipts			
DC Receipt U	DC_RECEIPT_UNITS	Total DC receipt units by time period.	DC_RECEIPT_UNITS for the week SUM[Weekly DC_RECEIPT_UNITS]
DC Receipt $	DC_RECEIPT_$	Total DC receipt dollars by time period.	DC Receipt U * InitialRetail SUM[Weekly DC_RECEIPT_$]

Plan Worklist Fields

The following fields contain information about items. These fields are either user-input fields or calculated fields, as indicated in the table. For more information, see the *Plan User Guide*.

Table 9–4   Plan Worklist Fields

Column Heading Fields	Input/Calculated	Description
Item Info		
Item ID	Input	Actual or mock item ID
Description	Input	Item description
Purchase Type	Input	Fashion or Basic.
Colors	Input	Number of item c olors in the plan.
Status		
Actualize	Input	Actual or mock item.
AP	Calculated	Item status. I = In Progress. R = Ready. V = Validated. S = Submitted. C = Changed.
Active	Calculated	The current actrivity status of an in-season item. Active = at least one item contains actual sales. Inactive = item has not yet started selling. Late = no sales relative to the Plan Start Date.
CO	Calculated	Carryover item (Y/N).
Rcpt %	Calculated	Receipt percentage of AP data segment. User = edited. Sys = system-generated.
Sales %	Calculated	Sales percentage of AP data segment. User = edited. Sys = sytem-generated.
Hierarchy Info		
Sub-department	Input	Departments within the plan department.
Class	Input	Class within sub-department.
Sub-class	Input	Sub-class within class.
Required Info		
Product Group	Input	Item product group.
Cost	Input	Item cost.
Retail	Input	Item initial retail cost.
IMU	Calculated	Calculated value (1 - cost/Retail).
Pack	Calculated	Average prepack approximate size.
Min	Input	Minimum receipt quantity per store per item per plan period.
Pack Min	Input	Initial presentation minimum required.

Table 9–4   (Cont.) Plan Worklist Fields

Column Heading		
Fields	**Input/Calculated**	**Description**
Optional Info		
Brand	Input	This optional information in the Plan worklist is for display only. These values do not drive an calculation login within the application.
Vendor	Input	
Collection	Input	
Size Range	Input	
Floor Set	Input	
Color Family	Input	
Season Code	Input	
Vendor Style	Input	
Prod Type	Input	
Alloc Flag	Input	
Attributes 1 - 5	Input	
Notes	Input	
Client Info		
Fabric	Input	This optional information in the Plan worklist is for display only. These values do not drive an calculation login within the application.
Fit	Input	
Rise/Sleeve	Input	
Fabric Characteristics	Input	
Reg/Petite/Tall	Input	
Lifestyle	Input	
Features	Input	
Neck	Input	
Theme Code	Input	
Theme Name	Input	
Buy Parameters		
Like Item	Input	Indicates whether like item has been set.
Store Base	Input	Indicates whether eligible stores have been selected.
Store Flow	Input	Indicates whether store item flow has been defined.
Pricing Plan	Input	Indicates whether pricing strategy has been completed.
DC flow	Input	Indicates whether DC flow has been completed.
Forecast	Input	Forecast status. Completed, Pending, Failed, or Invalid.
Pack Configurations		Indicates whether an item has valid packs. View = valid packs. Not Available = no valid packs.

Configurable Metrics Framework

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.

Index

S

T

U

Y