Chapter 3. Customizing Workforce Planning

About Customizing Your Application
Adding a Performance Rating
  With Performance Management Architect Applications
  With Classic Applications
Changing the Pay Rate for Maternity Status
  With Performance Management Architect Applications
  With Classic Applications
Changing the Frequency of Bonuses
  With Performance Management Architect Applications
  With Classic Applications
Adding a Salary Adjustment Type
  With Performance Management Architect Applications
  With Classic Applications

Appendix A. Workforce Planning Structure
Predefined Data Forms
Predefined Accounts
Predefined Member Formulas
  Adjusted Annual Salary
  Adjusted Annual Salary (Prior)
  Annual Overtime
  Annual Salary
  Bonus
  Bonus %
  Bonus Basis
  Cal TP-Index
  Contractor FTE
  Contractor Headcount
  CYTD Sign On Bonus
  CYTD Total Salary
  CYTD Total Salary (Prior)
  Departed Headcount
  Differential %
  Fiscal TP-Index
  FT/PT
  FUTA
  FUTA Cap
Contents

Temporary FTE ........................................................................ 84
Temporary Headcount .......................................................... 85
Turnover Adjustment .............................................................. 85
Turnover Headcount Adjustment ............................................. 85
Predefined Smart Lists .......................................................... 86
Predefined Smart List Entries .................................................. 86
Predefined Menus .................................................................. 88
Predefined Business Rules ....................................................... 90
  Add TBH Hourly ................................................................ 90
  Add TBH Salary ................................................................ 91
  Change Employee Status ..................................................... 91
  CopyProps ....................................................................... 92
  Plan Departure .................................................................. 93
Reconcile TBH ....................................................................... 93
Remove TBH ......................................................................... 94
Rollup .................................................................................. 94
Transfer ................................................................................ 94
  Transfer In ....................................................................... 95
  Transfer Out ..................................................................... 96
Unreconciled Transfers .......................................................... 97

Index .................................................................................... 99
About Workforce Planning

Oracle's Hyperion® Workforce Planning is a Web-based workforce planning and budgeting solution. It gives corporations a tool to model future headcount and related expenses, and provides a reliable source of up-to-date information on workforce expenses. By planning salary expense in direct correlation to headcount, planners can effectively manage one of their largest variable expenses.

With Workforce Planning, all decision-makers and front-line managers can communicate which course of action to take and get budget holders to collaborate so that the headcount process is optimized and efficient. When a material event occurs that causes a change in direction, planners have the flexibility to adapt rapidly, ensuring that plans are relevant and useful.

More specifically, Workforce Planning:

- Calculates such workforce-related expenses as headcount, payroll, salary, taxes, and health care benefits
- Includes transfer functionality to facilitate the management of headcount across dynamic organizations
- Provides a framework for customizing the planning process to meet the needs of global enterprises
- Provides ease of use through the Web with drill-through capability from summary values to the underlying detailed data
- Includes event-based activities, such as new hires that can trigger expenses for space allocation, equipment, and hiring bonus
- Integrates with actual Oracle's Hyperion® Planning – System 9 data for reconciliation, forecasting, and reporting purposes
Integrates with other systems to load information: with flat files for Oracle's Enterprise Performance Management Architect applications, and typically with Oracle's Hyperion® Data Integration Management Adapter for Planning or Oracle's Hyperion® Application Link Adapter for Hyperion Planning for Classic applications

- Supports working with Workforce Planning data forms using Oracle's Hyperion® Smart View for Office, including taking them offline.

**Prerequisites**

This section describes prerequisites for setting up and managing Workforce Planning. You should understand:

- Planning functionality (see the *Hyperion Planning – System 9 Administrator’s Guide*, the *Hyperion Planning – System 9 User’s Guide* and their online help)
- The Workforce Planning business model (see “Business Model” on page 8)
- If you are using Performance Management Architect, see the *Hyperion Enterprise Performance Management Architect Administrator’s Guide* and online help.

**Business Model**

All companies create plans to help prepare for the future, aligning their limited corporate resources—people and dollars—against the strategies that they believe best leverage their competitive market advantage. Through collaborative planning, departments coordinate and allocate the finite resources of the company. Companies that can best detect market opportunities and quickly realign their resources gain a competitive advantage. A company’s workforce represents its most critical resource.

Using Workforce Planning, companies determine the employee resources needed to achieve their targets, assign existing employees to various positions, and plan for adding new employees. Companies must also gauge the various direct and indirect costs incurred by employees, such as health care, equipment, and taxes. Workforce Planning calculates these expenses—both simple and complex—based on certain drivers. Driver-based planning runs key business assumptions through models, providing the insight to proactively manage the volatility of future financial performance. For example, performance drives bonus and merit increases, primary factors in determining total compensation.

Workforce Planning allows actions on employees, such as transferring them to another department, planning for their departure, and placing them on maternity or leave of absence.

Corporate planners, operational managers, or department managers prepare the workforce plans, sometimes including multiple scenarios. They submit them to senior financial and Human Resource managers for review and approval. A corporate planner typically consolidates the plan and prepares reports on the workforce. Companies can iterate plans as often as needed to respond to changing conditions.
Creating and Initializing a Workforce Planning Application

This section describes how to create and initialize a new Workforce Planning application, using either Performance Management Architect application administration (see “Using Performance Management Architect Application Administration” on page 10) or Classic application administration (see “Using Classic Application Administration” on page 14).

Initializing Workforce Planning loads predefined:

● Workforce Planning account, entity, employee, scenario, and version members
● Data forms
● Smart Lists
● Member formulas
● Business rules
● Menus
● UDA

Before you initialize Workforce Planning:

● Study Appendix A, “Workforce Planning Structure”. Review the predefined elements, identifying which elements you can use and which elements you need to customize. The better you understand the model and plan your application, the easier it will be for planners to use Workforce Planning.

● Update your dimension outlines to resolve differences between Workforce Planning member names and names in other applications.

● Plan the structure of the Entity dimension if you intend to use both Workforce Planning and Oracle's Hyperion® Capital Expense Planning because initializing Workforce Planning produces one No Entity member, whereas initializing Capital Expense Planning produces a few Entity members. You can change the Entity members after you initialize the applications.

● Plan the structure of the Entity dimension if you intend to use both Workforce Planning and Capital Expense Planning because initializing Workforce Planning produces one No Entity member, whereas initializing Oracle's Hyperion® Capital Expense Planning produces a few Entity members. You can change the Entity members after you initialize the applications.

Note:

The Workforce Planning model is based on a 12-month calendar, and the default calculations in Workforce Planning support a single year. You can modify the calculations, however, to support multiple years. Also, some member formulas such as spreading rely on a monthly calendar. If your application uses custom time periods, you must modify these formulas to support custom time periods.
Using Performance Management Architect Application Administration

Follow the steps in this section to use Performance Management Architect application administration.

First: Create an Import Profile

An import profile includes important information about the dimensions to be imported such as new dimensions, whether to merge or replace dimensions, and dimension properties.

To create an import profile:

1. Download the Workforce Planning flat file from the Oracle® E-Delivery site.
2. Log on to Performance Management Architect.
3. Select Navigate > Administer > Dimension Library.
4. In the Dimension Library, select File > Import > Create Profile. This enables you to create and execute an import profile to load dimensions and members from the Workforce Planning flat file into the Master View, so you can drag and drop Workforce Planning dimensions and members into the new Workforce Planning application.
5. In New Profile, enter a name for the new profile.
6. Select Flat File as the Import Type.
7. Enter the name of the Workforce Planning flat file. Or click Upload, browse for the file, and click Upload again.
8. Click OK.
9. Optional: Select Strip Quotes String to remove single quotes from a string.
10. Optional: Select Trim Spaces to remove spaces in the load file.
11. Click Next.
12. In Map Dimensions, select Create Dimensions For the Non-mapped Dimensions with the Source Dimension Name. When you select this option, the Master View column is automatically populated with the names of the dimensions from the Workforce Planning flat file. You must import all the dimensions from the Workforce Planning flat file.

When you import the dimensions from the Workforce Planning flat file into Performance Management Architect for the first time, the dimensions do not yet exist in the Performance Management Architect Master View and are unmapped (that is, they are not automatically or manually mapped to dimensions in the Performance Management Architect Master View). For any dimension that is unmapped, the name of the dimension in the Workforce Planning flat file is the name used to create the new dimension in the Performance Management Architect Master View. For information on manual and automatic mapping methods, see the Hyperion Enterprise Performance Management Architect Administrator’s Guide or help system.
Note:
Because you are creating new dimensions, the merge and replace options are not applicable for the unmapped dimensions that use the dimension name from the source.

13 Click Next.

14 To import dimension properties from the Workforce Planning flat file, select the properties to import in the Master View column.

Note:
You can double-click on a cell in the Master View column for a drop-down list of available properties.

15 Optional: Select Ignore Nulls for any property to ignore blank values in the Workforce Planning flat file.

16 Click Finish.

Second: Import Dimensions
You import dimensions from the Workforce Planning flat file into Performance Management Architect.

➤ To import dimensions:

1 In Performance Management Architect, select Navigate > Administer > Dimension Library.

2 In the Dimension Library, select File > Import > Import Dimensions.

3 Select an import profile to use.
   This populates the Name text box.

4 Select Flat File as the import file type.

5 Enter the name of the Workforce Planning flat file. Or click Upload, browse for the file, and click Upload again.

6 Click Import.
   If the dimension loaded into the Dimension Library is related to an Application View, it is loaded to the Application View.

Now you are ready to create a Workforce Planning application.

Third: Create a Workforce Planning Application
To add a Workforce Planning application to an existing Planning application, skip to the next section.

➤ To create a Workforce Planning application:

1 In Performance Management Architect, select Navigate > Administer > Application Library.

2 In the Application Library, select File > New > Application View.
In New Application View:
  a. Type an application name (for example, Wrkplan).
  b. Select the application type, Planning.
  c. Enter a description.

Click OK.

In the Dimension Library, from the Master View, select the new Application View (for example, Wrkplan).

Drag and drop dimensions and members from the Master View into the Application View to add them to the Workforce Planning application.

The Workforce Planning dimensions and members are listed in the flat file with the prefix planning_WF_ (for example, planning_WF_Employee)

Note:
Add all the Workforce Planning dimensions to the application.

In Properties of [application name], enter these values:
  a. From Category, select Planning.
  b. Select Valid for Wrkforce, and, optionally, the check boxes next to the other plan types for which the application is valid.

Note:
The Valid for Wrkforce check box is selected by default from the Workforce Planning dimensions that you imported.
  c. Select or change other values for the new application.

For each dimension in the application, select the Valid For plan types for which the dimension is valid.

Click Save.

Fourth: Add a Workforce Planning Application to an Existing Planning Application

If you have an existing Planning application, follow these steps. Otherwise, skip to the next section.

To add a Workforce Planning application to a Planning application:

1. In Performance Management Architect, select Navigate > Administer > Application Upgrade to access the Application Upgrade Wizard.
   a. In Upgrade Summary, review the list of the applications you can upgrade. Click Next.
   b. In Select application, use the arrow keys to select the Planning application to upgrade. You can select more than one application. Click Next.
   c. In Summary, review the list of applications you selected to upgrade. Click Next.
d. In Upgrade Status, click Finish to see the status of the upgrade.

2 In the Application Library, right-click the upgraded Planning application and select Open.

3 In the Dimension Library, drag and drop dimensions and members from the Master View into the Application View to add them to the application.

The Workforce Planning dimensions and members are listed in the flat file with the prefix planning_WF_. For example, Employee is listed as planning_WF_Employee.

Note:

Add all of the Workforce Planning dimensions to the Planning application.

4 For the upgraded application, in Properties of [application name], enter values for these properties:
   a. From Category, select Planning.
   b. Select the check boxes next to the plan types for which the application is valid. By default, Valid for Wrkforce is selected from the Workforce Planning dimensions that you imported.
   c. Select or change other values for the new application.

5 For each dimension in the application, select the Valid For plan types for which the dimension is valid.

6 Click Save.

Fifth: Deploying a Workforce Planning Application to Planning

Before you deploy a Workforce Planning Application View to Planning:

- Set up a relational database and a data source in the Planning system database. See the Hyperion Planning – System 9 Installation Guide.
- Ensure that there is a valid Planning/Oracle’s Hyperion® Business Rules connection so Workforce Planning business rules are imported into the Oracle’s Hyperion® Business Rules repository. See the Hyperion Enterprise Performance Management Architect Administrator’s Guide or help system.

Ensure that all the properties meet your needs and update any settings before Application Views. If applications are not set up correctly, deployment fails and error messages are displayed. Your Workforce Planning application is initialized when you successfully deploy an Application View with Valid for Wrkforce selected.

➢ To deploy a Workforce Planning Application View to Planning:

1 In Performance Management Architect, select Navigate > Administer > Application Library.

2 In the Application Library, right-click the Wrkplan Application View and select Deploy.

3 If the Application Validation Errors dialog box is displayed, view the errors and fix them. After you fix the errors, click Deploy again.

4 In the Deploy dialog box, enter or select:
   - The instance name for this deployment of the Workforce Planning application
- The application server IP address to which to deploy the new application
- The Oracle's Hyperion® Shared Services project name in which to register the new application
- The data source file name in which the relational tables will be generated
- The Create outline check box, if this is your first deployment and you want to create a new database outline in Oracle's Hyperion® Essbase® – System 9
- The Refresh outline check box, if you have an Essbase outline and want to update it with new data
- The Create security filters check box, to generate security filters for all users of the new application
- The Shared members security filters check box, to apply access permissions to shared and base members
- The Validate security filter limit check box, to flag security filters that exceed the Essbase limit of 64 KB per row

5 Click Deploy.

After you deploy the Application View from Performance Management Architect to Planning, communicate the URL for logging on, the log on instructions, and information about the planning process.

Tip:
You can put the URL on your company's intranet.

Using Classic Application Administration

If you upgrade your Workforce Planning application from an earlier release and want to use Classic application administration, install the new release of Planning, using the instructions in the Hyperion Planning – System 9 Installation Guide.

To create a new Workforce Planning application and use Classic application administration, see the Hyperion Planning – System 9 Installation Guide and the Hyperion Planning – System 9 Administrator’s Guide. When using the Create Application wizard to create a Classic application, select Wrkforce as the plan type.

After upgrading or creating your Workforce Planning Classic application, you must initialize it.

To initialize Workforce Planning:
1 Start Oracle's Essbase® Administration Services server.
2 Start and log on to Planning.
3 In Planning, select Edit > Initialize Workforce.
   A message confirms when initialization is successful; Initialize Workforce no longer displays on the menu.
Ensure that the Workforce Planning model suits your company's needs.


Refresh the application.


Caution!

To prevent overwriting of your Actual data, tag your Actual scenarios with the UDA named ACTUAL before you run any calculations.

Loading Information into Workforce Planning

If your Workforce Planning application is new, you may want to load information, such as the existing employee structure, employee properties, and payroll data, from enterprise systems:

- If you are using Performance Management Architect application administration, load information using a flat file (see the Hyperion Enterprise Performance Management Architect Administrator's Guide).

- If you are using Classic application administration, you can use DIM Adapter for Planning (see “About Loading Information Using DIM Adapter for Planning” on page 16) or Application Link (see “About Loading Information Using Application Link Adapter for Hyperion Planning” on page 16).

You can also:

- Manually enter information into Workforce Planning.

- Load metadata and data that generates data and rules files for loading metadata and data through Administration Services. See the Hyperion Application Link Adapter for Hyperion Planning – System 9 Online Help.

To load workforce information—data and metadata—using DIM Adapter for Planning or Application Link Adapter for Hyperion Planning into a Classic application:

1 Load metadata (for example, employees and departments) into Workforce Planning.

   Load Account, Entity, Employee, and user-defined dimensions from any flat file or ODBC-compliant database. Load members, shared members, and attribute values into dimensions. Hyperion recommends that you first load a small sample of employees. Verify the results, make any needed changes, and load the entire dimension.

   See Appendix A, “Workforce Planning Structure” for a list of Workforce Planning accounts and their properties.

2 Refresh the application to update the Essbase outline.

3 Set up the data load dimension and dimension driver members.
The data load dimension is the dimension to which you are loading data, and corresponds to the target table in the DIM Adapter for Planning and to the method in Application Link Adapter for Hyperion Planning – System 9 Online Help. The driver dimension is the dimension to which you are loading data in an Essbase database. See “Loading Data” in Hyperion Planning – System 9 Administrator’s Guide.

4 Load data and employee properties into the Essbase database for the Workforce Planning application.

The outlines must match; you can specify only the members and parent member names.

5 Refresh the application to update the data.

Each time you modify the application structure, reload the data.

Note:

Load fixed accounts only into the BegBalance member and load varying accounts into all time periods. You set the effective POV for each record using the POV port.

About Loading Information Using DIM Adapter for Planning

After you install and configure DIM Adapter for Planning, you can install and configure adapters that enable you to retrieve and write data for other Hyperion products. After you configure an adapter, you must configure an application connection in Workflow Manager before you can extract data from sources or write data into targets. See the Hyperion Planning – System 9 Administrator’s Guide, “Loading Members and Data” chapter. For specifics on loading workforce-related information into Planning, see the DIM Adapter for Planning Online Help.

Note:

You can load information using Oracle’s Hyperion® Data Integration Management Adapter for Planning only into Classic applications.

About Loading Information Using Application Link Adapter for Hyperion Planning

Application Link Adapter for Hyperion Planning provides a graphical user interface for creating integrations from your source system into Planning. It includes a wizard—Oracle’s Hyperion® Translation Manager—that helps you map your source data into Planning equivalents. For more information about Application Link Adapter for Hyperion Planning, see the Hyperion Planning – System 9 Administrator’s Guide, “Loading Members and Data” chapter. For specifics on loading workforce-related information into Planning, see the Application Link Adapter for Hyperion Planning Online Help.

Note:

You can load information using Application Link Adapter for Hyperion Planning only into Classic applications.
Logging On and Accessing Workforce Planning

You set up Workforce Planning in the Oracle’s Hyperion® Workspace environment. The default Workspace URL is http://<web server>:<port>/workspace/, where <web server> is the Web server computer hostname and <port> is the Web server listen port. For information on installing and configuring Workspace, see the Hyperion Reporting and Analysis – System 9 Installation Guide and Hyperion Workspace Administrator’s Guide.

To log on to Workspace and access Performance Management Architect (for Performance Management Architect application administration), Planning, and Workforce Planning:

1. Ensure that the Web server is started and the Web application server is running in the Services panel.
2. In the Web browser, enter the URL for the Workspace Log On page.
3. Enter your system user name.
4. Enter your system password.
5. Click Log On.
6. For Performance Management Architect applications: To access Performance Management Architect, select an option from the Navigate menu.

   For example, select Navigate > Administer, then select Dimension Library or Application Library. For information about Performance Management Architect, see the Hyperion Enterprise Performance Management Architect Administrator’s Guide or online help.

7. Select Navigate > Applications > Planning, and select an application.
8. To work with data forms:
   a. To view and work with data forms: Select Data Form, expand Forms, and select Workforce.
   b. To manage data forms: Select Administration > Manage Data Forms. From the Data Forms folder, select Workforce, select a data form, and click Edit.
Working with Workforce Planning

In This Chapter

- Working with Employees ................................................................. 19
- Planning Department Expenses ................................................... 22
- Working with Accounts ................................................................. 22
- Working with Data Forms ............................................................... 24
- Working with Smart Lists and Menus .............................................. 24
- Working with Member Formulas ..................................................... 25
- Working with Business Rules .......................................................... 25
- Working with UDAs ........................................................................ 26
- Managing Access Permissions ......................................................... 27
- What’s Next? ................................................................................ 27

Working with Employees

This section helps you understand Workforce Planning employee tasks. To change employee status, planners can click cells in data forms and select items from Smart Lists or menus. They can also run business rules, for example, to transfer employees into new departments.

Employee properties are account members such as Position, Grade, or Salary. Employee members can have associated Smart Lists, and they may depend on another employee property. For example, the value in the FT/PT (full time/part time) account member depends on the value entered in the FTE (full-time equivalent) account. You can customize the employee properties as described in Chapter 3, “Customizing Workforce Planning.” For example, you can change the employee type, grade, FTE, status, and performance that drive calculations.

Transferring Employees

Planners transfer employees using business rules. Depending on your business needs, employees can either be transferred in two steps, using Transfer Out and Transfer In, or in one step, using Transfer.

The two-step transfer process provides security so that a manager in Department A cannot see member data for Department B, a department for which they do not have access. The owner of Department A should transfer out an employee during the same month that the owner of Department B transfers in the employee. To use the two-step process, planners launch these business rules:
- Transfer Out—Transfers employees out of a department
- Transfer In—Transfers employees into a new department

Planners can use a one-step transfer process if security is not an issue. The Transfer business rule transfers employees out of one department and into a new department.

To undo the action for Transfer Out or Transfer In, planners can change the Action value in the data form back to the desired value. In most cases, the desired value is #missing. After an employee is transferred, employee data stays in the old department, but salary is not calculated for that department. When the employee is transferred to the new department, the status is set to Active for that department and the employee’s salary is calculated in the new department.

To undo the action for Transfer, planners can launch the rule again, using the same settings but reversing the order of the departments for transfer.

The Transfer In business rule clears the Action account before the transfer in month. The Transfer Out business rule clears this account after the transfer out month. While no other information is changed or cleared, other accounts are based on the Action account, so transfers can cause a ripple effect after other accounts are calculated. For example, Action drives Status, which drives Salary, which drives Total Salary, which drives Taxes and Total Compensation. Total Compensation + Taxes drives Expenses, and so on.

Planners can transfer an employee into a department, such as Development, even if the employee is already in the department. This is useful when planners want to move the transfer-in date.

**Tip:**

Hyperion recommends that planners review the Reconcile Transfer Report before approving a plan. See “Reconciling Transfers” on page 20.

You can create task lists to add workflow to the transfer process to suit your business needs. For example, tasks can send e-mail reminders to department managers in the old and new departments. For information about managing task lists, see the Hyperion Planning – System 9 Administrator’s Guide.

**Reconciling Transfers**

Employees are typically transferred out of one department and into another in the same month. To ensure that the timing of transfers correctly reflects your intentions, Workforce Planning includes two reports that compare employee Transfer Out and In months, enabling you to make corrections.

Before approving plans, open the Reconcile Transfer Report data form and launch the Unreconciled Transfers business rule, which validates that transferred employees’ Transfer In and Transfer Out months match. When Transfer Out and In months are not identical, discrepancies display for employees to which you have access. (If there are no discrepancies, no rows display.) Review displayed rows:
Table 1

<table>
<thead>
<tr>
<th>Displayed Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#missing</td>
<td>No errors (Transfers In equal Transfers Out) for the month</td>
</tr>
<tr>
<td>-1</td>
<td>Transfers Out exceed Transfers In for the month</td>
</tr>
<tr>
<td>0</td>
<td>Too many transfers in the month (Transfers In equal Transfers Out but the number of transfers is not 1)</td>
</tr>
<tr>
<td>1</td>
<td>Transfers In exceed Transfers Out in the month</td>
</tr>
</tbody>
</table>

**Note:**
You cannot transfer an employee into and out of the same department in the same month.

For example, if two employees are transferred out of Department A in January, but aren’t transferred to any department until March, their row’s January cells display -1 and their March cells display 1.

**Tip:**
Create a Smart List to display meaningful text instead of numerical values. See the *Hyperion Enterprise Performance Management Architect Administrator’s Guide* or help system.

To make corrections, right-click on an individual’s name, and select the menu item to launch the Reconcile Employee Transfer report, where you can change transfer months.

**Increasing the Maximum Number of TBHs**

Workforce Planning includes 30 TBH (To Be Hired) employee members for each department, so departments can add 30 new hires during the current plan year, 30 more for the next year, and so on for all subsequent years. You can determine the number of TBHs to allow for each department. To add TBHs, load them with Application Link Adapter for Hyperion Planning or add them on the Dimensions tab.

**Converting TBHs to Hires**

After TBHs are hired, planners convert the TBH members to the names of newly-hired employees. They create a member for the new employee’s name and run the Reconcile TBH business rule. This reconciles the data for the TBH member for the department to the actual new hire.
Changing Employee Status

To change an employee’s status, planners use a data form to select a new value for Status, such as Maternity Leave or Departed. They can also use the Change Employee Status business rule.

Terminating Employees

Planners can terminate employees by using the Plan Departure business rule, or by changing the employee’s Status in the data form to Departed or Resigned. If necessary, bonus payments can be made after employees are terminated.

After an employee is terminated, planners can undo the action in the data form by clicking the cell that contains the Departed value, and pressing Delete to remove the value. If an employee was on a non-active status such as Sabbatical when they were terminated, planners must manually change the status after undoing a termination.

Planning Department Expenses

You can plan expenses by individuals or departments. Department expenses are not associated with specific employees, and include items such as printers, telephone, and furniture. You can use the Department General employee member to plan for expenses charged to departments. If you use this member, pay special attention to taxes.

Tip:

You can plan turnover with the Turnover data form.

Working with Accounts

You can add new accounts that roll up into aggregates. You can also associate accounts with calculations by adding member formulas or business rules for them. See “Predefined Accounts” on page 53.

Spreading Data Using System Members

You can use the account members under System Members to spread data for calculations in Workforce Planning. You can spread driver members across time, including annual accounts such as yearly salary, or member values based on another member.

To spread data, you can use:

- A predefined spread type, such as Spread_Average, Balance, First, Flow, 445, 454, Actual_365, and Actual_Actual. If you delete the spread type members, calculations based on them do not work.

Example: Salary = “Annual Salary” * Spread_445
The time balance spread formula expression, \([\text{TimeBalanceSpread}]\). This formula expression returns the spread type member based on the account’s Time Balance spreading property. It relies on the predefined System Members, which are only available to Workforce Planning applications. Hyperion recommends using time balance spread formula expressions so that changes to your outline do not affect Workforce Planning calculations.

Example: \(\text{Salary} = "\text{Annual Salary}" \times [\text{TimeBalanceSpread}]\)

**About Salary Accounts**

Salary accounts include members that indicate how much employees make, how much they are paid, and modifying factors such as Status = Departed, Salary Basis = Hourly, or FTE (full-time employment rating) = 0.50.

Annual Salary is calculated:

**Table 2  Annual Salary Calculation**

<table>
<thead>
<tr>
<th>Salary Basis</th>
<th>Pay Type</th>
<th>Salary Rate</th>
<th>Annual Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hourly</td>
<td>Exempt</td>
<td>Amount the employee makes per hour</td>
<td>Salary Rate \times \text{Number of hours worked per week} \times 52</td>
</tr>
<tr>
<td>Hourly</td>
<td>NonExempt</td>
<td>Amount the employee makes per hour</td>
<td>Salary Rate \times \text{lesser of:} &lt;\text{Number of hours worked per week or 40 hours}&gt; \times 52 weeks</td>
</tr>
<tr>
<td>Annual</td>
<td>NA (Pay Type is considered only for Hourly employees)</td>
<td>Amount the employee makes per year</td>
<td>Salary Rate \times \text{FTE}</td>
</tr>
</tbody>
</table>

Conceptually, the Employee Expenses member is calculated thus, using hypothetical rounded numbers:

**Table 3  Model of Salary Calculations**

<table>
<thead>
<tr>
<th>Type of Value</th>
<th>Input (I), Calculated (C), or Either (I/C)</th>
<th>Member Name</th>
<th>Example Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Salary Rate</td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>FTE</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Annual Salary</td>
<td>75,000</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> See Table 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Status</td>
<td>Maternity (65% = 48,000)</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Salary</td>
<td>4,000 (48,000 / 12)</td>
<td></td>
</tr>
<tr>
<td>I/C</td>
<td>Merit Adjustment %</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>Type of Value</td>
<td>Member Name</td>
<td>Example Amount</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Merit</td>
<td>200 (4,000 * 5%)</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Adjusted Salary</td>
<td>4,200 (4,000 + 200)</td>
<td></td>
</tr>
<tr>
<td>I/C</td>
<td>Bonus %</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Bonus (annual total, based on two 6-month bonuses)</td>
<td>4,800</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Total Salary</td>
<td>9,000 (4,200 + 4,800)</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Health Plan</td>
<td>Family Plan</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Health Care Cost</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Total Compensation</td>
<td>9,300 (9,000 + 300)</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Taxes</td>
<td>2,000</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Employee Expenses</td>
<td><strong>11,300</strong></td>
<td></td>
</tr>
</tbody>
</table>

Some of the values are input by planners, and others are calculated by Workforce Planning. For values listed as I/C in the table, planners can accept the default value or override it. For example, a planner can override the calculated Bonus % to enter a higher value if an employee far exceeds expectations.

Essbase calculates members in the order of the outline. If you move accounts in the outline, you must be careful about dependencies. Likewise, if you rewrite the logic for the Salary calculations, you may need to reorder the outline. For a list of expense members and the outline hierarchy, see “Predefined Accounts” on page 53.

### Working with Data Forms

When you make changes to data forms in your model, be sure that changes are synchronized with business logic (such as business rules, member formulas, and outline structure). Whenever you modify your business logic, you should check your data forms.

### Working with Smart Lists and Menus

After you enter data based on Smart List values, any calculations based on the Smart Lists may not be correct if the lists are changed. You may need to adjust data in the data base to reflect the new values. To understand the implications of changing Smart Lists, see the examples in “Customizing Workforce Planning” on page 29. You can delete predefined menus without impacting predefined calculations.
For Smart Lists and menus, Workforce Planning provides predefined labels that are set as resource strings in the HspCustomMsgs template file. For a list of predefined Smart Lists and menus, see “Predefined Smart Lists” on page 86 and “Predefined Menus” on page 88.

If you add Smart Lists and menus to your application, you should add the labels to the HspCustomMsgs file. You should also update the file whenever you modify the predefined Smart Lists or menus.

You can localize the labels in your application by updating the localized versions of the HspCustomMsgs file. For instructions on modifying labels or preparing files for localization, see Chapter 14, “Customizing Planning Web Client” in the Hyperion Planning – System 9 Administrator’s Guide.

**Note:**

Modifying the HspCustomMsgs file overrides default values installed with Workforce Planning or creates new values.

---

**Working with Member Formulas**

Workforce Planning member formulas include formula expressions, such as [TimeBalanceSpread]. Hyperion recommends that you include formula expressions when you write new formulas, making your formulas independent of your outline, so calculations still work if you change the outline. See “Predefined Member Formulas” on page 59.

See the Hyperion Enterprise Performance Management Architect Administrator’s Guide or help system.

---

**Working with Business Rules**

Planners use business rules to perform calculations on employee data, such as:

- Adding new employees to departments. For example, when employees are hired, planners use the Add TBH Hourly or Add TBH Salary business rule to add new hires to departments.
- Changing employee status. For example, when employees take a leave of absence, planners use the Change Employee Status business rule to change the employee's status from Active to Leave of Absence.
- Transferring employees in and out of departments. For example, when employees take jobs in other departments, planners use the Transfer business rule to transfer employees out of the old departments and into the new departments.

See “Predefined Business Rules” on page 90.

Administrators work with business rules in the Administration Services Console (Administration Console). You can use the Administration Console to create, validate, launch, and manage business rules. You can also view the content of predefined Workforce Planning business rules. (See the Hyperion Business Rules Administrator’s Guide.) In Planning, you associate business rules to menus and data forms, and set properties for the business rules.
Planners and other users launch the business rules to set and calculate employee data on data forms. (See the Hyperion Planning – System 9 User’s Guide.)

**Using Business Rules to Roll Up Higher Level Entities**

To roll up upper-level entities (for example, you select Division instead of Department on the page drop-down of the 8. Headcount and FTE data form), create a new business rule similar to the predefined Rollup business rule and associate it with the data form. In the new business rule, include an additional rollup of the Entity dimension. If your application is multi-currency, create a new rule in Oracle's Essbase® Administration Services, copy and paste the following text, and save the new business rule:

```plaintext
SET AGGMISSG ON;
FIX ([Scenario], [Version], [Year], "Local", "HSP_InputValue")
CALC DIM ("Account", "Period", "Employee", "Entity");
ENDFIX
```

**Note:**

If your application uses a single currency, remove “Local” from the FIX statement. If you have renamed dimensions, the CALC DIM statement should list those dimension names instead.

Run the Rollup business rule instead of the Calc Data on Form calc script on Workforce Planning data forms that contain accounts with member formulas (such as 7. Employee Expenses Summary form or 8. Headcount and FTE form). These data forms already have the Rollup business rule associated with them. On other customized data forms, you may need to associate this rule with the data form during data form design.

**Working with UDAs**

You can use user-defined attributes (UDAs)—words or phrases describing a particular characteristic of members—within calc scripts and reports so they return lists of members associated with the specified UDA. Workforce Planning includes one predefined UDA called ACTUAL. This UDA flags the Actual scenario and ignores member formulas. This prevents actual data from being overwritten.

**Note:**

Member formulas do not calculate data in any scenario that has the ACTUAL UDA assigned. Before or after initializing Workforce Planning, assign the ACTUAL UDA to the scenario that is used to store actuals data.

**Managing Access Permissions**

You control access to Workforce Planning—who can view, add, delete, and change information, and to which departments they have access. See how to:

- Set up users and groups in the *Hyperion Security Administration Guide*
- Assign access in “Setting Up Access Permissions” and “Assigning Access to Members” in the *Hyperion Planning – System 9 Administrator’s Guide*

Before planners can add and modify information for a department, they must have access to the department and the Department General entity member.

If you give planners access to a member, they also have access to that member’s data. Because salary information is sensitive, you can hide it from viewing and reporting by denying users or groups access—the access type of None—to members or to their parents. You can also screen information by withholding access to certain data forms.

**What’s Next?**

After setting up your application and loading employee data, you can:

- Set up task lists for planners. To help guide users in the planning process, administrators can set up task lists. The task list shows the activities for which users are responsible and due dates for tasks. It might also include instructions for completing the tasks. See the *Hyperion Planning – System 9 Administrator’s Guide*.
- Assess rolling forecasts by loading prior payroll actual data (typically from the General Ledger and HR system) into the history time periods of the Forecast scenario (typically, by using Oracle's Hyperion® Application Link Adapter for Hyperion Planning). Historical information such as prior month actuals remains intact, enabling managers to compare actual results to the same month’s forecast, the same month’s budget, and the prior month’s actuals.
- Customize your Workforce Planning application to accommodate any of your company’s special requirements. See “Customizing Workforce Planning” on page 29.
About Customizing Your Application

This chapter provides examples to help you understand how to customize Workforce Planning applications. Before customizing a Workforce Planning application, evaluate the predefined Workforce Planning model—described in Appendix A, “Workforce Planning Structure”—to identify which elements you can use and which you need to customize. Study the logic, formulas, and interrelationships of the predefined members.

Whenever you change the Workforce Planning model, validate the model to test modifications and refresh the application.

Adding a Performance Rating

In this example, we:

- Add a new Performance rating called Star Performer to the Smart List called Performance.
- Update its associated member formulas to calculate a merit increase and bonus of 20% each.

You can customize other Smart Lists and member formulas in the same manner. In this example, no business rules are affected.

Tip:

Smart Lists are used in predefined member formulas, and drive the calculations of the Workforce Planning model. Review how Smart List entries affect member formulas so that you understand the effects of changing or adding a Smart List.
To add Star Performer as a member of the Performance Smart List dimension in Performance Management Architect applications:

1. In Performance Management Architect, select Navigate > Administer > Dimension Library.
2. In Master View, select the Performance Smart List to displays its properties.
3. Right-click Performance, and select Create Member > As Child.
4. Enter the name Star Performer, and click OK.
   Star Performer is displayed as a Performance Smart List member.
5. In the Star Performer properties, in the Value column, enter Star Performer as the label. Leave all other default values.

Tip:
Labels are displayed in the user interface. Labels can either be text or they can reference a resource variable by name. For example, to set a label to Star Performer, either set it to Star Performer directly or set it to the name of a resource—such as LABEL_STAR_PERFORMER—which may be localized. For information on localizing labels, see the Hyperion Planning – System 9 Administrator’s Guide.
6. Click Save.
   The Performance Smart List dimension now includes a new performance rating called Star Performer on which you can base other calculations.
7. In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.

Note:
The Redeploy option is only available after an application is successfully deployed.
8. Select the Shared Services Project.
   The Shared Services Project is only available after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Performance Management Architect, or if you register the Application View using Shared Services.
9. Select Refresh Outline, and select additional security options, as necessary.
10. Click OK.
    The application is redeployed to Planning. For more information on deploying applications, see the Hyperion Enterprise Performance Management Architect Administrator’s Guide or help system.

Add a 20% merit increase calculation and 20% bonus calculation based on Star Performer performance ratings. To do this, we extend the existing member formulas:

1. In Performance Management Architect, select Navigate > Administer > Dimension Library.
2 From the Master View, expand WFP-Accounts, WFP-Assumptions, and Compensation Assumptions to select the member Rec. Merit %.

3 In the Rec. Merit % properties, in the Value column, double-click Member Formula. Then click the ellipsis button to display and edit its value.

4 In the Memo Editor, to assign a 20% merit increase to employees with a Star Performer rating, add the following to the member formula:

```
ELSEIF ("Performance" == [Performance.StarPerformer])
0.20;
```

The member formula now looks like this:

5 Click Validate to validate the member formula.

Note:
You can only validate a member formula after you deploy the Workforce Planning application to Planning. See the Hyperion Enterprise Performance Management Architect Administrator’s Guide.

6 Click OK to save the formula.

7 Make similar changes to the Rec. Bonus % member formula.

8 Click Save.

9 In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.

Note:
The Redeploy option is only available after an application is successfully deployed.

10 Select the Shared Services Project.

The Shared Services Project is only available after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Performance Management Architect, or if you register the Application View using Shared Services.

11 Select Refresh Outline, and select additional security options, as necessary.
With Classic Applications

To add Star Performer as a Smart List entry for Performance in Classic applications:

1. Select Administration > Manage Smart Lists.
2. Select Performance and click Edit.
3. Select Entries and click Add.
4. Enter the following:
   - In the Name column, enter: StarPerformer
   - In the Label column, enter: Star Performer

Tip:
Labels are displayed in the user interface. Labels can either be text or they can reference a resource variable by name. For example, to set a label to Star Performer, either set it to Star Performer directly or set it to the name of a resource—such as LABEL_STAR_PERFORMER—which may be localized. See the “Customizing Planning Web Client” chapter in the Hyperion Planning – System 9 Administrator’s Guide.

5. Click Save.

The Smart List named Performance now includes a new performance rating called Star Performer on which you can base other calculations.

Now add a 20% merit increase calculation and 20% bonus calculation based on Star Performer performance ratings. To do this, we extend the existing member formulas:

1. Select Administration > Dimensions and select the Account dimension.
2. Under Compensation Assumptions, select the Rec. Merit % member.
3. Click Edit and select Member Formula.
   - Notice that the member formula specifies merit increases ranging from 0 to 10%, based on the different performance ratings.
4. To assign a 20% merit increase to employees with a Star Performer rating, add the following to the member formula:
   ```python
   ELSEIF ("Performance" == [Performance.StarPerformer])
   0.20;
   ```
   - The member formula now looks like this:
5 Click Validate Member Formula, OK, and Save.

6 Make similar changes to the Rec. Bonus % member formula.

7 Refresh the application to update the outline in Essbase.

### Changing the Pay Rate for Maternity Status

In this example, we change the salary percentage that employees on Maternity status are paid. The current model pays at 65%, and we will change it to 80%. To do this, we modify the member formula for the Salary account member.

### With Performance Management Architect Applications

To change the rate of pay for Maternity status from 65% to 80% in Performance Management Architect applications:

1 In Performance Management Architect, select Navigate > Administer > Dimension Library.

2 From the Master View, expand WFP-Accounts, Employee Expenses, Total Compensation, Total Salary, and Adjusted Salary to select the member Salary.

3 In the Salary properties, in the Value column, double-click Member Formula. Then click the ellipsis button to display and edit its value.

The part of the member formula that calculates the pay rate for Maternity status looks like this:
4 In the Memo Editor, change .65 to .80:

```
ELSEIF ("Status" == [Status.Maternity])
0.80 * "Annual Salary" * [TimeBalanceSpread];
```

5 Click Validate to validate the member formula.

Note:

You can validate a member formula only after you deploy the Workforce Planning application to Planning.

6 Click OK.

Now employees whose Status is Maternity are paid 80% of their salary each month, until their status changes. You must calculate the employee data before the data reflects this change.

7 In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.

Note:

The Redeploy option is available only after an application is successfully deployed.

8 Select the Shared Services Project.

The Shared Services Project is available only after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Performance Management Architect, or if you register the Application View using Shared Services.

9 Select Refresh Outline, and select additional security options, as necessary.

10 Click OK.

The application is redeployed to Planning.
With Classic Applications

➤ To change the rate of pay for Maternity status from 65% to 80% in Classic applications:

1. Select Administration > Dimension and select the Account dimension.
2. Select the member Salary.
3. Click Edit and select the Member Formula tab.

The part of the member formula that calculates the rate of pay for Maternity status looks like this:

4. Change .65 to .80:

   ELSEIF ("Status" == [Status.Maternity])

   0.80 * "Annual Salary" * [TimeBalanceSpread];

5. Click Save.

Now employees whose Status is Maternity are paid 80% of their salary each month, until their status changes.

6. Refresh the application to update the outline in Essbase.

Note:

You must calculate the employee data before the data reflects this change.

Changing the Frequency of Bonuses

This example shows how to change the frequency of bonuses from twice a year (March and September) to quarterly (March, June, September, and December). To do this, we:

- Add the June and December members to the Compensation Adjustment data form.
- Extend the member formula for the Bonus Basis member to include June and December.
Change the Compensation Adjustments data form to reflect the new bonus months:

1. In Planning, select Administration > Manage Data Forms, and select the Compensation Adjustments data.
2. Click Edit and select the Row/Column Layout tab.
3. Under Column Dimension(s), select Edit Columns.
4. To the right of Column Definition 2, click Edit.

5. Click for the Period dimension.

6. Expand YearTotal, Q2 and Q4, and select Jun and Dec.

7. Reorder the members so they display in this order:

8. Click Submit and Submit again at the Row/Column Layout page. Save the data form.

The Compensation Adjustments data form now reflects the new bonus months.

Next we change the member formula for the Bonus Basis member to include calculations for the new bonus months. “With Performance Management Architect Applications” on page 30

With Performance Management Architect Applications

After adding the June and December members to the Compensation Adjustment data form, described in “Changing the Frequency of Bonuses” on page 35.
To change the member formula for the Bonus Basis member in Performance Management Architect applications:

1. In Workspace, select Navigate > Administer > Dimension Library.

2. From the Master View, expand WFP-Accounts, WFP-Assumptions, Compensation Assumptions, and Bonus Basis to select the member Bonus Basis.

3. In the Bonus Basis properties, in the Value column, double-click Member Formula. Then click the ellipsis button to display and edit its value.

The member formula currently allows bonuses for employees whose type is Regular (as set by the Smart List Employee Type) for the first and third Quarter:

```
[OpenInputValueBlock]
IF ("Employee Type" == [EmployeeType.Regular] AND (@ISMBR("Mar") OR @ISMBR("Jun") OR @ISMBR("Sep") OR @ISMBR("Dec")))
3;
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]
```

4. **Extend and save the member formula to review bonuses every three months instead of six:**

```
[OpenInputValueBlock]
IF ("Employee Type" == [EmployeeType.Regular] AND (@ISMBR("Mar") OR @ISMBR("Jun") OR @ISMBR("Sep") OR @ISMBR("Dec")))
3;
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]
```

Notice that the 6 is now 3, indicating that the formula looks to the last three months instead of six.

5. **Click Validate** to validate the member formula.

 **Note:**

You can validate a member formula only after you deploy the Workforce Planning application to Planning. See the Hyperion Enterprise Performance Management Architect Administrator’s Guide.

6. **Click OK.**

Planners can now input bonus values into the Compensation Adjustments data form quarterly instead of semi-annually. You must calculate the employee data before the data reflects this change.

7. **In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.**
Note:
The Redeploy option is available only after an application is successfully deployed.

8 Select the Shared Services Project.

The Shared Services Project is only available after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Performance Management Architect, or if you register the Application View using Shared Services.

9 Select Refresh Outline, and select additional security options, as necessary.

10 Click OK.

The application is redeployed to Planning.

With Classic Applications

➤ To change the member formula for the Bonus Basis member in Classic applications:

1 In Planning, select Administration > Dimensions and select the Account dimension.

2 Select the Bonus Basis member, click Edit, and select the Member Formula tab.

3 The member formula currently allows bonuses for employees whose type is Regular (as set by the Smart List Employee Type) for the first and third Quarter:

```
[OpenInputValueBlock]
IF ("Employee Type" == [EmployeeType.Regular] AND (@ISMBR("Mar") OR @ISMBR("Jun") OR @ISMBR("Sep") OR @ISMBR("Dec"))) THEN
3;
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]
```

4 Extend and save the member formula to review bonuses every three months instead of six:

```
[OpenInputValueBlock]
IF ("Employee Type" == [EmployeeType.Regular] AND (@ISMBR("Mar") OR @ISMBR("Jun") OR @ISMBR("Sep") OR @ISMBR("Dec"))) THEN
3;
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]
```

Notice that the 6 is now 3, indicating that the formula looks to the last three months instead of six.

5 Refresh the application to update the outline in Essbase.

Planners can now input bonus values into the Compensation Adjustments data form quarterly instead of semi-annually.
Adding a Salary Adjustment Type

In this example, we add a new salary adjustment type: an equity adjustment, which enables managers to adjust employees’ salaries by a specified percent to correct their being significantly under- or over-paid. We do this by:

- Creating input members for equity month and adjustment percent.
- Adding the input members to the Compensation Adjustments data form, into which planners select the effective month (using a Smart List) and enter the equity percent change (usually an increase).
- Adding members that calculate the adjusted salary as of the effective month.

With Performance Management Architect Applications

To add the Equity Month and Equity Adjustment % input members in Performance Management Architect applications:

1. In Performance Management Architect, select Navigate > Administer > Dimension Library.
2. In the Master View, expand WFP-Accounts, WFP-Assumptions, and Employee Properties to select the member Merit Month.
3. Right-click Merit Month, and select Create Member > As Sibling.
4. Enter the name, Equity Month, for the new member, and click OK.
   
   Equity Month is inserted below Merit Month in the Master View. Note that we are associating this member with the Smart List named Month.

Note:
Whenever you create members, consider assigning access permissions to them based on functions or department.

5. In the Equity Month properties, under the Value column, set these properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
</tbody>
</table>
In the Master View, expand WFP-Accounts, WFP-Assumptions, and Compensation Assumptions to select the member Merit Adjustment %.

Right-click Merit Adjustment %, and select Create Member > As Sibling.

Enter the name, Equity Adjustment %, for the new member, and click OK.

Equity Adjustment % is inserted below Merit Adjustment % in the Master View.

In the Equity Adjustment % properties, under the Value column, set these properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Exchange Rate Type</td>
<td>None</td>
</tr>
<tr>
<td>Data Type</td>
<td>Percentage</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store</td>
</tr>
<tr>
<td>Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Source Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

Click Save.

In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.

Note:
The Redeploy option is available only after an application is successfully deployed.

Select the Shared Services Project.
The Shared Services Project is available only after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Performance Management Architect, or if you register the Application View using Shared Services.

13 Select Refresh Outline, and select additional security options, as necessary.
14 Click OK.

The application is redeployed to Planning.

➤ To add the new input members—Equity Month and Equity Adjustment %—to the Compensation Adjustments data form:

1 In Planning, select Administration > Manage Data Forms, and select the Compensation Adjustments data form.
2 On the Row/Column Layout tab, under Column Dimension(s), to the right of the Account members, select
3 Select the Equity Month and Equity Adjustment % members and order them like this:
4 Save the data form.

➤ Add the calculated members: Equity Adjustment % Cum and Equity:

1 In Oracle's Hyperion® Workspace, select Navigate > Administer > Dimension Library.
2 In the Master View, expand WFP-Accounts, WFP-Assumptions, and Compensation Assumptions to select the member Equity Adjustment %.
3 Right-click Equity Adjustment %, and select Create Member > As Sibling.
4 Enter the name, Equity Adjustment % Cum, for the new member, and click OK.
   Equity Adjustment % Cum is inserted below Equity Adjustment % in the Master View.
5 Set these properties for Equity Adjustment % Cum:
### Table 6  Equity Adjustment % Cum Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Exchange Rate Type</td>
<td>None</td>
</tr>
<tr>
<td>Data Type</td>
<td>Percentage</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Dynamic Calc</td>
</tr>
<tr>
<td>Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Source Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

6. **Save the properties.**

7. **Double-click Member Formula.** Then click the ellipsis button to display and edit its value.

8. **In the Memo Editor, define and save the formula for Equity Adjustment % Cum:**

   ```plaintext
   [OpenInputValueBlock]
   IF ("Cal TP-Index" == 1 AND "BegBalance"->"Equity Month"<= [Month.Jan])
       "BegBalance"->"Equity Adjustment %";
   ELSEIF ("Cal TP-Index" == 2 AND "BegBalance"->"Equity Month"<= [Month.Feb])
       "BegBalance"->"Equity Adjustment %";
   ELSEIF ("Cal TP-Index" == 3 AND "BegBalance"->"Equity Month"<= [Month.Mar])
       "BegBalance"->"Equity Adjustment %";
   ELSEIF ("Cal TP-Index" == 4 AND "BegBalance"->"Equity Month"<= [Month.Apr])
       "BegBalance"->"Equity Adjustment %";
   ELSEIF ("Cal TP-Index" == 5 AND "BegBalance"->"Equity Month"<= [Month.May])
       "BegBalance"->"Equity Adjustment %";
   ELSEIF ("Cal TP-Index" == 6 AND "BegBalance"->"Equity Month"<= [Month.Jun])
       "BegBalance"->"Equity Adjustment %";
   ```
ELSEIF ("Cal TP-Index" == 7 AND "BegBalance"->"Equity Month"<= [Month.Jul])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 8 AND "BegBalance"->"Equity Month"<= [Month.Aug])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 9 AND "BegBalance"->"Equity Month"<= [Month.Sep])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 10 AND "BegBalance"->"Equity Month"<= [Month.Oct])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 11 AND "BegBalance"->"Equity Month"<= [Month.Nov])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 12 AND "BegBalance"->"Equity Month"<= [Month.Dec])
"BegBalance"->"Equity Adjustment %";
ELSE
#MISSING;
ENDIF

9 Click Validate to validate the member formula.

**Note:**

You can validate a member formula only after you deploy the Workforce Planning application to Planning. See the *Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

10 Click OK.

11 In the Master View, expand WFP-Accounts, Employee Expenses, Total Compensation, Total Salary, and Adjusted Salary to select the member Merit.

12 Right-click Merit, and select Create Member > As Sibling.

13 Enter the name, Equity, for the new member, and click OK.

Equity is inserted below Merit in the Master View.

14 Set Equity properties:
### Table 7  Equity Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Expense</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Flow</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Exchange Rate Type</td>
<td>Average</td>
</tr>
<tr>
<td>Data Type</td>
<td>Currency</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store</td>
</tr>
<tr>
<td>Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Source Plan Type</td>
<td>Wrkforce</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

15. **Double-click Member Formula.** Then click the ellipsis button to display and edit its value.

16. **In the Memo Editor, enter and save the following formula for Equity:**

```
[OpenInputValueBlock]
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
"Salary" * "Equity Adjustment % Cum";
ENDIF
[CloseInputValueBlock]
```

**Note:**

If you add an equity adjustment account, you must modify the Salary member formula to reflect the equity adjustment.

17. **Click Validate** to validate the member formula.

**Note:**

You can validate a member formula only after you deploy the Workforce Planning application to Planning.

18. **Click OK.**

19. **In the Application View, right-click the application, and select Redeploy to redeploy the application to Planning.**

**Note:**

The Redeploy option is available only after an application is successfully deployed.
20 Select the Shared Services Project.

The Shared Services Project is available only after an Application View is registered in Shared Services. Application Views are automatically registered upon deployment in Oracle's Enterprise Performance Management Architect, or if you register the Application View using Oracle's Hyperion® Shared Services.

21 Select Refresh Outline, and select additional security options, as necessary.

22 Click OK.

The application is redeployed to Planning.

Now planners can input salary adjustments into the Equity Adjustment % member on the Compensation Assumptions data form, select the effective month for the adjustment from a Smart List. The calculated equity value is added to the employee's adjusted salary, starting from the effective month.

With Classic Applications

To add the Equity Month and Equity Adjustment % input members in Classic applications:

1 In Planning, select Administration > Dimensions and select the Account dimension.

2 After Merit Month, add the Equity Month input member and set its properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Data Type</td>
<td>Non-Currency</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store</td>
</tr>
<tr>
<td>Plan Type: Wkforce</td>
<td>Ignore</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>Month</td>
</tr>
</tbody>
</table>

Note that we are associating this member with the Smart List called Month.

Note:
Whenever you create members, consider assigning access permissions to them, based on functions or department.

3 Now we add the Equity Adjustment % input member after Merit Adjustment % and set its properties:
Table 9  Equity Adjustment % Member Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Data Type</td>
<td>Percentage</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store</td>
</tr>
<tr>
<td>Plan Type: Wrkforce</td>
<td>Ignore</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

➢ To add the new input members—Equity Month and Equity Adjustment %—to the Compensation Adjustments data form:

1 In Oracle’s Hyperion® Planning – System 9, select Administration > Manage Data Forms and select the Compensation Adjustments data form.

2 On the Row/Column Layout tab, under Column Dimension(s), to the right of the Account members, select 📏.

3 Select the Equity Month and Equity Adjustment % members and order them like this:

Annual Salary
Pay Type
Salary Basis
Salary Rate
Hours per week
Differential %
Performance
Merit Month
Rec. Merit %
Override Merit %
Merit Adjustment %
“Equity Month”
“Equity Adjustment %”
4 Save the data form.

To add the calculated members: Equity Adjustment % Cum and Equity:

1 After the Equity Adjustment % member, create the Equity Adjustment % Cum member and set its properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Saved Assumption</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Balance</td>
</tr>
<tr>
<td>Skip</td>
<td>None</td>
</tr>
<tr>
<td>Data Type</td>
<td>Percentage</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Dynamic Calc</td>
</tr>
<tr>
<td>Plan Type Wrkforce</td>
<td>Ignore</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

2 Save the properties and, on the Member Formula tab, define and save the formula for Equity Adjustment % Cum:

```
[OpenInputValueBlock]
IF ("Cal TP-Index" == 1 AND "BegBalance"->"Equity Month"<= [Month.Jan])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 2 AND "BegBalance"->"Equity Month"<= [Month.Feb])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 3 AND "BegBalance"->"Equity Month"<= [Month.Mar])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 4 AND "BegBalance"->"Equity Month"<= [Month.Apr])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 5 AND "BegBalance"->"Equity Month"<= [Month.May])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 6 AND "BegBalance"->"Equity Month"<= [Month.Jun])
"BegBalance"->"Equity Adjustment %";
```
ELSEIF ("Cal TP-Index" == 7 AND "BegBalance"->"Equity Month"<= [Month.Jul])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 8 AND "BegBalance"->"Equity Month"<= [Month.Aug])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 9 AND "BegBalance"->"Equity Month"<= [Month.Sep])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 10 AND "BegBalance"->"Equity Month"<= [Month.Oct])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 11 AND "BegBalance"->"Equity Month"<= [Month.Nov])
"BegBalance"->"Equity Adjustment %";
ELSEIF ("Cal TP-Index" == 12 AND "BegBalance"->"Equity Month"<= [Month.Dec])
"BegBalance"->"Equity Adjustment %";
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]

3 Add the Equity member after the Merit member and set and save its properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Expense</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Expense</td>
</tr>
<tr>
<td>Time Balance</td>
<td>Flow</td>
</tr>
<tr>
<td>Exchange Rate Type</td>
<td>Average</td>
</tr>
<tr>
<td>Data Type</td>
<td>Currency</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store</td>
</tr>
<tr>
<td>Plan Type: Wrkforce</td>
<td>Addition</td>
</tr>
<tr>
<td>Smart Lists</td>
<td>None</td>
</tr>
</tbody>
</table>

4 On the Member Formula tab, enter and save the following formula for Equity:
Note:
If you add an equity adjustment account, you must modify the Salary member formula to reflect the equity adjustment.

5 Refresh the application to update the outline in Oracle’s Hyperion® Essbase® – System 9.

Now planners can input salary adjustments into the Equity Adjustment % member on the Compensation Assumptions data form and select the effective month for the adjustment from a Smart List. The calculated equity value is added to the employee’s adjusted salary, starting from the effective month.
In This Appendix

Predefined Data Forms ................................................................. 51
Predefined Accounts ................................................................. 53
Predefined Member Formulas .................................................... 59
Predefined Smart Lists ............................................................. 86
Predefined Smart List Entries ................................................... 86
Predefined Menus ................................................................. 88
Predefined Business Rules ....................................................... 90

This appendix lists the predefined elements of Workforce Planning, helping you determine how elements affect each other.

Predefined Data Forms

Planners use data forms to work with employee information.

Table 12  Predefined Data Forms

<table>
<thead>
<tr>
<th>Data Form Name</th>
<th>Axis Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Manage Existing Employees Status</td>
<td>Row: Employee, Account</td>
</tr>
<tr>
<td></td>
<td>Column: Period</td>
</tr>
<tr>
<td></td>
<td>Page: Entity</td>
</tr>
<tr>
<td></td>
<td>POV: Currency, Scenario, Version, Year</td>
</tr>
<tr>
<td>2. Manage Existing Employees</td>
<td>Row: Employee</td>
</tr>
<tr>
<td></td>
<td>Column: Period</td>
</tr>
<tr>
<td></td>
<td>Page: Entity, Account</td>
</tr>
<tr>
<td></td>
<td>POV: Currency, Scenario, Version, Year</td>
</tr>
<tr>
<td>3. Existing Employee Related Expenses</td>
<td>Row: Account</td>
</tr>
<tr>
<td></td>
<td>Column: Period</td>
</tr>
<tr>
<td></td>
<td>Page: Entity, Employee</td>
</tr>
<tr>
<td></td>
<td>POV: Currency, Scenario, Version, Year</td>
</tr>
<tr>
<td>Data Form Name</td>
<td>Axis Definitions</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Begin vs End</td>
<td>Row: Employee&lt;br&gt;Column: Account, Period&lt;br&gt;Page: Entity&lt;br&gt;POV: Currency, Scenario, Version, Year</td>
</tr>
<tr>
<td>Budget vs Target</td>
<td>Row: Account&lt;br&gt;Column: Version&lt;br&gt;Page: Entity&lt;br&gt;POV: Currency, Employee, Period, Scenario, Year</td>
</tr>
<tr>
<td>Department Level Expenses</td>
<td>Row: Account&lt;br&gt;Column: Period&lt;br&gt;Page: Entity</td>
</tr>
</tbody>
</table>
### Data Form Name | Axis Definitions
--- | ---
**Department Level Headcount** | Row: Account  
Column: Period  
Page: Entity  
POV: Currency, Employee, Scenario, Version, Year

**Reconcile Employee Transfer** | Row: Entity  
Column: Period  
Page: Employee  
POV: Account, Currency, Scenario, Version, Year

**Reconcile Transfer Report** | Row: Employee  
Column: Period  
Page: N/A  
POV: Account, Currency, Entity, Period, Scenario, Version, Year

**Tax Rates** | Row: Account  
Column: Year  
Page: N/A  
POV: Currency, Employee, Entity, Period, Scenario, Version

**Turnover** | Row: Employee, Account  
Column: Period  
Page: Entity  
POV: Currency, Scenario, Version, Year

### Predefined Accounts

Keep in mind the fixed and varying nature of accounts when working with member formulas and loading data. Some accounts store data that may change during the year (for example, an employee’s status, position, and salary), whereas other account data does not usually change during the year (for example, an employee’s health care plan). Varying accounts store their data in the descendants of Year Total. Fixed accounts store their data in the BegBalance member.

When a formula on a fixed member references a varying member, or when a formula on a varying member references a fixed member, you may need to use the CrossDim operator (-->) with the appropriate time period, based on where its data is stored. This calculation provides an example of a varying component (Status) that references a fixed component (Start Month), for determining Status using the CrossDim operator:

```plaintext
[OpenInputValueBlock]

IF (@ISDESC("New Employees"))
```
IF ("BegBalance"->"Start Month" != #MISSING AND "Cal TP-Index">= "BegBalance"->"Start Month")"Status" = [Status.Active];
ENDIF
ELSE
IF ("Action" == [Status.TransferIn])"Status" = [Status.Active];
ELSE"Status" = "Action";
ENDIF
IF("Status" == #MISSING)"Status" = @PRIORS(SKIPMISSING, "Status");
ENDIF
ENDIF
[CloseInputValueBlock]

For information on the CrossDim operator, see the Hyperion Essbase – System 9 Database Administrator’s Guide.

The table shows how Workforce Planning sets up the predefined account members. These accounts are fixed: Merit Month, Start Month, Health Plan, Performance, and Merit Adjustment %. All the other accounts are varying.

<table>
<thead>
<tr>
<th>Table 13 Predefined Accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account</strong></td>
</tr>
<tr>
<td>System Members</td>
</tr>
<tr>
<td>Cal TP-Index</td>
</tr>
<tr>
<td>Fiscal TP-Index</td>
</tr>
<tr>
<td>NumPeriods</td>
</tr>
<tr>
<td>Spread_Average</td>
</tr>
<tr>
<td>Spread_Balance</td>
</tr>
<tr>
<td>Spread_First</td>
</tr>
<tr>
<td>Spread_Flow</td>
</tr>
<tr>
<td>Spread_445</td>
</tr>
<tr>
<td>Spread_454</td>
</tr>
<tr>
<td>Spread_544</td>
</tr>
<tr>
<td>Spread_Actual_365</td>
</tr>
<tr>
<td>Spread_Actual_Actual</td>
</tr>
<tr>
<td>Unreconciled Transfers</td>
</tr>
<tr>
<td>Account</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>WFP-Accounts</td>
</tr>
<tr>
<td>Assumption Input</td>
</tr>
<tr>
<td>SSTax Rate1 Input</td>
</tr>
<tr>
<td>SSTax Rate2 Input</td>
</tr>
<tr>
<td>SSTax Cap Input</td>
</tr>
<tr>
<td>SUI Rate Input</td>
</tr>
<tr>
<td>SUI Cap Input</td>
</tr>
<tr>
<td>FUTA Rate Input</td>
</tr>
<tr>
<td>FUTA Cap Input</td>
</tr>
<tr>
<td>Medicare Rate Input</td>
</tr>
<tr>
<td>WFP-Assumptions</td>
</tr>
<tr>
<td>Employee Properties</td>
</tr>
<tr>
<td>Employee Type</td>
</tr>
<tr>
<td>Pay Type</td>
</tr>
<tr>
<td>FT/PT</td>
</tr>
<tr>
<td>FTE</td>
</tr>
<tr>
<td>Merit Month</td>
</tr>
<tr>
<td>Start Month</td>
</tr>
<tr>
<td>Position</td>
</tr>
<tr>
<td>Grade</td>
</tr>
<tr>
<td>Tax Region</td>
</tr>
<tr>
<td>Health Plan</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>Salary Basis</td>
</tr>
<tr>
<td>Account</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>FUTA Rate</td>
</tr>
<tr>
<td>FUTA Cap</td>
</tr>
<tr>
<td>Medicare Rate</td>
</tr>
<tr>
<td>Total Headcount</td>
</tr>
<tr>
<td>Department Headcount</td>
</tr>
<tr>
<td>Total Regular Headcount</td>
</tr>
<tr>
<td>Regular Headcount</td>
</tr>
<tr>
<td>Departed Headcount</td>
</tr>
<tr>
<td>LOA Headcount</td>
</tr>
<tr>
<td>Maternity Headcount</td>
</tr>
<tr>
<td>On Sabbatical Headcount</td>
</tr>
<tr>
<td>Turnover Headcount Adjustment</td>
</tr>
<tr>
<td>Contractor Headcount</td>
</tr>
<tr>
<td>Temporary Headcount</td>
</tr>
<tr>
<td>Other Headcount</td>
</tr>
<tr>
<td>Total FTE</td>
</tr>
<tr>
<td>Regular FTE</td>
</tr>
<tr>
<td>Contractor FTE</td>
</tr>
<tr>
<td>Temporary FTE</td>
</tr>
<tr>
<td>Other FTE</td>
</tr>
<tr>
<td>Employee Expenses</td>
</tr>
<tr>
<td>Total Compensation</td>
</tr>
<tr>
<td>Total Salary</td>
</tr>
<tr>
<td>Adjusted Salary</td>
</tr>
<tr>
<td>Salary</td>
</tr>
<tr>
<td>Merit</td>
</tr>
<tr>
<td>Overtime</td>
</tr>
<tr>
<td>Bonus</td>
</tr>
<tr>
<td>Account</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Sign On Bonus</td>
</tr>
<tr>
<td>Commissions</td>
</tr>
<tr>
<td>Health Care Costs</td>
</tr>
<tr>
<td>Severance</td>
</tr>
<tr>
<td>Other Compensation</td>
</tr>
<tr>
<td>Turnover Adjustment</td>
</tr>
<tr>
<td>Taxes</td>
</tr>
<tr>
<td>Social Security Tax</td>
</tr>
<tr>
<td>Medicare</td>
</tr>
<tr>
<td>SUI</td>
</tr>
<tr>
<td>FUTA</td>
</tr>
<tr>
<td>Employee Related Expenses</td>
</tr>
<tr>
<td>Training</td>
</tr>
<tr>
<td>Travel</td>
</tr>
<tr>
<td>Telephone</td>
</tr>
<tr>
<td>Office Supplies</td>
</tr>
<tr>
<td>Departure Costs</td>
</tr>
<tr>
<td>Miscellaneous</td>
</tr>
<tr>
<td>Total Employee Capital Purchases</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td>Furniture</td>
</tr>
<tr>
<td>New Hire Related Expenses</td>
</tr>
<tr>
<td>Recruiting</td>
</tr>
<tr>
<td>Relocation</td>
</tr>
<tr>
<td>YTD Accounts</td>
</tr>
<tr>
<td>FYTD Total Salary</td>
</tr>
<tr>
<td>FYTD Total Salary (Prior)</td>
</tr>
</tbody>
</table>
Predefined Member Formulas

Member formulas are used for employee calculations.

Adjusted Annual Salary

Formula

\[
\text{Adjusted Annual Salary} = \text{Annual Salary} \times (1 + \text{Merit Adjustment \% Cum})
\]

Description
Calculates the adjusted annual salary for the period.

Adjusted Annual Salary (Prior)

Formula

\[
\begin{align*}
\text{Adjusted Annual Salary (Prior)} &= \text{IF}\left( \text{ISLEV}\left(\text{Dimension("DIM\_NAME\_PERIOD")}, 0\right)\right) \\text{PRIOR}\left(\text{Adjusted Annual Salary}\right); \\
&\quad \text{ELSE}\text{Adjusted Annual Salary}; \\
\end{align*}
\]

Description
Calculates the adjusted annual salary for the prior period.

Annual Overtime

Formula

\[
\text{IF}\left(\"\text{Salary Basis}\" = \text{[SalaryBasis.Annual]} \text{ OR } \"\text{Salary Basis}\" = \#\text{MISSING} \text{ OR } \"\text{Hours per week}\" \leq 40\right)
\]

```
Calculates the annual overtime for hourly employees.

**Annual Salary**

**Formula**

```
IF ("Salary Basis" == [SalaryBasis.Annual] OR "Salary Basis" == #MISSING)
"Salary Rate" * "FTE";
ELSEIF ("Salary Basis" == [SalaryBasis.Hourly])
IF ("Pay Type" == [PayType.Exempt] OR "Pay Type" == #MISSING)
"Hours per week" * "Salary Rate" * 52;
ELSEIF ("Pay Type" == [PayType.NonExempt])
@MIN("Hours per week", 40) * "Salary Rate" * 52;
ENDIF
ENDIF
```

Calculates the annual salary. Considers FTE factor.

**Bonus**

**Formula**

```
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))"Bonus %" * "Adjusted Salary" * "Bonus Basis";
ENDIF
```

Description
Calculates bonuses based on a bonus percentage and basis.

**Bonus %**

**Formula**

```plaintext
[OpenInputValueBlock]
IF("Override Bonus %" == #MISSING)"Rec. Bonus %";
ELSE"Override Bonus %";
ENDIF
[CloseInputValueBlock]
```

**Description**

Calculates the bonus percentage based on a recommended percentage. Allows bonus percentage overrides.

**Bonus Basis**

**Formula**

```plaintext
[OpenInputValueBlock]
IF ("Employee Type" == [EmployeeType.Regular] AND (@ISMBR([Period("FIRST_QTR_PERIOD")]) OR @ISMBR([Period("THIRD_QTR_PERIOD")])))
6;
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]
```

**Description**

Calculates the bonus basis used to determine bonuses.

**Cal TP-Index**

**Formula**

```plaintext
[OpenInputValueBlock]
[CalendarTPIndex]
[CloseInputValueBlock]
```

**Description**
System member that returns the time period index based on a calendar year. Example: Fiscal year starting in Jul: Jan=1, Feb=2, Mar=3, and so on.

**Contractor FTE**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))

ELSE

#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

**Description**

Calculates Full Time Equivalent (FTE) for Contractor employee types.

**Contractor Headcount**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))

ELSE

#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

**Description**

Calculates Headcount for Contractor employee types.
**CYTD Sign On Bonus**

**Formula**

[OpenInputValueBlock]
[CYTD("Sign On Bonus")]
[CloseInputValueBlock]

**Description**

Calculates the calendar year-to-date total for Sign On Bonus.

**CYTD Total Salary**

**Formula**

[OpenInputValueBlock]
[CYTD("Total Salary")]
[CloseInputValueBlock]

**Description**

Calculates the calendar year-to-date for Total Salary.

**CYTD Total Salary (Prior)**

**Formula**

[OpenInputValueBlock]
IF ("CYTD Total Salary"<= "Total Salary")
#MISSING;
ELSE"CYTD Total Salary" - "Total Salary";
ENDIF
[CloseInputValueBlock]

**Description**

Calculates the calendar year-to-date total for prior period’s Total Salary.

**Departed Headcount**

**Formula**

[OpenInputValueBlock]
IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))

Predefined Member Formulas
IF("Employee Type" == [EmployeeType.Regular] AND ("Status" == [Status.Departed] OR "Status" == [Status.Resigned]))
1;
ELSE
#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

Description
Calculates Headcount for Regular employee type with a Status of Departed.

**Differential %**

**Formula**
[OpenInputValueBlock]
IF (@ISLEV("Employee", 0))
IF ("Salary Basis" == [SalaryBasis.Annual])
("Annual Salary" - "Target Rate") / "Target Rate";
ELSE
#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

Description
Calculates the differential percentage between Target Base Annual salary and the actual Annual salary.

**Fiscal TP-Index**

**Formula**
[OpenInputValueBlock]
[FiscalTPIndex]
[CloseInputValueBlock]

Description
System member that returns the time period index based on a fiscal year. Example: Fiscal year starting in Jul: Jul=1, Aug=2, Sep=3, and so on.

**FT/PT**

**Formula**

[OpenInputValueBlock]

IF ("Salary Basis" == [SalaryBasis.Annual])

IF ("FTE" == #MISSING)

#MISSING;

ELSEIF ("FTE" < 1)

[FT_PT.PT];

ELSE

[FT_PT.FT];

ENDIF

ELSEIF ("Salary Basis" == [SalaryBasis.Hourly])

IF ("Hours per week" < 40)

[FT_PT.PT];

ELSE

[FT_PT.FT];

ENDIF

ELSE

#MISSING;

ENDIF

[CloseInputValueBlock]

**Description**

Defines whether an employee is a full-time or part-time employee.

**FUTA**

**Formula**

[OpenInputValueBlock]

IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO"), "ACTUAL")))

IF ("Tax Region" == [TaxRegion.USA])

IF ("CYTD Total Salary" <= "FUTA Cap") "FUTA Rate" * "Total Salary";
ELSEIF("CYTD Total Salary (Prior)"<= "FUTA Cap")"FUTA Rate" * ("FUTA Cap" - "CYTD Total Salary (Prior)")
ELSE
#MISSING;
ENDIF
ENDIF
ENDIF
[CloseInputValueBlock]

Description
Calculates simple Federal Unemployment Taxes based on a tax region, rate and cap.

FUTA Cap

Formula
[OpenInputValueBlock]
[CrossRef("FUTA Cap Input")]
[CloseInputValueBlock]

Description
Returns a reference to Federal Unemployment Tax (FUTA) cap at a predefined location. This member shields other calculations from knowing where the value is actually stored.

FUTA Rate

Formula
[OpenInputValueBlock]
[CrossRef("FUTA Rate Input")]
[CloseInputValueBlock]

Description
Returns a reference to Federal Unemployment Tax (FUTA) rate at a predefined location. This member shields other calculations from knowing where the value is actually stored.

FYTD Total Salary

Formula
[OpenInputValueBlock]
@SUMRANGE("Total Salary", @CURRMBRRANGE([Dimension("DIM_NAME.PERIOD")], LEV, 0, ,0));
Description
Calculates the fiscal year-to-date for Total Salary.

**FYTD Total Salary (Prior)**

**Formula**

```plaintext
@SUMRANGE("Total Salary", @CURRMBRRANGE([Dimension("DIM_NAME_PERIOD")], LEV, 0, ,-1));
```

Description
Calculates the fiscal year-to-date for prior period’s Total Salary.

**Health Care Costs**

**Formula**

```plaintext
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
IF ("Salary"<> #MISSING) AND ("Employee Type" == [EmployeeType.Regular])
IF("BegBalance"->"Health Plan" == [HealthPlan.IndividualPlan])
100;
ELSEIF("BegBalance"->"Health Plan" == [HealthPlan.IndividualPlus1])
175;
ELSEIF("BegBalance"->"Health Plan" == [HealthPlan.FamilyPlan])
300;
ELSE
#MISSING;
ENDIF
ELSE
#MISSING;
ENDIF
ENDIF
```

[CloseInputValueBlock]
**Description**
Calculates health care costs based on selected health plan.

**LOA Headcount**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")), "ACTUAL") OR @ISMBR ("Department General")))

IF("Employee Type" == [EmployeeType.Regular] AND "Status" == [Status.LeaveOfAbsence])

1;
ELSE
MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

**Description**
Calculates Headcount for Regular employee type with a Status of Leave Of Absence.

**Maternity Headcount**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")), "ACTUAL") OR @ISMBR ("Department General")))

IF("Employee Type" == [EmployeeType.Regular] AND "Status" == [Status.Maternity])

1;
ELSE #MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

**Description**
Calculates Headcount for Regular employee type with a Status of Maternity.
**Medicare**

**Formula**

```plaintext
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
  IF ("Tax Region" == [TaxRegion.USA])"Total Salary" * "Medicare Rate";
ENDIF
ENDIF
```

**Description**

Calculates simple Medicare costs based on a tax region, rate and cap.

**Medicare Rate**

**Formula**

```plaintext
[OpenInputValueBlock]
[CrossRef("Medicare Rate Input")]  
[CloseInputValueBlock]
```

**Description**

Returns a reference to Medicare rate at a predefined location. This member shields other calculations from knowing where the value is actually stored.

**Merit**

**Formula**

```plaintext
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))"Salary" * "Merit Adjustment % Cum";
ENDIF
```

**Description**

Calculates merit increases based on a merit percentage.

**Merit Adjustment %**

**Formula**

```plaintext
[OpenInputValueBlock]
```
IF (@ISMBR("BegBalance"))
IF("Override Merit %" == #MISSING)"Rec. Merit %";
ELSE"Override Merit %";
ENDIF
ELSE
#MISSING;
ENDIF

[CloseInputValueBlock]

Description
Calculates the merit adjustment percentage based on a recommended percentage. Allows merit percentage overrides.

**Merit Adjustment % Cum**

Formula

[OpenInputValueBlock]
IF ("Cal TP-Index" == 1 AND "BegBalance"->"Merit Month"<= [Month.Jan])
"BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 2 AND "BegBalance"->"Merit Month"<=
[Month.Feb]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 3 AND "BegBalance"->"Merit Month"<=
[Month.Mar]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 4 AND "BegBalance"->"Merit Month"<=
[Month.Apr]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 5 AND "BegBalance"->"Merit Month"<=
[Month.May]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 6 AND "BegBalance"->"Merit Month"<=
[Month.Jun]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 7 AND "BegBalance"->"Merit Month"<=
[Month.Jul]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 8 AND "BegBalance"->"Merit Month"<=
[Month.Aug]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 9 AND "BegBalance"->"Merit Month"<=
[Month.Sep]) "BegBalance"->"Merit Adjustment %";
ELSEIF ("Cal TP-Index" == 10 AND "BegBalance"->"Merit Month"<=
[Month.Oct]) "BegBalance"->"Merit Adjustment %";

[CloseInputValueBlock]
ELSEIF ("Cal TP-Index" == 11 AND "BegBalance"->"Merit Month"<= [Month.Nov])"BegBalance"->"Merit Adjustment %";

ELSEIF ("Cal TP-Index" == 12 AND "BegBalance"->"Merit Month"<= [Month.Dec])"BegBalance"->"Merit Adjustment %";

ENDIF

[CloseInputValueBlock]

Description

Calculates the cumulative merit adjustment percentages. Use for calculating merit.

NumPeriods

Formula

[OpenInputValueBlock]

[NumberOfPeriodsInYear];

[CloseInputValueBlock]

Description

System member that returns the number of periods in a year.

On Sabbatical Headcount

Formula

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))

IF("Employee Type" == [EmployeeType.Regular] AND "Status" == [Status.OnSabbatical])

1;
ELSE

#MISSING;

ENDIF

ENDIF

[CloseInputValueBlock]

Description

Calculates Headcount for Regular employee type with a Status of On Sabbatical.
**Other FTE**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))


ELSE

#MISSING;
ENDIF
ENDIF

[CloseInputValueBlock]

**Description**

Calculates Full Time Equivalent (FTE) for Other employee types.

**Other Headcount**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))


ELSE #MISSING;
ENDIF
ENDIF

[CloseInputValueBlock]

**Description**

Calculates Headcount for Other employee types.

**Overtime**

**Formula**
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")), "ACTUAL"))
IF (@ISDESC("New Employees"))
IF ("BegBalance"->"Start Month" != #MISSING)
IF ([FiscalOffset] == 0)
IF ("Cal TP-Index" >= "BegBalance"->"Start Month") "Annual Overtime" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF
ELSEIF ("BegBalance"->"Start Month" - [FiscalOffset]) > 0)
IF ("Fiscal TP-Index" >= ("BegBalance"->"Start Month" - [FiscalOffset])) "Annual Overtime" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF
ELSEIF ("Fiscal TP-Index" >= ("BegBalance"->"Start Month" + [NumberOfPeriodsInYear] - [FiscalOffset])) "Annual Overtime" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF;
ELSE
ELSE
#MISSING;
ENDIF;
ELSE
IF (@ISMBR("BegBalance"))
#MISSING;
ELSE
IF ("Annual Overtime" == #MISSING) OR ("Status" == [Status.Departed])
OR ("Status" == [Status.Resigned]) OR ("Status" == [Status.TransferOut]) OR ("Status" == #MISSING))
#MISSING;
ELSEIF ("Status" == [Status.LeaveOfAbsence])
0;
ELSEIF ("Status" == [Status.Maternity])
0.65 * "Annual Overtime" * [TimeBalanceSpread];
ELSE
"Annual Overtime" * [TimeBalanceSpread];
ENDIF
ENDIF;
ENDIF;
ENDIF;
ENDIF;
[CloseInputValueBlock]

Description
Calculates overtime for new and existing employees. Existing employees’ overtime takes employee status into account.

Rec. Bonus %

Formula

[OpenInputValueBlock]

IF ("Grade"<= 2) "Rec. Bonus %" = 0.10;
ELSEIF ("Grade"<= 4) "Rec. Bonus %" = 0.15;
ELSE "Rec. Bonus %" = 0.20;
ENDIF
IF ("Employee Type" == [EmployeeType.Regular])
IF ("BegBalance"->"Performance" ==
[Performance.FailsToMeetExpectations])
0;
ELSEIF ("BegBalance"->"Performance" ==
[Performance.NeedsImprovement]) "Rec. Bonus %" * 0.50;
ELSEIF ("BegBalance"->"Performance" ==
[Performance.MeetsExpectations]) "Rec. Bonus %";
ELSEIF ("BegBalance"->"Performance" ==
[Performance.ExceedsExpectations]) "Rec. Bonus %" * 1.1;
ELSEIF ("BegBalance"->"Performance" ==
[Performance.FarExceedsExpectations]) "Rec. Bonus %" * 1.25;
ELSE
#MISSING;
ENDIF
ELSE

[OpenInputValueBlock]
IF (@ISMBR("BegBalance"))
IF ("Performance" == [Performance.FailsToMeetExpectations])
    0.00;
ELSEIF ("Performance" == [Performance.NeedsImprovement])
    0.02;
ELSEIF ("Performance" == [Performance.MeetsExpectations])
    0.04;
ELSEIF ("Performance" == [Performance.ExceedsExpectations])
    0.06;
ELSEIF ("Performance" == [Performance.FarExceedsExpectations])
    0.10;
ELSE
    #MISSING;
ENDIF
[CloseInputValueBlock]

Description
Calculates a predefined bonus percentage based on performance.

Rec. Merit %

Formula
[OpenInputValueBlock]
IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR
("Department General")))

Description
Calculates a predefined merit percentage based on performance.

Regular FTE

Formula
[OpenInputValueBlock]
IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR
("Department General")))
ELSE
#MISSING;
ENDIF
[CloseInputValueBlock]

Description
Calculates Full Time Equivalent (FTE) for Regular employee types.

**Regular Headcount**

**Formula**
[OpenInputValueBlock]
IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR ("Department General")))
IF("Employee Type" == [EmployeeType.Regular] AND "Status" == [Status.Active])
1;
ELSE
#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]

Description
Calculates Headcount for Regular employee type with a Status of Active.

**Salary**

**Formula**
[OpenInputValueBlock]
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
IF (@ISDESC("New Employees"))
IF ("BegBalance"->"Start Month" != #MISSING)
IF ([FiscalOffset] == 0)
IF ("Cal TP-Index" >= "BegBalance"->"Start Month") "Annual Salary" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF
ELSEIF ((("BegBalance"->"Start Month" - [FiscalOffset]) > 0))
IF ("Fiscal TP-Index" >= ("BegBalance"->"Start Month" - [FiscalOffset])) "Annual Salary" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF
ELSEIF ("Fiscal TP-Index" >= ("BegBalance"->"Start Month" + [NumberOfPeriodsInYear] - [FiscalOffset])) "Annual Salary" * [TimeBalanceSpread];
ELSE
#MISSING;
ENDIF
ELSE
#MISSING;
ENDIF
ELSE
IF (@ISMBR("BegBalance"))
#MISSING;
#MISSING;
ELSE
IF ("Annual Salary" == #MISSING) OR ("Status" == [Status.Departed]) OR 
("Status" == [Status.Resigned]) OR ("Status" == [Status.TransferOut])
OR ("Status" == #MISSING))
#MISSING;
ELSEIF ("Status" == [Status.LeaveOfAbsence])
0;
ELSEIF ("Status" == [Status.Maternity])
0.65 * "Annual Salary" * [TimeBalanceSpread];
ELSE
"Annual Salary" * [TimeBalanceSpread];
ENDIF
ENDIF
Description
Calculates salary for new and existing employees. Existing employees’ salary takes employee status into account.

Social Security Tax

Formula

[OpenInputValueBlock]

IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
IF ("Tax Region" == [TaxRegion.USA])
IF ("CYTD Total Salary"<= "SSTax Cap")"SSTax Rate1" * "Total Salary";
ELSEIF("CYTD Total Salary (Prior)"<= "SSTax Cap")"SSTax Rate1" *
("SSTax Cap" - "CYTD Total Salary (Prior)") + "SSTax Rate2" * ("CYTD Total Salary" - "SSTax Cap")
ELSE"SSTax Rate2" * "Total Salary";
ENDIF
ENDIF
ENDIF

[CloseInputValueBlock]

Description
Calculates simple Social Security Taxes (FICA) based on a tax region, rate and cap.

Spread_445

Formula

[OpenInputValueBlock]

[TimeBalanceFormula("Spread_445")]

[CloseInputValueBlock]

Description
System member that returns a spread factor following 4 week, 4 week, 5 week quarter.
Spread_454

Formula

[OpenInputValueBlock]
[TimeBalanceFormula("Spread_454")]
[CloseInputValueBlock]

Description

System member that returns a spread factor following 4 week, 5 week, 4 week quarter.

Spread_544

Formula

[OpenInputValueBlock]
[TimeBalanceFormula("Spread_544")]
[CloseInputValueBlock]

Description

System member that returns a spread factor following 5 week, 4 week, 4 week quarter.

Spread_Actual_365

Formula

[OpenInputValueBlock]
[TimeBalanceFormula("Spread_Actual_365")]
[CloseInputValueBlock]

Description

System member that returns a spread factor following a 365-day calendar year.

Spread_Actual_Actual

Formula

[OpenInputValueBlock]
[TimeBalanceFormula("Spread_Actual_Actual")]
[CloseInputValueBlock]

Description

System member that returns a spread factor following the actual days in a calendar year.
**Spread_Average**

**Formula**

[OpenInputValueBlock]

[TimeBalanceFormula("Spread_Average")]

[CloseInputValueBlock]

**Description**

System member that returns a spread factor following the time balance average pattern.

---

**Spread_Balance**

**Formula**

[OpenInputValueBlock]

[TimeBalanceFormula("Spread_Balance")]

[CloseInputValueBlock]

**Description**

System member that returns a spread factor following the time balance last pattern.

---

**Spread_First**

**Formula**

[OpenInputValueBlock]

[TimeBalanceFormula("Spread_First")]

[CloseInputValueBlock]

**Description**

System member that returns a spread factor following the time balance first pattern.

---

**Spread_Flow**

**Formula**

[OpenInputValueBlock]

[TimeBalanceFormula("Spread_Flow")]

[CloseInputValueBlock]

**Description**

System member that returns a spread factor following the normal flow pattern.
**SSTax Cap**

**Formula**

[OpenInputValueBlock]

[CrossRef("SSTax Cap Input")]

[CloseInputValueBlock]

**Description**

Returns a reference to Social Security Tax (FICA) cap at a predefined location. This member shields other calculations from knowing where the value is actually stored.

**SSTax Rate1**

**Formula**

[OpenInputValueBlock]

[CrossRef("SSTax Rate1 Input")]

[CloseInputValueBlock]

**Description**

Returns a reference to Social Security Tax (FICA) primary rate at a predefined location. This member shields other calculations from knowing where the value is actually stored.

**SSTax Rate2**

**Formula**

[OpenInputValueBlock]

[CrossRef("SSTax Rate2 Input")]

[CloseInputValueBlock]

**Description**

Returns a reference to Social Security Tax (FICA) secondary rate at a predefined location. This member shields other calculations from knowing where the value is actually stored.

**Status**

**Formula**

[OpenInputValueBlock]

IF (@ISDESC("New Employees"))

IF ("BegBalance"->"Start Month" != #MISSING)

IF ([FiscalOffset] == 0)
IF ("Cal TP-Index">= "BegBalance"-"Start Month")"Status" = [Status.Active];
ENDIF ELSEIF ("BegBalance"-"Start Month" - [FiscalOffset]) > 0)
IF ("Fiscal TP-Index">= ("BegBalance"-"Start Month" - [FiscalOffset]))"Status" = [Status.Active];
ENDIF
ELSEIF ("Fiscal TP-Index">= ("BegBalance"-"Start Month" + [NumberOfPeriodsInYear] - [FiscalOffset]))"Status" = [Status.Active];
ENDIF
ENDIF
ELSE
IF ("Action" == [Status.TransferIn])"Status" = [Status.Active];
ELSE"Status" = "Action";
ENDIF
IF("Status" == #MISSING)"Status" = @PRIORS(SKIPMISSING, "Status");
ENDIF
ENDIF
[CloseInputValueBlock]

Description
Calculates the employee’s status based on last action applied. Status is used to determine whether an employee is paid.

SUI

Formula
[OpenInputValueBlock]
IF (NOT @ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL"))
IF ("Tax Region" == [TaxRegion.USA])
IF ("CYTD Total Salary"<= "SUI Cap")"SUI Rate" * "Total Salary";
ELSEIF("CYTD Total Salary (Prior)"<= "SUI Cap")"SUI Rate" * ("SUI Cap" - "CYTD Total Salary (Prior)");
ELSE
#MISSING;
ENDIF
ENDIF
[CloseInputValueBlock]
SUI Cap

Formula

```plaintext
[OpenInputValueBlock]
[CrossRef("SUI Cap Input")]
[CloseInputValueBlock]
```

Description

Returns a reference to State Unemployment Insurance (SUI) cap at a predefined location. This member shields other calculations from knowing where the value is actually stored.

SUI Rate

Formula

```plaintext
[OpenInputValueBlock]
[CrossRef("SUI Rate Input")]
[CloseInputValueBlock]
```

Description

Returns a reference to State Unemployment Insurance (SUI) rate at a predefined location. This member shields other calculations from knowing where the value is actually stored.

Target Rate

Formula

```plaintext
IF (@ISLEV("Employee", 0))
IF ("Salary Basis" == [SalaryBasis.Annual])
IF("Grade" == 1)"Target Rate" = 80000;
ELSEIF("Grade" == 2)"Target Rate" = 90000;
ELSEIF("Grade" == 3)"Target Rate" = 100000;
ELSEIF("Grade" == 4)"Target Rate" = 110000;
ELSEIF("Grade" == 5)"Target Rate" = 120000;
```

Description

Calculates simple State Unemployment Insurance (SUI) based on a tax region, rate and cap.
ELSEIF("Grade" == 6) "Target Rate" = 130000;
ELSEIF("Grade" == 7) "Target Rate" = 140000;
ELSEIF("Grade" == 8) "Target Rate" = 150000;
ELSEIF("Grade" == 9) "Target Rate" = 160000;
ELSEIF("Grade" == 10) "Target Rate" = 170000;
ELSEIF("Grade" == 11) "Target Rate" = 180000;
ELSEIF("Grade" == 12) "Target Rate" = 190000;
ELSEIF("Grade" == 13) "Target Rate" = 200000;
ELSE "Target Rate" = #MISSING;
ENDIF
ELSEIF ("Salary Basis" == [SalaryBasis.Hourly]) "Target Rate";
ENDIF
ENDIF

[CloseInputValueBlock]

Description
Calculates the predefined Target Rate based on employee grade.

Temporary FTE

Formula

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")), "ACTUAL") OR @ISMBR ("Department General")))

ELSE

#MISSING;
ENDIF
ENDIF

[CloseInputValueBlock]

Description
Calculates Full Time Equivalent (FTE) for Temporary employee types.
**Temporary Headcount**

**Formula**

[OpenInputValueBlock]

IF (NOT (@ISUDA([Dimension("DIM_NAME_SCENARIO")], "ACTUAL") OR @ISMBR("Department General")))


1;
ELSE
MISSING;
ENDIF
ENDIF

[CloseInputValueBlock]

**Description**

Calculates Headcount for Temporary employee types.

**Turnover Adjustment**

**Formula**

[OpenInputValueBlock]

IF (@ISMBR("Department General"))"Turnover Adjustment" = ("Total Employees"->"Total Compensation" + "Turnover Adjustment") * "Turnover %";
ELSE
MISSING;
ENDIF

[CloseInputValueBlock]

**Description**

Calculates a turnover adjustment.

**Turnover Headcount Adjustment**

**Formula**

[OpenInputValueBlock]
IF (@ISMBR("Department General"))"Turnover Headcount Adjustment" =
@ROUND(("Total Employees"->"Total Headcount" + "Turnover Headcount
Adjustment") * "Turnover %", 0);
ELSE
#MISSING;
ENDIF

[CloseInputValueBlock]

Description
Calculates a turnover headcount adjustment for employees.

## Predefined Smart Lists
Planners use Smart Lists in data forms to work with employee data.

<table>
<thead>
<tr>
<th>Table 14</th>
<th>Predefined Smart Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Smart List</strong></td>
<td><strong>Associated Account Member</strong></td>
</tr>
<tr>
<td>EmployeeType</td>
<td>Employee Type</td>
</tr>
<tr>
<td>FT_PT</td>
<td>FT/PT</td>
</tr>
<tr>
<td>HealthPlan</td>
<td>Health Plan</td>
</tr>
<tr>
<td>Month</td>
<td>Start Month</td>
</tr>
<tr>
<td>PayType</td>
<td>Pay Type</td>
</tr>
<tr>
<td>Performance</td>
<td>Performance</td>
</tr>
<tr>
<td>Position</td>
<td>Position</td>
</tr>
<tr>
<td>SalaryBasis</td>
<td>Salary Basis</td>
</tr>
<tr>
<td>Status</td>
<td>Status</td>
</tr>
<tr>
<td>TaxRegion</td>
<td>Tax Region</td>
</tr>
</tbody>
</table>

## Predefined Smart List Entries
Review the predefined Smart Lists to determine if you need to change any of them or add new ones. Smart Lists are used in predefined member formulas and drive the calculations of Oracle's
Hyperion® Workforce Planning model; before changing or adding Smart Lists, review how Smart List entries affect member formulas.

Hyperion recommends adding new entries to the predefined Smart Lists instead of replacing them. If you change Smart Lists when there is data already in the system, you must update the data to the new values. Consider potential future changes when designing your Smart Lists.

You may want to customize these predefined Smart Lists entries:

- **Tax Region**—your company’s tax regions, depending on the level of granularity of tax calculations in your application. To support new tax types, add a new Tax Region and add the appropriate member with the business logic under Taxes.
- **Performance**—the employee performance categories for your company, such as Meets Expectations.
- **Position**—position titles such as Software Engineer and Director (not used in predefined calculations, so you can easily change the Performance Smart List entries).
- **Status**—employee status, such as Active, Departed, and Maternity.

<table>
<thead>
<tr>
<th>Table 15</th>
<th>Predefined Smart List Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Smart List</strong></td>
<td><strong>Entries</strong></td>
</tr>
</tbody>
</table>
| Employee Type | ○ Regular  
|  | ○ Contractor  
|  | ○ Temporary  
| FT_PT | ○ PT  
|  | ○ FT  
| HealthPlan | ○ IndividualPlan  
|  | ○ IndividualPlus1  
|  | ○ FamilyPlan  
| Month | ○ Jan  
|  | ○ Feb  
|  | ○ Mar  
|  | ○ Apr  
|  | ○ May  
|  | ○ Jun  
|  | ○ Jul  
|  | ○ Aug  
|  | ○ Sep  
|  | ○ Oct  
|  | ○ Nov  
|  | ○ Dec  
| PayType | ○ Exempt  
<p>|  | ○ NonExempt |</p>
<table>
<thead>
<tr>
<th>Smart List</th>
<th>Entries</th>
</tr>
</thead>
</table>
| Performance | ✦ FailsToMeetExpectation  
            | ✦ NeedsImprovement  
            | ✦ MeetsExpectations  
            | ✦ ExceedsExpectations  
            | ✦ FarExceedsExpectation |
| Position | ✦ Assistant  
          | ✦ Manager  
          | ✦ Supervisor  
          | ✦ Director  
          | ✦ VP |
| Salary Basis | ✦ Annual  
              | ✦ Hourly |
| Status | ✦ Active  
        | ✦ Departed  
        | ✦ Disability  
        | ✦ LeaveOfAbsence  
        | ✦ Maternity  
        | ✦ OnSabbatical  
        | ✦ TransferOut  
        | ✦ TransferIn  
        | ✦ Resigned |
| Tax Region | ✦ NoRegion  
             | ✦ USA |

**Predefined Menus**

Planners use menus to work with employee data in data forms. The information listed in the Label Value column is displayed when planners click a row member.

**Table 16 Predefined Menu: WFPMenu**

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Employee Status</td>
<td>Runs the predefined business rule to change the status of selected employees</td>
</tr>
<tr>
<td>Transfer In</td>
<td>Runs the predefined business rule to transfer selected employees into a department</td>
</tr>
<tr>
<td>Transfer Out</td>
<td>Runs the predefined business rule to transfer selected employees out of a department</td>
</tr>
<tr>
<td>Plan Departure</td>
<td>Runs the predefined business rule to plan for employee departures</td>
</tr>
<tr>
<td>Edit Related Expenses</td>
<td>Lets planners edit related expenses</td>
</tr>
</tbody>
</table>
### Table 17  Predefined Menu: WFPMenuNewHires

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add TBH</td>
<td>Menu header for Add Salary TBH and Add Hourly TBH menu items</td>
</tr>
<tr>
<td>Add Salary TBH</td>
<td>Adds annual salaried TBH to the selected department</td>
</tr>
<tr>
<td>Add Hourly TBH</td>
<td>Adds hourly paid TBH to the selected department</td>
</tr>
<tr>
<td>Remove TBH</td>
<td>Runs the predefined business rule to remove new hires</td>
</tr>
<tr>
<td>Reconcile TBH</td>
<td>Runs the predefined business rule to reconcile TBHs with actual new hires</td>
</tr>
</tbody>
</table>

### Table 18  Predefined Menu: WFPMenuRecXfer

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Transfer Report</td>
<td>Runs the predefined business rule to calculate the transferred employees report</td>
</tr>
<tr>
<td>Reconcile Employee Transfer</td>
<td>Goes to the Reconcile Employee Transfer Data form</td>
</tr>
</tbody>
</table>

### Table 19  Predefined Menu: WFPMenuToMEE

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Manage Existing Employees</td>
<td>Goes to the Manage Existing Employee Status data form</td>
</tr>
<tr>
<td>To Manage Existing Employees</td>
<td>Goes to the Manage Existing Employee data form</td>
</tr>
</tbody>
</table>

### Table 20  Predefined Menu: WFPMenuToNewHires

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To New Hires</td>
<td>Goes to the New Hires data form</td>
</tr>
</tbody>
</table>

### Table 21  Predefined Menu: WFPMenuToRecXfer

<table>
<thead>
<tr>
<th>Label Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go To Reconcile Transfer Report</td>
<td>Goes to the Reconcile Transfer Report data form</td>
</tr>
</tbody>
</table>

This table lists the predefined data forms that have associated menus.

### Table 22  Predefined Menus in Data Forms

<table>
<thead>
<tr>
<th>Data Form</th>
<th>Associated Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Manage Existing Employee Status</td>
<td>WFPMenu</td>
</tr>
<tr>
<td>2. Manage Existing Employees</td>
<td>WFPMenu</td>
</tr>
<tr>
<td>3. Existing Employee Related Expenses</td>
<td>WFPMenuToMEE</td>
</tr>
<tr>
<td>4. New Hires</td>
<td>WFPMenuNewHires</td>
</tr>
</tbody>
</table>
Predefined Business Rules

Planners use predefined business rules to calculate employee data.

Add TBH Hourly

Description

Adds hourly-paid employees to the selected department.

Formula

```plaintext
SET CREATENONMISSINGBLK ON;
VAR numTBH = [NumTBH];
VAR validTBH = 0;
FIX(@CHILDREN("New Employees"), [Scenario], [Version], [Department], [Year], "Local", "HSP_InputValue")
  FIX(@LEVMBRS("Period", 0))
  "Start Month"
  }
  IF (@ISMBR("BegBalance"))
    validTBH = 0;
    IF (numTBH > 0 AND @MAXS(SKIPMISSING, @CHILDREN("Employee Properties")) == #MISSING )
      "Start Month" = [StartMonth]->"Cal TP-Index";
      "Health Plan" = [HealthPlan];
      numTBH = numTBH - 1;
      validTBH = 1;
    ENDIF
    ENDIF
  IF (validTBH == 1)
    "Grade" = [Grade];
    IF ([MarketAdjustment] != 0)
      "Market Adjustment" = [MarketAdjustment];
    ENDIF
    "FTE" = [FTE];
    "Position" = [Position];
    "Tax Region" = [TaxRegion];
    "Employee Type" = [EmployeeType];
    "Pay Type" = [PayTypeNonExempt];
    "Salary Basis" = 2;
    "Hours per week" = [HoursPerWeek];
    "Target Rate" = [SalaryRate];
    "Salary Rate" = [SalaryRate] + [MarketAdjustment];
  ENDIF
```

90  Workforce Planning Structure
Add TBH Salary

**Description**

Adds salaried employees to the selected department.

**Formula**

```
SET CREATENONMISSINGBLK ON;
VAR numTBH = [NumTBH];
VAR validTBH = 0;
FIX(@CHILDREN("New Employees"), [Scenario], [Version], [Department], [Year], "Local", "HSP_InputValue")
   FIX(@LEVMBRS("Period", 0))
   "Start Month"
   (  
      IF (@ISMBR("BegBalance"))
         validTBH = 0;
      IF (numTBH > 0 AND @MAXS(SKIPMISSING, @CHILDREN("Employee Properties")) == #MISSING )
         "Start Month" = [StartMonth]->"Cal TP-Index";
         "Health Plan" = [HealthPlan];
         numTBH = numTBH - 1;
         validTBH = 1;
      ENDFIX
   IF (validTBH == 1)
      "Grade" = [Grade];
      IF ([MarketAdjustment] != 0)
         "Market Adjustment" = [MarketAdjustment];
      ENDFIX
      "FTE" = [FTE];
      "Position" = [Position];
      "Tax Region" = [TaxRegion];
      "Employee Type" = [EmployeeType];
      "Salary Basis" = 1;
      "Pay Type" = [PayTypeExempt];
   ENDFIX
)
CALC DIM ("Account");
ENDFIX

ENDFIX

Change Employee Status

**Description**

Predefined Business Rules 91
Changes employee status.

Formula

```
SET CREATENONMISSINGBLK ON;
VAR startIndex;
VAR endIndex;
VAR existingAction = #MISSING;
FIX ([Scenario], [Version], [Employees], [Department], [Month], [Year], "Local", "HSP_InputValue")
"Action"
(
    startIndex = [Month]->"Fiscal TP-Index";
    endIndex = startIndex + [Duration];
    IF (endIndex < startIndex)
        endIndex = startIndex;
    ENDIF
);
ENDFIX
FIX ([Scenario], [Version], [Employees], [Department], [Year], "Local", "HSP_InputValue")
"Action"
(
    IF ("Fiscal TP-Index" == startIndex)
        "Action" = [Action];
    ELSEIF ("Fiscal TP-Index" > startIndex AND "Fiscal TP-Index" < endIndex)
        existingAction = "Action";
        "Action" = #MISSING;
    ELSEIF ("Fiscal TP-Index" == endIndex AND "Action" == #MISSING)
        IF (existingAction == #MISSING)
            "Action" = 1;
        ELSE
            "Action" = existingAction;
       ENDIF
    ELSE
        "Action" = existingAction;
   ENDIF
);
CALC DIM ("Account");
ENDFIX
```

CopyProps

Description
Copies employee properties from BegBalance into all time periods.

Formula

```
FIX ([Scenario], [Version], [Department], [Year], @CHILDREN("New Employees"), @LEVMBRS ("Period", 0), "Local", "HSP_InputValue")
"Start Month" (  
    "Grade" = "BegBalance"->"Grade";
    "FTE" = "BegBalance"->"FTE";
    "Position" = "BegBalance"->"Position";
    "Tax Region" = "BegBalance"->"Tax Region";
    "Employee Type" = "BegBalance"->"Employee Type";
    "Pay Type" = "BegBalance"->"Pay Type";
    "Salary Basis" = "BegBalance"->"Salary Basis";
```
"Salary Rate" = "BegBalance"->"Target Rate" + "BegBalance"->"Market Adjustment";
"Target Rate" = "BegBalance"->"Target Rate";
"Market Adjustment" = "BegBalance"->"Market Adjustment";
IF ("Salary Basis" == 2)
  "Hours per week" = "BegBalance"->"Hours per week";
ENDIF

Plan Departure

Description
Lets users plan for an employee’s departure.

Formula

SET AGGMISSG ON;
FIX ([Scenario], [Version], [Employees], [Department], [Year], @LEVMBRS("Period", 0),
"Local", "HSP_InputValue")
"Action"
(  IF ("Fiscal TP-Index" > [Month]->"Fiscal TP-Index")
    "Action" = #MISSING;
  ELSEIF ("Fiscal TP-Index" == [Month]->"Fiscal TP-Index")
    "Action" = [Action];
  ENDIF
);
CALC DIM ("Account");
ENDFIX

Reconcile TBH

Description
Reconciles TBHs with actual new hires.

Formula

FIX ([Scenario], [Version], [Department])
  DATACOPY [SrcEmployee] TO [DestEmployee];
ENDFIX
FIX ([Scenario], [Version], [Department], [DestEmployee], [Year], @LEVMBRS("Period", 0),
"Local", "HSP_InputValue")
"Start Month"
  IF ("Cal TP-Index" == "BegBalance"->"Start Month")
    "Action" = 1;
    "BegBalance"->"Start Month" = #MISSING;
  ENDIF
"Grade" = "BegBalance"->"Grade";
"FTE" = "BegBalance"->"FTE";
"Position" = "BegBalance"->"Position";
"Tax Region" = "BegBalance"->"Tax Region";
"Employee Type" = "BegBalance"->"Employee Type";
"Pay Type" = "BegBalance"->"Pay Type";
"Salary Basis" = "BegBalance"->"Salary Basis";
"Salary Rate" = "BegBalance"->"Target Rate" + "BegBalance"->"Market Adjustment";
IF ("Salary Basis" == 2)
"Hours per week" = "BegBalance"->"Hours per week";
ENDIF
); CALC DIM ("Account");
ENDFIX
FIX ([Scenario], [Version], [Department])
  CLEARDATA [SrcEmployee];
  CALC DIM ("Account");
ENDFIX

Remove TBH
Description
Removes a TBH.

Formula

FIX ([Scenario], [Version], [Department])
  CLEARDATA [LocalEmployee];
  CALC DIM ("Account");
ENDFIX

Rollup
Description
Rolls up certain dimensions.

Formula

SET AGGMISSG ON;
FIX ([Scenario], [Version], [Department], [Year], "Local", "HSP_InputValue")
  CALC DIM ("Account", "Period", "Employee");
ENDFIX

Transfer
Description
Performs a one-step employee transfer.

Formula

SET CREATENONMISSINGBLK ON;
VAR savedAction = 0;
FIX ([Scenario], [Version], [Employees], [Year])
  DATACOPY [FromDepartment] TO [ToDepartment];
ENDFIX
FIX ([Scenario], [Version], [Employees], [FromDepartment], [Year], @LEVMBRS("Period", 0), "Local", "HSP_InputValue")
"Action"
{
  IF ("Fiscal TP-Index" == [Month]->"Fiscal TP-Index")
    "Action" = 7;
  ELSEIF ("Fiscal TP-Index" > [Month]->"Fiscal TP-Index")
    "Action" = #MISSING;
  ENDIF
}
CALC DIM ("Account");
ENDFIX

FIX ([Scenario], [Version], [Employees], [ToDepartment], [Year], [Month], @LEVMBRS("Period", 0), "Local", "HSP_InputValue")
"Action"(
  IF ("Fiscal TP-Index" == [Month]->"Fiscal TP-Index")
    "Action" = 8;
  ELSEIF ("Fiscal TP-Index" < [Month]->"Fiscal TP-Index")
    "Action" = #MISSING;
  ELSE
    IF (savedAction <> 7 AND "Action" == 8)
      "Action" = #MISSING;
    ELSEIF ("Action" == 7)
      savedAction = 7;
    ENDIF
  ENDIF
)
CALC DIM ("Account");
ENDFIX

Transfer In

Description
Transfers an employee into a department.

Formula

SET CREATENONMISSINGBLK ON;
VAR savedAction = 0;
FIX ([Scenario], [Version], [Employees], [Department], [Year], @LEVMBRS("Period", 0), "Local", "HSP_InputValue")
"Action"(
  IF ("Fiscal TP-Index" == [Month]->"Fiscal TP-Index")
    "Action" = 8;
  ELSEIF ("Fiscal TP-Index" < [Month]->"Fiscal TP-Index")
    "Action" = #MISSING;
  ELSE
    IF (savedAction <> 7 AND "Action" == 8)
      "Action" = #MISSING;
    ELSEIF ("Action" == 7)
      savedAction = 7;
    ENDIF
  ENDIF
)
ENDFIX
FIX ([Scenario], [Version], [Employees], [Department], [Year], @LEVMBRS("Period", 0), "Local", "HSP_InputValue")
"Action"
IF ([FTE] != #MISSING)
    "FTE" = [FTE];
ENDIF;
IF ([Position] != #MISSING)
    "Position" = [Position];
ENDIF;
IF ([Grade] != #MISSING)
    "Grade" = [Grade];
ENDIF;
IF ([TaxRegion] != #MISSING)
    "Tax Region" = [TaxRegion];
ENDIF;
IF ([EmployeeType] != #MISSING)
    "Employee Type" = [EmployeeType];
ENDIF;
IF ([HealthPlan] != #MISSING)
    "Health Plan" = [HealthPlan];
ENDIF;
IF ([MeritMonth] != #MISSING)
    "Merit Month" = [MeritMonth];
ENDIF;
    IF ([SalaryBasis] != #MISSING)
        "Salary Basis" = [SalaryBasis];
        IF ("Salary Basis" == 2)
            "Hours per week" = [HoursPerWeek];
        ENDIF
    ENDIF
ENDIF;
IF ([SalaryRate] != #MISSING)
    "Salary Rate" = [SalaryRate];
ENDIF;
IF ([PayType] != #MISSING)
    "Pay Type" = [PayType];
ENDIF;
);
CALC DIM ("Account");
ENDFIX

Transfer Out

Description

Transfers an employee out of a department.

Formula

FIX ([Scenario], [Version], [Employees], [Department], [Year], @LEVMBRS("Period", 0), "Local", "HSP_InputValue")
"Action"
(
    IF ("Fiscal TP-Index" == [Month]->"Fiscal TP-Index")
        "Action" = 7;
    ELSEIF ("Fiscal TP-Index" > [Month]->"Fiscal TP-Index")
        "Action" = #MISSING;
    ENDIF
);
CALC DIM ("Account");
ENDFIX

### Unreconciled Transfers

**Description**

Compares transfers out against transfers in to produce a reconciliation report.

**Formula**

```plaintext
SET CREATENONMISSINGBLK ON;

FIX ([Scenario], [Version], [Year], @REMOVE(@LEVMBRS("Employee",0), @LIST(@CHILDREN("New Employees"), "Department General", "No Employee")), @LEVMBRS("Period", 0), "No Entity", "Local", "HSP_InputValue")

   "TransferOutCount" = #MISSING;

   "TransferInCount" = #MISSING;

ENDFIX

FIX ([Scenario], [Version], [Year], @REMOVE(@LEVMBRS("Employee",0), @LIST(@CHILDREN("New Employees"), "Department General", "No Employee")), @LEVMBRS("Period", 0), "Local", "HSP_InputValue")

FIX(@REMOVE(@LEVMBRS("Entity",0), @LIST("No Entity")))

   "Action" (  
      IF  ("Action" == 7)
         "No Entity"->"TransferOutCount" = "No Entity"->"TransferOutCount" + 1;
      ELSEIF ("Action" == 8)
         "No Entity"->"TransferInCount" = "No Entity"->"TransferInCount" + 1;
      ENDIF
   )

ENDFIX

ENDFIX

FIX ([Scenario], [Version], [Year], @REMOVE(@LEVMBRS("Employee",0), @LIST(@CHILDREN("New Employees"), "Department General", "No Employee")), @LEVMBRS("Period", 0), "No Entity", "Local", "HSP_InputValue")

"Unreconciled Transfers"(

   IF  ("TransferInCount" == "TransferOutCount")
      IF  ("TransferInCount" == 1 OR "TransferInCount" == #MISSING)
         #MISSING;
      ELSE
         0;
   )
```

Predefined Business Rules 97
ENDIF
ELSEIF ("TransferInCount" > "TransferOutCount")
    1;
ELSE
    -1;
ENDIF
);
ENDFIX
Index

A
access permissions, setting up, 27
accessing
  Architect, 17
  Planning, 17
  Workforce Planning, 17
  Workspace, 17
accounts
  adding, 22
  fixed and varying, 54
  fixed and varying, loading data into, 16
  predefined, 53
  salary, 23
ACTUAL UDA, 15, 26
Add TBH Hourly business rule, 90
  defined, 90
  using, 25
Add TBH Salary business rule
  defined, 91
  using, 25
annual salary calculation, 23
Application Library, working with, 17
Application Views
  deploying in Architect, 13
applications
  adding in Architect, 12
  creating in Architect, 11
  creating, overview, 9
  deploying in Architect, 13
  initializing Classic applications, 14
  initializing in Architect, 13
  localizing, 25
  selecting, 17
Architect
  accessing, 17
  adding a performance rating, example, 30
  adding a salary adjustment type, example, 39
  changing a pay rate, example, 33
  changing bonus frequency, example, 36
  creating an application, 10
  documentation for, 17
  loading information from enterprise systems, 15
B
BegBalance member, 53
benefits of Workforce Planning, 7
bonuses, changing the frequency, 35
business model, 8
business rules
  predefined, 90
  working with, 25
C
calc scripts, predefined, 90
calculations, performing, 25
calendar, 9
Change Employee Status business rule, 22, 25
Classic applications
  adding a performance rating, example, 32
  adding a salary adjustment type, example, 45
  changing a pay rate, example, 35
  changing bonus frequency, example, 38
  creating a Workforce Planning application, 14
  initializing, 14
  loading information into, 15
  creating
    applications using Architect, 10
    Classic applications, 14
    import profiles, 10
CrossDim operator, 54
customizing
  before you start, 29
  business rules and calc scripts, 90
  examples, 29
  menus, 88
Smart List entries, 86
Smart Lists, 86
understanding effects of, 51

D
data
  avoid losing, 15
  loading using Application Link Adapter for Hyperion Planning, 16
  loading, overview, 15
  spreading, 22
data forms
  managing, 17
  predefined, 51
  viewing and managing, 17
  working with, 24
Data Forms folder, 17
department expenses, planning, 22
Department General member, 22
departments, maximum number of TBHs, 21
deploying Architect applications, 13
Dimension Library, working with, 17
dimensions
  importing using Architect, 11
  mapping, 10

E
editing data forms, 17
elements of Workforce Planning, 51
Employee Type Smart List, 87
employees
  changing status, 22
  converting TBHs to hires, 21
  terminating, 22
  transferring, 19
  working with, 19
equity adjustment, example of adding, 39
examples of customizing Workforce Planning, 29

F
fixed accounts
  described, 54
  loading data into, 16
formula expressions
  time balance, 23
  working with, 25
FT_PT Smart List, 87

H
HealthPlan Smart List, 87
Hourly
  effect on Annual Salary calculation, 23
  Smart List entry, 88
HspCustomMsgs template file, 25

I
import profiles, creating, 10
importing, 15
importing dimensions into Architect, 11
initializing Workforce Planning applications, 9
  with Architect, 13
  with Classic application administration, 14
introduction, 7

L
labels
  about localizing, 30, 32
  adding or modifying, 25
loading data
  overview, 15
  using Application Link Adapter for Hyperion Planning, 15
  using DIM Adapter for Planning, 15
localizing applications, 25
logging on
  to Workforce Planning, 17
  to Workspace, 17

M
mapping dimensions, 10
maternity status, changing pay rate, 33
member formulas
  predefined, 59
  working with, 25
menus
  predefined, 88
  working with, 24
metadata
  loading using Application Link Adapter for Hyperion Planning, 16
  loading using DIM Adapter for Planning, 16
loading, overview, 15

Navigating the menu, working with, 17

Overview, 7

Password, 17
PayType Smart List, 87
Performance rating, example of adding a new, 29
Performance Smart List, 87
Position Smart List, 87
prerequisites, 8

Reconcile Employee Transfer data form
defined, 53
using, 21
Reconcile Transfer Report data form
defined, 53
using, 20

Salary
account members described, 23
adding an adjustment type, 39
calculating annual salary, 23
described, 54
example of changing pay rate, 33
SalaryBasis Smart List, 88
scenario, Actual, 26
security. See access permissions
setting up Workforce Planning, 7
Smart Lists
entries, 86
described, 29
predefined, 86
working with, 24
spreading data
described, 54
and custom time periods, 9
using system members, 22
Status Smart List, 87
status, changing for employees, 22
structure of Workforce Planning, 51

system members, spreading data with, 22

Task lists, using for process management, 27
Tax Region Smart List, 87
TBHs
adding, 21
converting to hires, 21
increasing the maximum number, 21
terminating employees, 22
time balance spread formula expressions, 23
time periods, custom, 9
TimeBalanceSpread, 23, 25
Transfer business rule
defined, 94
using, 19
Transfer In business rule
defined, 95
using, 19
Transfer Out business rule
defined, 96
using, 19
transferring employees
described, 19
reconciling transfers, 20

UDAs
ACTUAL UDA, 15, 26
working with, 26
Unreconciled Transfers business rule
defined, 97
using, 20
URL for Workspace, 17
user name, 17

Valid for Workforce check box, 12
varying accounts
described, 54
loading data into, 16

working with
Application Library, 17
Dimension Library, 17
Workspace, logging on, 17

Y
Year Total, 53
years, supporting multiple, 9