Oracle® Fusion Cloud EPM

Design Best Practices

F96806-08

Oracle Fusion Cloud EPM Design Best Practices,

F96806-08

Copyright © 2025, Oracle and/or its affiliates.

Primary Author: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Documentation Accessibility

Documentation Feedback

1 Introduction to Design Best Practices

2 Creating and Running an EPM Center of Excellence

3 Designing Your Application

Getting Started with Your Design 1
- Gathering Design Information 2
- Planning Your Application 5
- Applying Your Design 6
- Building, Testing, and Rolling Out the Application 7

Application Design 9
Cube Types Design 12
Hybrid BSO Design 12
ASO Design 16
Migrating from Legacy SKUs Design 17
Dimension Design 17
Best Practices for Designing Business Rules 28
Data Maps Design 33
Smart Push Design 34
Form Design 34
Dashboards Design 37
Approvals Design 39
Tasks Design 40
Navigation Flow Design 40
Access Permissions Design 41
Security Design 45

Copyright © 2025, Oracle and/or its affiliates.

Reports Design 45
Integration Design 46

4 Addressing Common Requirements

5 Use Case Examples

6 Design Checklist

Copyright © 2025, Oracle and/or its affiliates.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Copyright © 2025, Oracle and/or its affiliates.

Documentation Feedback

To provide feedback on this documentation, click the feedback button at the bottom of the page in any Oracle Help Center topic. You can also send email to epmdoc_ww@oracle.com.

Copyright © 2025, Oracle and/or its affiliates.

1

Introduction to Design Best Practices

This Design Best Practices guide helps you plan and design high-quality, scalable Planning applications by providing clear, proven guidelines that enhance efficiency, usability, and long-term success.

Whether you are starting a new implementation or enhancing an existing solution, this information will support better outcomes across the project lifecycle.

By following the design best practices in this guide, you can improve delivery efficiency and ensure that applications align with strategic objectives and user expectations.

Key benefits of the guide include:

- **Structured design guidelines** – Apply a consistent, proven approach to application design that enhances usability, performance, and scalability.
- **Solutions to common requirements** – Gain insights into frequently encountered business and technical needs, and learn how to meet them using best-practice recommendations.
- **Real-world use case examples** – Learn from practical examples that illustrate how to successfully apply design principles in real scenarios.
- **Design checklist** – Use the included checklist to validate your designs and enhance quality and consistency across your implementations.

Use this guide to design with greater confidence, ensure alignment with leading practices, and deliver solutions that drive greater business impact and user satisfaction.

Copyright © 2025, Oracle and/or its affiliates.

2

Creating and Running an EPM Center of Excellence

A best practice for EPM is to create a CoE (Center of Excellence).

An **EPM CoE** is a unified effort to ensure adoption and best practices. It drives transformation in business processes related to performance management and the use of technology-enabled solutions.

Cloud adoption can empower your organization to improve business agility and promote innovative solutions. An EPM CoE oversees your cloud initiative, and it can help protect and maintain your investment and promote effective use.

The EPM CoE team:

- Ensures cloud adoption, helping your organization get the most out of your Oracle Fusion Cloud EPM investment
- Serves as a steering committee for best practices
- Leads EPM-related change management initiatives and drives transformation

All customers can benefit from an EPM CoE, including customers who have already implemented EPM.

How Do I Get Started?

Click to get best practices, guidance, and strategies for your own EPM CoE: Introduction to EPM Center of Excellence.

Learn More

- Watch the Cloud Customer Connect webinar: Creating and Running a Center of Excellence (CoE) for Cloud EPM
- Watch the videos: Overview: EPM Center of Excellence and Creating a Center of Excellence.
- See the business benefits and value proposition of an EPM CoE in *Creating and Running an EPM Center of Excellence*.

Copyright © 2025, Oracle and/or its affiliates.

Copyright © 2025, Oracle and/or its affiliates.

3
Designing Your Application

Review the topics in this section to design applications that meet your business needs.

Start by understanding how to design your application.

Then check out these topics:

- Application Design
- Cube Types Design
- Hybrid BSO Design
- ASO Design
- Migrating from Legacy SKUs Design
- Dimension Design
- Best Practices for Designing Business Rules
- Data Maps Design
- Smart Push Design
- Form Design
- Dashboards Design
- Approvals Design
- Tasks Design
- Navigation Flow Design
- Access Permissions Design
- Security Design
- Reports Design
- Integration Design

Getting Started with Your Design

Follow these basic steps to design an application based on best practices that meets your business needs.

1. Gathering design information
2. Planning your application
3. Applying your design
4. Building, testing, and rolling out the application

Copyright © 2025, Oracle and/or its affiliates.

Training for Application Design

Your Goal	Watch This Video or Training
Learn about application deployment, dimensions, and forms.	Application Design for EPM Planning Part I: App Deployment, Dimensions, and Forms
Learn about designing with Calculation Manager rules and hybrid BSO.	Application Design for EPM Planning Part II: Calculation Manager Rules and Hybrid BSO
Learn about designing with ASO, data maps, and Smart Push.	Application Design for EPM Planning Part III: ASO, Data Maps, and Smart Push
Learn abut designing with different application types and migration from legacy SKUs.	Application Design for EPM Planning Part IV: Application Types and Migration from Legacy SKUs
Learn how to gather design documentation.	Gathering Design Documentation in Oracle Planning and Budgeting Cloud
Learn how to design an application.	Designing Planning Applications in Oracle Planning and Budgeting Cloud
Learn how to set up dimensions.	Setting Up Dimensions in Oracle Planning and Budgeting Cloud
Learn how to design forms.	Designing Forms in Cloud EPM Planning
Learn how to build effective and efficient calculations.	Building Effective and Efficient Calculations in Oracle Planning and Budgeting Cloud
Learn how to monitor system activities.	Monitoring System Activities in Oracle Planning and Budgeting Cloud

Gathering Design Information

Before you create your application, gather design information for your current processes. This is a best practice to ensure that the application meets your business requirements.

Follow these steps to gather your design information:

- Analyze your current processes
- Gather any existing spreadsheets
- Gather your current reporting requirements
- Review your current financial statements
- Gather planning and forecast requirements
- Understand your data sources
- Identify process improvements

Watch this video to learn more: Gathering Design Documentation in Oracle Planning and Budgeting Cloud

Copyright © 2025, Oracle and/or its affiliates.

Analyze Your Current Processes

Look at what is working in your current processes, and plan how to build that in to the application. Consider taking this opportunity to improve current processes.

- Review your company's financial statements. Determine the key revenue and expense areas that make up 80% of profit and loss.
- Focus on the major drivers of your business. For this subset of accounts, understand the business drivers and required dimensions. For example, employee compensation typically accounts for 50-60% or more of expenses.
- Decide if detailed modeling is required, or if another methodology exists. For each major account in the profit and loss statement, understand how to derive appropriate values for the account. Business logic could include simple calculations like Units x Rates or other methods.
- For the remaining 20% of the accounts, plan or forecast using some simple logic such as percentage growth or simple entry.

Gather Any Existing Spreadsheets

Based on your current process, gather your input sources and understand their purpose. Review the sources to determine the business logic, relationships, and other calculations.

- Analyze the spreadsheets to understand your current processes. Understand the sources of information and the business logic that they support. Find out if your business process logic applies to all organizations.
- Find out if your business process logic applies to other business units and divisions. If the process differs, decide if you want to make processes consistent.

Gather Your Current Reporting Requirements

- Review reports and consider what type of information is included.
- Understand the business logic, relationships, and calculations for your current processes.
- Find out if your business process logic applies to other business units and divisions. If the process differs, decide if you want to make processes consistent.

Review your Current Financial Statements

- Identify the dimensions in reports, for example, if reports are done by cost center, division, account, or customer. Decide if you need to add dimensions, such as product.
- Review financial statements and examine the intersections. Learn if reports are done by account and entity or by other slices.
- Understand the business drivers for accounts, including the dimensions involved. Review profit and loss line by line to determine if modeling is involved or if the information is simply added by users.
- Note the layout of reports, including which elements are on the rows and columns.
- Determine the number of unique report formats such as by Cost Center or by Division. Specify a report format for each type of report you need.

Gather Planning and Forecast Requirements

Collect the requirements for producing an annual plan and forecast. Conduct an analysis and understand if all processes are the same.

Copyright © 2025, Oracle and/or its affiliates.

- **Determine currency requirements**. Identify the currencies in which you plan and report, and whether you plan or forecast in multiple currencies. Verify foreign currency requirements. Confirm if you perform any currency sensitivity analysis. Are multiple currencies planned for the same intersection of data as compared to simply having a different Local currency per Entity, Cost Center, or Department?
- **Determine the planning horizon**. Know how many years in the future you plan, such as one, two, three, or five years into the future. Plan how much historical data to bring into the application. It's typical to include one or two years of historical data.
- **Identify the planning and forecasting process**. Know whether you prepare annual budgets and forecasts. Understand how often you revisit your forecast, such as monthly or quarterly.
- **Determine the planning processes you want to support**, such as Plan or Forecast. Establish if planning and forecasting processes are similar. Know whether you want to prepare a rolling forecast.
- **Decide if you'll plan toward a target**. If you set high level targets for plan or forecast, decide on the level of accounts to target. Account parent members are usually dynamically calculated and can't have data entered into them for a Target Version. Typically, you can pick a Level 0 member within the parent to enter the Target data.
- **Determine if multiple iterations are required**. Establish if you need to store multiple submissions of Plan or Forecast. Determine how many forecasts you maintain. Decide if you want to do comparisons, such as comparing forecasts. Typically, users enter data into the same Version iteration, and the admin copies the snapshot of data to the milestone or snapshot for comparisons.
- Additional questions to help you plan the process:
  - What are the expense drivers for your business, such as compensation, material, travel, or capital and equipment expenses?
  - What are the revenue drivers for your business, such as products, projects, or services?
  - If Product is one of the main revenue drivers, how many products do you have?
  - How many business units or entities do you have? What main metrics and KPIs are tracked? How many users are involved in financial planning? Who are the main users?
  - Determine the approval process that should be incorporated into the application. Is the approval process the same for Plan and Forecast?

Understand Your Data Sources

Know the file format and the contents of each data source. This helps you plan requirements for dimension and data integrations.

- Determine the source of dimensions such as Account, Entity, and custom dimensions.
- Determine the source and frequency of Actual results.

Identify Process Improvements

Identify any process improvements to include in your application.

Plan the strengths to incorporate going forward, and identify any weakness and areas to improve in the future.

Copyright © 2025, Oracle and/or its affiliates.

Planning Your Application

An important best practice is to establish the goals, key objectives, and scope of your application.

After you know the requirements, you can move forward with designing your application.

The information you gather can be included in your application as follows:

- Dimensions
- Forms
- Reports
- Calculations
- Users

Additional planning steps:

- **Design dimensions**.
  - Identify the dimensions required to support your process. Your application comes with these dimensions: Account, Entity, Version, Scenario, Years, Period, and Currency, where applicable.
  - Identify the members to be included in the application and the source of those dimensions. Understand the size of each dimension.
  - It's a best practice to document calculations. For each piece of the calculation, document how the result is determined and the source of the number. Confirm if the source is data entry or data fed from another system.
- **Design calculations**. Review your company's profit and loss statement line by line, and confirm how to plan or forecast the account. For accounts that require calculations, you can leverage Calculation Manager to build the logic.

  Understand calculations, and ask any necessary questions. For base accounts, confirm how each account is planned. Sample questions:
  - Does this calculation apply for all entities?
  - Are there any other dimensional aspects to this calculation?
  - Is the calculation by product or customer?
  - Is the calculation the same for Plan and Forecast?
- **Determine revenue calculations**. It's a best practice to establish the requirements for revenue and its drivers.
  - Know the business drivers you use, and understand if revenue is driven by product or services. Decide if you want to derive revenue in your model. Identify any other information to capture.
  - Establish the revenue drivers. Determine if other dimensions are required.

    If additional dimensions are required, determine if Revenue will be calculated in its own cube, for example, if more dimensions are needed for revenue than for expenses. Adding multiple dimensions to a common cube can impact performance and usability.
  - Decide on the logic for revenue planning. Identify the calculations you want to build to support your process, for example, Units x Price. Plan the logic behind your revenue calculations.

Copyright © 2025, Oracle and/or its affiliates.

  - Determine the forms and the layout to collect information and input from end users.
- **Determine expense requirements**.
  - Review your expense accounts to identify the key areas of your business.
  - Focus on the areas that make up the majority of your expense to identify the major drivers.
  - You can plan the remaining minor expenses in a simple and straightforward way, such as by using trends or by simply entering a value.
- **Determine expense drivers**.
  - Focus on this group of accounts, and determine the business drivers of the accounts and the required dimensions.
  - Review your expense accounts to identify the key areas of your business. Focus on the areas that make up the majority of your expense to identify the major drivers. You can plan the remaining minor expenses in a simple and straightforward way, such as by using trends or by simply entering a value.
  - Review employee compensation. Confirm how employee compensation will be planned, such as by employee, job, or grade.
  - Know which accounts are included in planning, and understand the sources of data to support compensation. Identify the type of reporting you do on compensation.
  - Assess other expense account requirements. Plan the calculations you need to build to support your process. Know the logic behind your expense calculations.
  - Document calculations. For each piece of the calculation, document how the result is determined and the source of the number. Confirm if the source is data entry or data fed from another system.
  - Determine the forms and the layout to collect information and inputs from the end users.

  Determine the approval process.
  - Determine the dimensions that drive the approval process for Plan and Forecast. The base dimensions of the approval process are Entity, Scenario, and Version. Decide if you want to include another dimension, such as Product, Project, or Market.
  - Document the approval process the application should support.

Applying Your Design

After reviewing the design guidelines and best practices in this section, you're on your way to designing a successful application that meets your business needs. You can now quickly and easily set up your application.

You can use the information that you gathered to set up your application. As an example, here is how some of the information you gathered corresponds to your application.

Table 3-1 Applying Your Design to the Application

Research on Requirements	Application Setting
The number of historical years to include	The Start Year for the application
Your planning horizon	The number of years chosen for the application

Copyright © 2025, Oracle and/or its affiliates.

Table 3-1 (Cont.) Applying Your Design to the Application

Research on Requirements	Application Setting
The planning processes to support	The scenarios included in the application. You can add scenarios while creating the application, or you can add them later.
Whether you need to store multiple submissions of Plan or Forecast	The number of submissions correlates to versions in the application. You can add versions in the application wizard, or you can add them later.
Whether you plan in multiple currencies	If you plan in multiple currencies, answer Yes in the application wizard. You'll add currencies later. Select your reporting currency in the wizard.
Time periods	If you require weekly distribution such as 445, 454, or 544, select the base time period as 12 Months, and then select an appropriate Weekly Distribution option. Even monthly distribution is the default. When you create an application, you can also set up the planning calendar based on 13 periods, as described in About 13-Period Calendars.

Building, Testing, and Rolling Out the Application

Use these best practices as you build and roll out your application.

Follow these key steps:

1. Build your application
2. Test
3. Roll out
4. Enable the application for users

Build Your Application

Start by building the foundation—your company's accounts and organizational structure. Next, add scenarios to support your internal processes, such as Plan, Actual, and Forecast. Add the variance members you report, such as Actual vs. Plan.

Create forms that will be used to collect data from your users and to perform reviews, analysis, and reporting. To support your business logic, you can leverage Calculation Manager to build your calculations. You can also create reports and apply access permissions before rolling out your application to users.

For help with this task, first review the topics in this guide. After that, you can follow the procedures in Creating a Planning Application in *Administering Planning.*

Test

Testing is a critical step in application development. All of the calculations, access permissions, and reports must be tested to ensure that they work appropriately.

Plan Unit Testing

Unit testing is the first step of formalized testing, and is the main building block of the test environment. Unit testing involves testing each functional area of the application as a separate unit to ensure that it performs as expected.

Copyright © 2025, Oracle and/or its affiliates.

For example, a test could confirm that a data load executes to completion without errors. Other tests could confirm that forms and reports are accessible, calculations complete, and so on. Tests should also confirm that calculation logic and math works correctly.

The person that builds or configures the application usually conducts unit testing.

Plan System Testing

System testing validates that the system operates without error and provides the required functionality. The main emphasis is to test the way the application has been configured and to look at how the team constructed the business processes and reports.

System testing focuses on testing the entire system, including unique parameter configuration, all functions that will be used, and any enhancements. System testing also looks beyond the software, and validates the effectiveness of manual procedures, forms and controls. It is a complete set of formal functional tests covering all aspects of functionality within the system being built.

This type of test is often combined with:

- Security Tests: Tests that the system security and database security is appropriate for the overall system and each specific user.
- Integration Tests: Tests the overall business solution, including the passage of data to and from other integrated systems. This confirms that the functionality remains valid when all aspects of the system have been combined.
- User Acceptance Tests: Users validate that the system operates correctly and meets requirements. If users are not involved in formal system testing or they request specific tests, there may be a need for further acceptance tests. However, in most cases, this type of testing is done as part of System and Integration Tests, provided that users recognize these tests as adequate for acceptance purposes.
- Full System Test: The sequence of calculations should be identified and no calculation, run in or out of sequence, should adversely impact another calc result (or anything else).

  Load Test: Before testing with real users, conduct a load test to ensure that the application can perform under the expected user load. Load testing should be done with realistic and representative data. The performance and load testing needs the cube to be full, and the data must represent the likely sparse combination of data intersections. A very good data set is critical to a success testing effort.

Production Regression Testing

Oracle recommends that you build regression testing. You can also request automated regression testing through the Regression Testing Program. For details, see Requesting Automated Regression Testing in *Oracle Enterprise Performance Management Cloud Operations Guide*.

Roll Out

During rollout, you can train end users on the system, and show them how to navigate and use functionality.

As a best practice, document your system to enable others to take over administration as necessary.

Training

All users of the system should be trained on the application. Users need to learn how to navigate comfortably around the application and understand the tasks assigned to them. Training should include logging into the application, navigating through task lists, entering data,

Copyright © 2025, Oracle and/or its affiliates.

running rules, using Oracle Smart View for Office, and using tools within the application. Training is typically the user's first exposure to the application, and a well planned and executed training session helps make a good first impression.

Document System and Administrative Information

After building your application, it is recommended that you create system and administrative documentation for the application.

Best practices:

- Create this documentation at the end of the build process when the information is fresh.
- Include information such as the sources of data, the application structure, how calculations work, and what maintenance is required for the application.
- List maintenance tasks broken down into timeframes, such as monthly and annual maintenance. This makes it possible for someone else to take over the system later if necessary. Examples include updating substitution variables, time horizons, and resetting historical data.

Enable the Application for Users

To enable the application for end users, open up the system.

If you are using process management, start approval units to enable the approvals process. After it is started, it moves from one reviewer to another until the process is complete as described in Managing Approvals in *Administering Planning.*

Application Design

Understand design best practices for application design and deployment.

These best practices for application design help ensure a smooth and efficient process:

- Do a Requirement Analysis
- Plan for Scalability and Performance Requirements
- Plan Application Design and Integration
- Decide on a Single Application or Multiple Applications

Do a Requirement Analysis

A requirement analysis helps ensure that you meet the needs of end users.

- **Understand your users' expectations**.
  - Understand the requirements of customers and users, especially planners and users who will enter data.
  - Know what processes they want in the application.
  - Understand the current planning process and if users are coming from an Excel-based solution or using a relational application.
  - Clearly documenting user expectations helps in mapping current processes to new planning applications, ensuring better usability and performance.
- **Know their measures of success.**
  - Know what they want in terms of usability.

Copyright © 2025, Oracle and/or its affiliates.

  - Know their expectations for performance of forms and calculations, and if they require calculations that are centralized or driven by end users.
- **Design the application to meet their expectations**.
  - Evaluate their current pain points, what they want to accomplish, and their goals as they move to this new application.
  - Do not simply map current processes to the new application. Instead, take the opportunity to achieve better usability and performance.

Plan for Scalability and Performance Requirements

Planning ahead helps ensure that the application can adapt to an increased number of users and data size.

- Plan for your future needs and long-term requirements, including the number of concurrent users and the application's growth over time.
- Consider how your application will grow over time. Ask what will the size of the application be after two to three years and which dimensions will grow. Planning for those requirements in advance will help you make optimal choices when designing your application.

Plan Application Design and Integration

- Separate planning and data entry requirements from reporting requirements.
- You will typically use hybrid BSO for complex calculations and ASO for instantaneous aggregations.
- Know the level of detail you want for your plan or forecast.
- Understand all data sources, both within and outside of Planning.
- Separate functionality into different cubes or applications where appropriate. It’s best not to model everything in a single monolithic cube.
- Keep integration requirements in mind during the design phase to help meet your needs for the future.
- Evaluate the out-of-the-box modules offered by Oracle EPM and consider using those industry best practices instead of creating a custom solution. The benefits include quick implementation and being set up for long-term success.

Decide on a Single Application or Multiple Applications

Determine whether to go with a single app or multiple apps. Review these scenarios to determine the best approach for you.

Table 3-2 Scenarios for Single and Multiple Applications

Requirement	Single Application Scenarios	Multiple Application Scenarios
Number of administrators	One administrator can manage the whole planning process. For example, the same administrator manages Workforce and Financials.	There is a separate administrator for each planning process. For example, data in Workforce is not visible to the administrator of Financials.

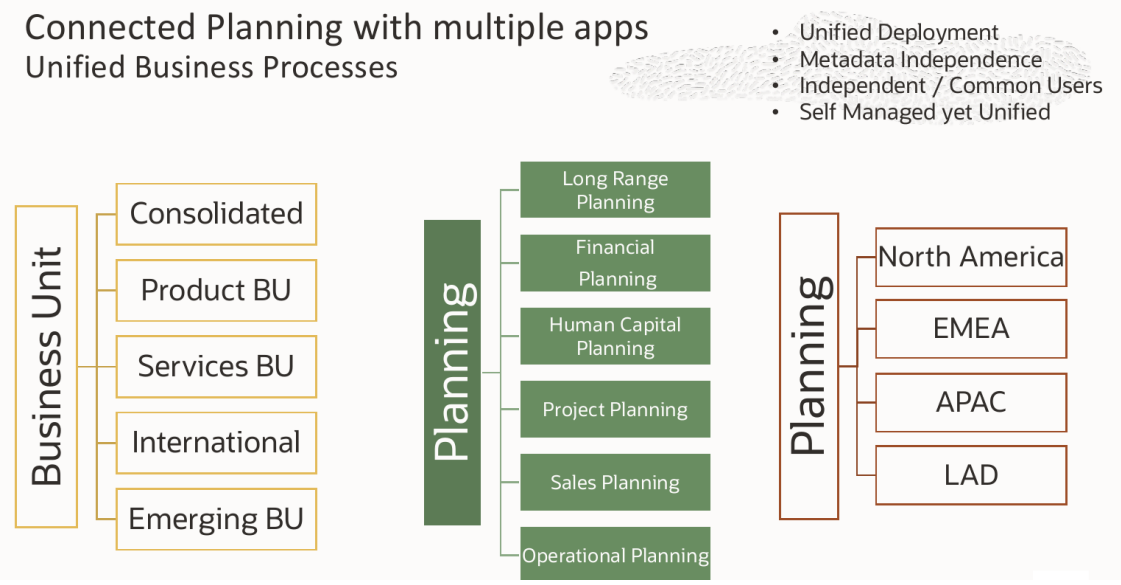
Copyright © 2025, Oracle and/or its affiliates.

Table 3-2 (Cont.) Scenarios for Single and Multiple Applications

Requirement	Single Application Scenarios	Multiple Application Scenarios
Planning business processes	Planning business processes are targeted for a single region or multiple regions. There are no variations in the planning process, such as the dimensional model and time granularity.	There is a difference in planning processes by region, or there are regulatory issues or data latency issues that require deploying separate applications in regional data centers.
Daily maintenance	Daily maintenance is not triggered during normal working business hours and does not affect application usage.	Different regions with different time zones or working hours make it difficult to schedule daily maintenance. This requires a separate application for each region.
Performance and scalability	Performance and scalability requirements can be handled with a single application.	There are requirements for performance or scalability requirements such as a large number of users or complex and time-intensive calculations.
Number of cubes	All the planning business processes can be implemented with the maximum number of cubes supported in an application.	The number of cubes supported in an application is not sufficient to implement all of the planning business processes in a single application.
Integration	The planning processes requires real-time integration.	The planning processes require infrequent or batch integration.
Users for planning processes	There are overlapping users for different planning processes.	There are separate users for different planning applications.

This example shows a sample scenario of using multiple applications.

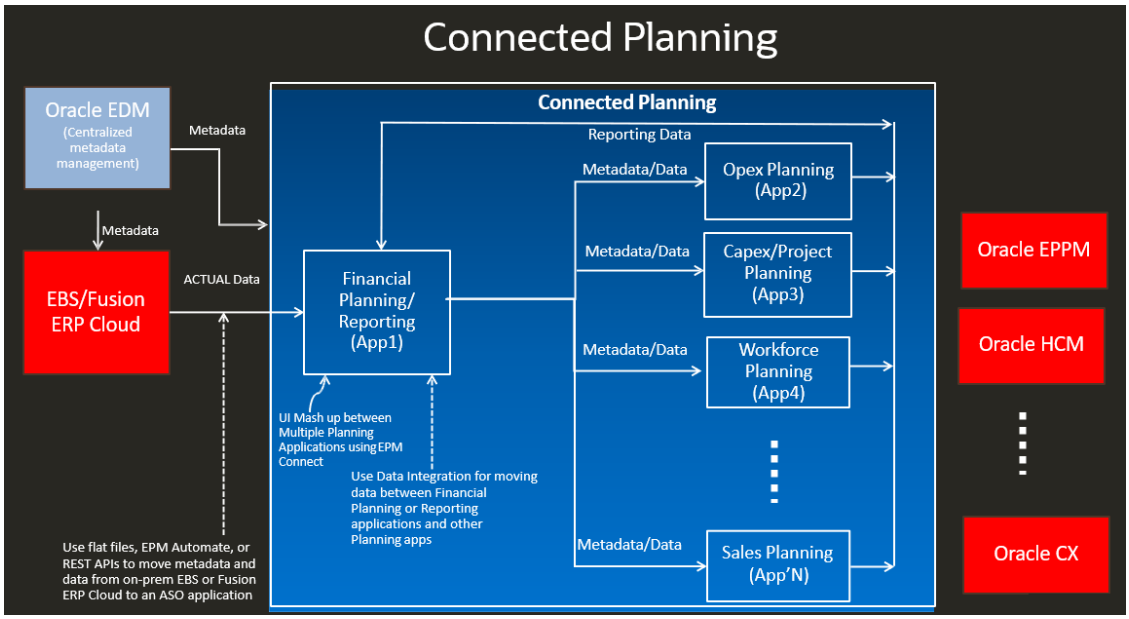
Figure 3-1 Example of Using Multiple Applications



Copyright © 2025, Oracle and/or its affiliates.

Even if your planning processes are spread across different applications, you can give a connected planning experience to your end users. For example, you can provide single sign-on, and you can move data between the applications using Oracle Enterprise Data Management to provide a seamless integration experience.

Figure 3-2 Example of Seamless Integration Between Applications



Cube Types Design

As part of the design process, gain an understanding of cube types.

Oracle Fusion Cloud EPM offers cube types that are tailored to different planning needs within organizations. Here's an overview, along with the best practices for each.

- Selecting the Application Type
- Available EPM Cloud Subscriptions
- Application Design for EPM Planning (Part IV): Application Types and Migration from Legacy SKUs

Hybrid BSO Design

During design, consider the benefits of hybrid BSO applications.

Watch this video to learn more: Application Design for EPM Planning Part II: Calculation Manager Rules and Hybrid BSO.

Why Hybrid BSO?

- Hybrid BSO provides multiple options for configuring and tuning applications.
- Application size is significantly reduced with Hybrid BSO because of dynamic sparse dimensions. This has a positive impact on performance and usability.

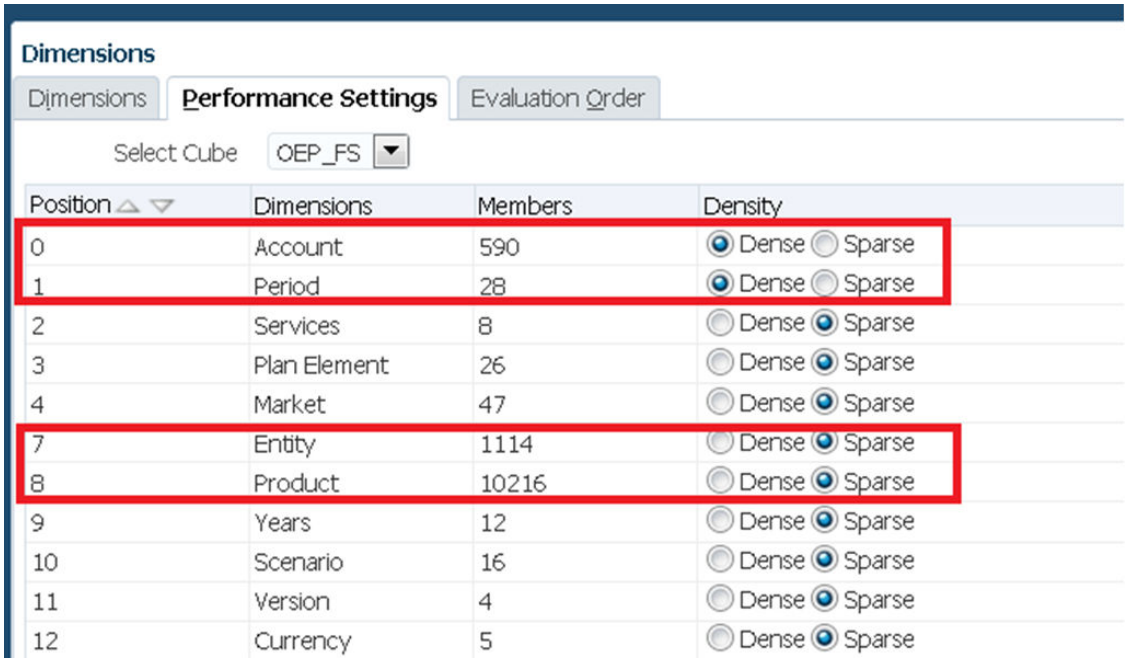
Copyright © 2025, Oracle and/or its affiliates.

- It is better than BSO but does not replace ASO for reporting purposes.
- BSO includes a best practices framework that monitors application health. It provides advance warnings for taking corrective actions to help ensure optimal application performance.

Note that hybrid BSO does not replace ASO for reporting purposes.

Example Comparison of BSO and Hybrid BSO

This example shows a comparison of BSO and Hybrid BSO for a sample application.



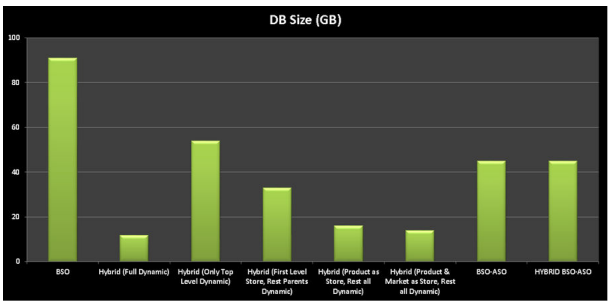
Position	Dimensions	Members	Density
0	Account	590	Dense Sparse
1	Period	28	Dense Sparse
2	Services	8	Dense Sparse
3	Plan Element	26	Dense Sparse
4	Market	47	Dense Sparse
7	Entity	1114	Dense Sparse
8	Product	10216	Dense Sparse
9	Years	12	Dense Sparse
10	Scenario	16	Dense Sparse
11	Version	4	Dense Sparse
12	Currency	5	Dense Sparse

- This example application has data for two years for all periods in Actual, Plan, and Forecast scenarios for over 30 accounts.
- The data is spread across entities and products. Each entity has data for 10 products, and each product has data for 40 market and five service members. It is fairly dense.
- Intermediate parents are added into Entity and Product dimensions, with each parent member not having more than 50 child members.
- The application size is around 11 GB (all leaf level data).

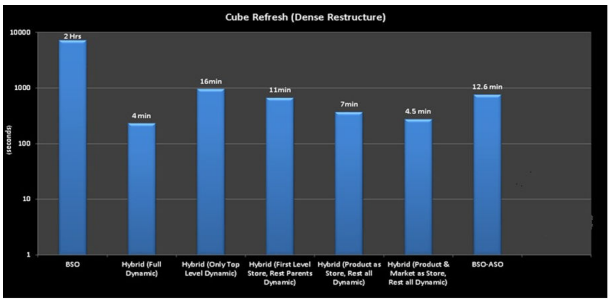
Based on this example application, here is a detailed comparison of BSO and hybrid BSO for database size, dense restructure time, rollup time, and form performance.

- **Database size**
  - BSO: 90 GB
  - Hybrid BSO: 11 GB

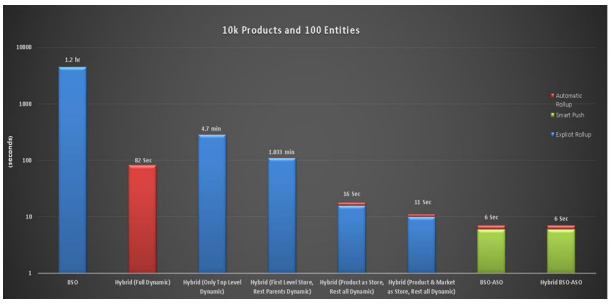
Copyright © 2025, Oracle and/or its affiliates.



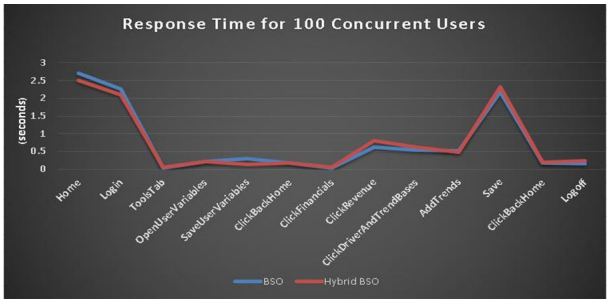
- **Dense restructure time**
  - BSO: 2 hours
  - Hybrid BSO: 4.5 minutes



- **Rollup time**
  - BSO: 1.2 hours
  - Hybrid BSO: 11 seconds



- **Form performance**



Best Practices for Hybrid BSO

All BSO best practices apply to hybrid BSO. However, there are some additional considerations.

Copyright © 2025, Oracle and/or its affiliates.

- Making all sparse dimension parent members dynamic can be considered as an option only for small-sized applications.
- A balanced approach with some sparse dimension parent members being stored or some intermediate parent members in sparse dimensions being stored gives better performance for large-sized applications.
- For very large-sized applications with large number of reporting dimensions and instantaneous aggregation and reporting requirements, you can leverage hybrid BSO-ASO with Smart Push and aggregate views in ASO.
- Leaf-level members in sparse dimensions should not be made dynamic, especially when there is no member formula associated with the member. This can have an adverse impact on form performance and make the Essbase query run as suppress missing rows and columns instead of suppress missing blocks.
- Query solve order can be set per dimension or member to ensure accurate results and to tune the performance of calculation scripts. (Aggregate before calculate wherever possible.) The higher the solve order setting for a member, the lower in the order the member is calculated. Solve order is set in the Simplified dimension editor. Here are the default solve order values:
  - Stored members 0
  - Sparse dimension 10
  - Sparse dimension – Two Pass 20
  - Dense dimension – Account 30
  - Dense dimension – Time 40
  - Dense dimension 50
  - Dense dimension – Two Pass Account 60
  - Dense dimension – Two Pass Time 70
  - Dense dimension – Two Pass 80
  - Attribute dimension 90
- Avoid creating asymmetric forms. Asymmetric grids are broken into multiple symmetric grids by the hybrid query engine, thereby affecting query performance. This is due to the serial nature of asymmetric report processing.

The Best Practices Framework

The best practices framework provides these benefits:

- Ensures application health and optimal performance by enforcing best practices.
- Provides advance warning so you can take corrective actions.
- Provides suggestions for corrective actions and following best practices.

These best practices are monitored by the framework.

- Block size
- Number of blocks
- Number of dense dimensions
- Maximum number of children under any dynamic or store parent
- Parents with single children
- Level 1 and above not set to dynamic calc or label only in dense dimensions.

Copyright © 2025, Oracle and/or its affiliates.

- Usage of dynamic x-refs
- Percentage customization

For additional information, see Improving Cube Performance in *Administering Planning*.

ASO Design

During design, understand ASO reporting and how to optimize data retrieval from ASO applications.

Watch this video to learn more: Application Design for EPM Planning (Part III): ASO, Datamaps, and Smart Push.

Optimize Data Retrieval from ASO

Follow these key best practices for ASO.

- **Enable query tracking for the cube**.
  - Run a few Smart Push activities that are representative of user activity.
  - Run any reports that query ASO data.
- **Build the optimal number of aggregate views**.
  - Perform the aggregate action again, including query tracking.
  - Using 2.00 to 10.00 times the size ratio works for most applications.
  - If query tracking for all your processes is not feasible, try using a 10 times size ratio.
  - This aggregation process needs to be executed any time a process drops the aggregate views.
  - This can be executed in a scheduled job during slow or off hours.
- **Review the dimension hierarchy types**.
  - Each ASO dimension is assigned a hierarchy type: store, dynamic, or multiple.
  - Because the hierarchy types you choose impact the number of aggregate views that can be created, Oracle recommends the following:
    - If possible, use only store type dimensions.
    - Use dynamic dimensions only if multiple or store type dimensions cannot be used .
    - For multiple hierarchy dimensions, set generation 2 to store rather than dynamic, if possible.
    - The account dimension must always be dynamic because it is the ASO compression dimension.
    - Avoid having members shared multiple times under the same parent. This forces you to make the hierarchy dynamic.

Best Practices

Follow these best practices for ASO cubes:

- Merge data slices and remove zeros. Run the Merge Slices job periodically to merge incremental slices to the main slice. Limit the incremental slices to small sizes during user activity so that the slices can be auto-merged.

Copyright © 2025, Oracle and/or its affiliates.

- Large data movements negatively affect the query performance of ASO cubes and thus slow down Smart Push. Perform several small Smart Pushes to move data into the ASO cube instead of doing one large push.
- Compact the outline.
- Review the dimension hierarchy types.
- Build the optimal number of aggregate views.

Resources:

- Optimizing Aggregate Storage Option Cubes
- Improving Cube Performance

Migrating from Legacy SKUs Design

Migrating requires careful planning and execution to ensure a smooth transition and data integrity.

For best practices on designing the migration process, see these resources:

- Migrating to Cloud EPM and EDM
  - Migration Paths for Legacy Snapshots
  - Migration Paths for EPM Standard and EPM Enterprise Subscription Snapshots
  - What Business Processes Can I Migrate to Cloud EPM?
- Migrating On-Premises Applications to Cloud EPM
- Application Design for EPM Planning (Part IV): Application Types and Migration from Legacy SKUs

Dimension Design

Designing dimensions effectively is important to ensure accurate reporting, analysis, and performance management.

Follow these best practices during dimension design.

- Design Dimensions to Create the Application Structure
- Follow this Process to Identify Dimensions
- Consider Common Use Cases for Dimensions
- Review Example Design Strategies
- Know the Top Ten Best Practices for Dimensions
- Plan the Entity Dimension
- Plan the Account Dimension
- Plan the Version Dimension
- Plan the Currency Dimension
- Plan Exchange Rates
- Plan the Period Dimension
- Plan Years and Substitution Variables

Copyright © 2025, Oracle and/or its affiliates.

- Design Custom Dimensions
- Additional Best Practices

Design Dimensions to Create the Application Structure

Add accounts, entities, and other dimensions to support your business process.

Dimensions categorize data values. Planning includes these dimensions: Account, Entity, Scenario, Version, Period, and Years. If you plan in multiple currencies, your application also has a Currency dimension.

You can use the Custom dimension to define your own values, such as Product, Customer, or Market. You can have up to 32 total dimensions. However, the best practice recommendation is to include fewer than 12. You can add dimensions using a load file, or you can build them in Oracle Smart View for Office.

Videos

Your Goal	Watch This Video
Learn how to export and import data in the application.	Exporting and Importing Data in Oracle Planning and Budgeting Cloud
Learn how to load dimensions using a file.	Importing Metadata in Oracle Planning and Budgeting Cloud

Follow this Process to Identify Dimensions

Use this process to identify which dimension to include in the application.

1. **Identify unique planning processes based on requirements**.
   Example: Marketing Planning, Sales Planning, Overhead Planning, Capital Planning, Cash flow Planning, Workforce Planning
2. **Identify dimensions for each planning process**.
   Example: Product, Market, Channel, Product Segment, Customer Segment
3. **Define how dimensions are related to each other**.
   Example: Product has a many-to-many relationship with Market. Product Segment and Product have a one-to-many relationship. Labor Resources and Material Resources have no relationship.
4. **Separate dimensions into buckets for planning and reporting**.
   Example: Product, Market, and Channel are Planning dimensions, while Product Segment and Customer Segment are reporting dimensions.
5. **Map planning processes to Planning Modules**.
   Example: Configure Marketing Planning using Projects or Financials, and configure Overhead Planning and Cash flow Planning using Financials and custom cubes.

Consider Common Use Cases for Dimensions

Review these common use cases for dimensions, and understand the guidelines for how to address them.

- **Standard (mandatory) dimensions**
  Most of the high level and common Financial Planning use cases can be addressed with standard dimensions, which can also be renamed.
- **Custom or optional dimensions**
  Extend dimensionality with custom or optional dimensions that can be added (enabled) or renamed per your requirements.

Copyright © 2025, Oracle and/or its affiliates.

- **Multiple hierarchies in a dimension**
  Combine two or more unrelated dimensions into a single dimension to avoid inter-dimensional irrelevance.
- **Alternate Hierarchies for Planning or Reporting**
  Use alternate hierarchies when the same members can be grouped under different parents for top-down allocation or reporting purpose.
- **Attribute dimensions for reporting**
  Attribute dimensions are useful to meet reporting requirements if they are related to one dimension and the relationship between the two dimension does not change over time.
- **Smart Lists and ASO dimensions for reporting**
  Using Smart Lists and ASO dimensions for reporting is useful when the relationship between the reporting dimension and other dimensions changes with time.
- **Smart Lists and multi-cube BSO dimensions for planning**
  This strategy is useful when the planning dimension is not a primary dimension for a planning process and there is a need to break it into sub-processes.

This sample worksheet shows an example of how to plan dimensions, including indentifying dimensions and listing their use cases.

	A	B	C	D	E	F
1	**Planning Process**	**Dimensions**	**Description**	**Is Planning vs Reporting?**	**Relation with other dimensions**	**Dimensionality Options**
17	Marketing Planning	Activity	Standard Activities for Campaigns	Planning	Plan activities by Campaign and Marketing expenses at activity level. Many to many relation with Product, Campaign, Customer & IO dimensions	Custom
18	Marketing Planning	Product	Hierarchy/List of Products (not SKU)	Planning	Plan marketing expenses by Product. Many to many relation with Campaign, Activity, Customer & IO dimensions	Custom
19	Marketing Planning	Customer	List of Major Customers	Planning	Plan marketing expenses by Customer. Many to many relation with Campaign, Activity, Product & IO dimensions	Custom
20	Marketing Planning	IO	Member on the fly generated for marketing expenses	Planning	Generate IO for marketing expenses. Many to many relation with Campaign, Activity, Customer & Product dimensions	Custom
21	Revenue Planning	Product	Hierarchy/List of Products (not SKU)	Planning	Plan Revenue by Product. Many to Many relation with Channel & Customer. Many to one relation with Product Segment dimension. No relation with customer segment	Custom
22	Revenue Planning	Channel	List of Sales Channels	Planning	Plan Revenue by Channel. Many to Many relation with Product & Customer. No relation with Product Segment & Customer Segment	Custom

Review Example Design Strategies

Review these examples to understand additional design strategies for dimensions.

Using attribute dimensions for reporting

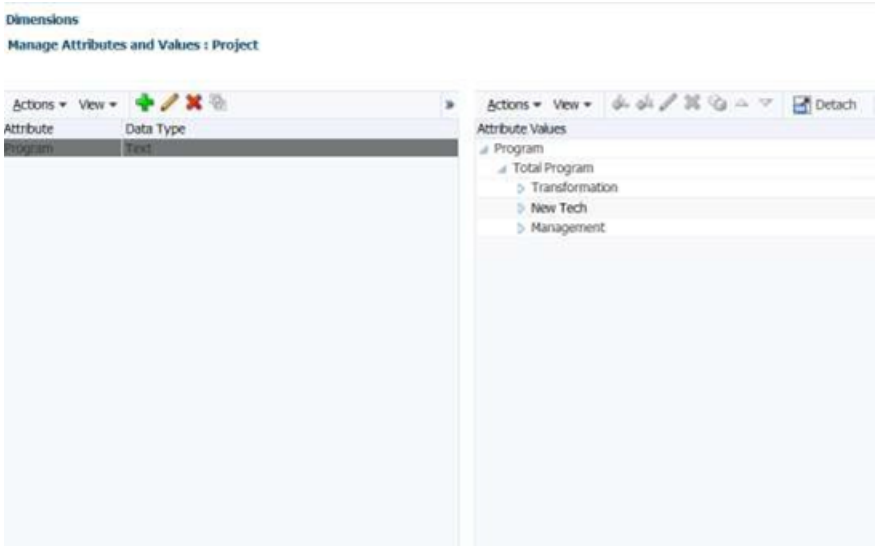
Using attribute dimensions for reporting can be useful to meet reporting requirements. The attribute is related to one dimension, and the relationship between the two dimension doesn't change with time.

Example:

- Define an attribute called Program on the Project dimension.

Copyright © 2025, Oracle and/or its affiliates.

- You can then create a hierarchy of members in the Program dimension. Top level members in the Program dimension will be automatically aggregated.
- Associate each Project member to a leaf level member in the Program dimension in an Add Project Groovy rule.
- This allows you to filter projects by program.
- Reporting forms can show program level Expenses and Revenue.



Note that using too many attribute dimensions on a form or report may degrade performance. This should be included in the performance and load testing aspects of an implementation.

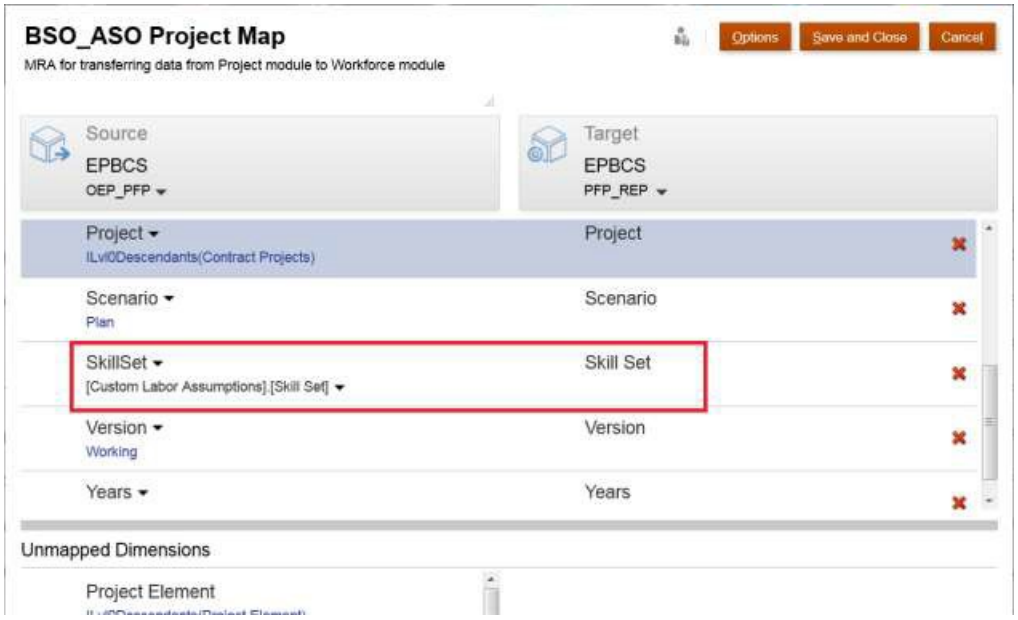
Using Smart Lists and ASO dimensions for reporting

This strategy is useful when the relationship between the reporting dimension and other dimensions changes with time.

Example:

- Add a Skillset dimension in ASO and map the Skillset Smart List to the Skillset dimension.
- Create a data map for moving data from BSO to the ASO cube.
- The data map can be run as a batch process, or you can implement Smart Push in forms or Groovy rules.
- You can use reporting forms on the ASO cube to showing the labor requirement by Skillset across projects.

Copyright © 2025, Oracle and/or its affiliates.

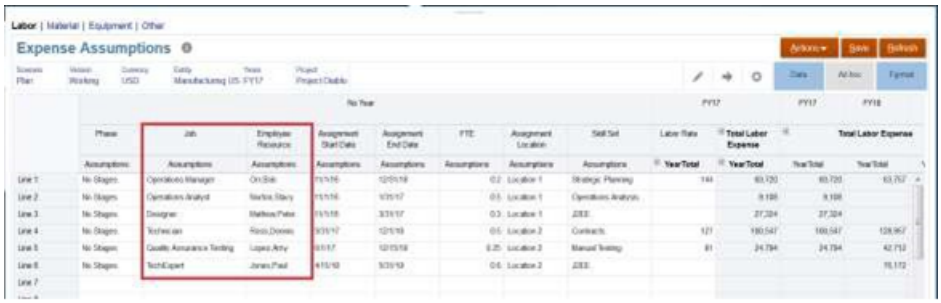


Using Smart Lists and multi-cube BSO design

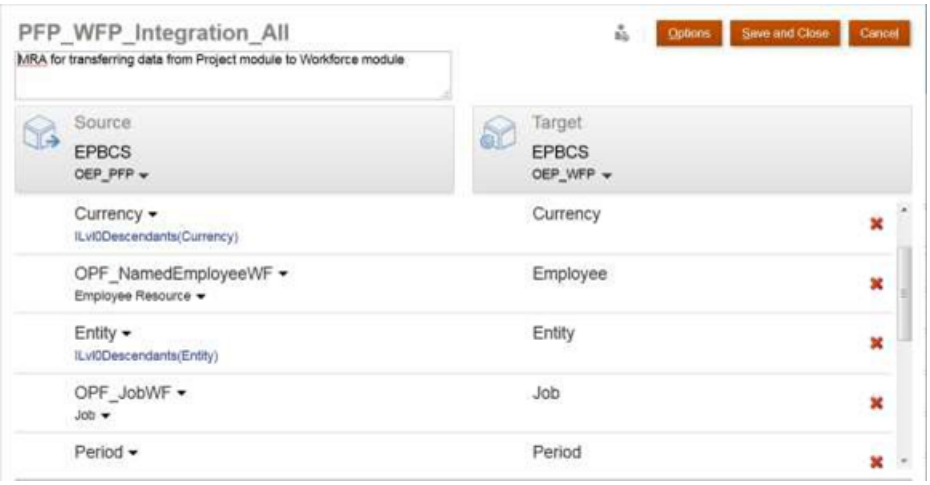
This strategy is useful when the planning dimension is not a primary dimension for a planning process and there is a need to break it into sub-processes.

Example:

- Employee and Job are dimensions in Workforce and Smart Lists in Project Planning.
- Project Planning uses line items to plan labor expenses at the Job and Employee level.



Copyright © 2025, Oracle and/or its affiliates.

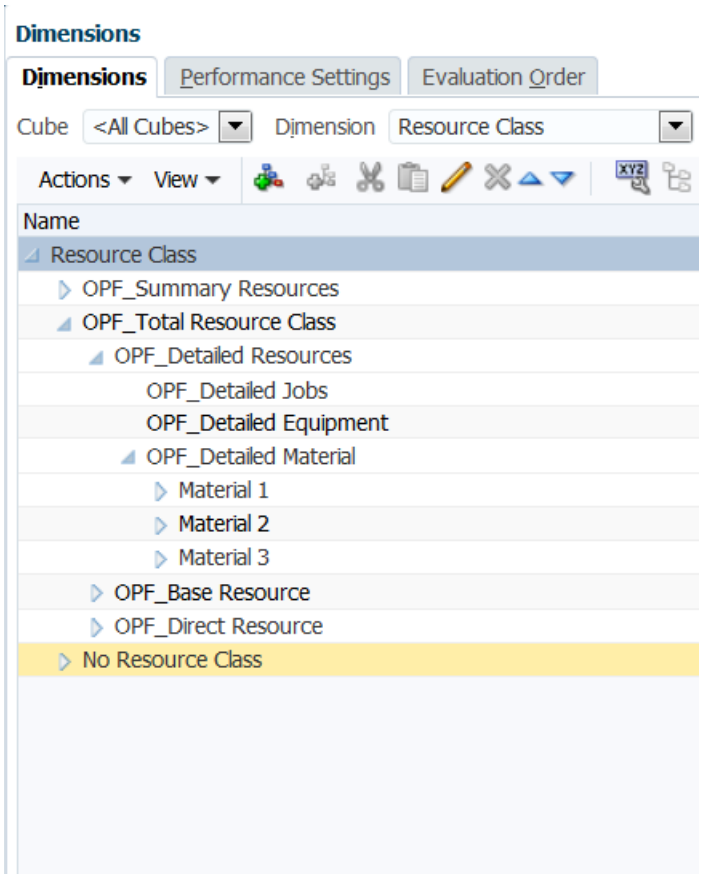


Multiple hierarchies in a dimension

You can use multiple hierarchies in a dimension. Combine two or more unrelated dimensions into a single dimension to avoid inter-dimensional irrelevance.

Example:

- Jobs, Equipment, and Material are unrelated in Projects, so they are combined into a single Resource Class dimension.
- Profit Centers and Cost Centers are unrelated in Financials, so they can be combined into a single Entity dimension.



Copyright © 2025, Oracle and/or its affiliates.

Alternate hierarchies for planning and reporting

Use alternate hierarchies when the same members can be grouped under different parents for top-down allocation or reporting purposes.

Example: Create alternate rollups in the Product dimension for planning and reporting by the Brand and Product categories.

Product	
Total Product	
All Product	
TM_NA	TM_NA - TM Not Assigned
TM_05	TM_05 - Coca-Cola
TM_06	TM_06 - Aquarius
TM_07	TM_07 - Fuze
TM_08	TM_08 - Glaceau
TM_25	TM_25 - Fanta
TM_30	TM_30 - Sprite
TM_35	TM_35 - Minute Maid
TM_45	TM_45 - Georgia
TM_50	TM_50 - Powerade
TM_56	TM_56 - Nestea
TM_58	TM_58 - Canada Dry
TM_59	TM_59 - Crush
TM_60	TM_60 - Dr Pepper
TM_61	TM_61 - Schweppes
TM_62	TM_62 - TCCC-All Other Schweppes Brands
TM_97	TM_97 - Dasani
TM_98	TM_98 - TCCC-All Other Brands
TM_99	TM_99 - All Other Manufacturers - All Brands
TM_A0	TM_A0 - Fuze Tea
No Product	
All Category	ALL - All Beverage Categories
CL_01	CL_01 - Sparkling Soft Drinks
CL_02	CL_02 - Energy Drinks
CL_03	CL_03 - Juices Dairy and Plant Based
CL_04	CL_04 - Water Sports Drinks
CL_05	CL_05 - Tea Coffee

Rollup by Brand

Rollup by Categories

Know the Top Ten Best Practices for Dimensions

Follow these important best practices when designing dimensions.

1. Dimension order should follow a modified hour glass format.
   With this format, the most dense dimension is the first dimension, followed by less dense dimensions. This should be followed by sparse dimensions, with aggregating sparse dimensions before non-aggregating sparse dimensions. Within sparse dimensions, you should have the most dense sparse dimensions before the least dense sparse dimensions.

   With hybrid BSO, the order is the same, except that you should have non-dynamic sparse dimensions before dynamic sparse dimensions.

   Learn more about dense and sparse dimensions in this tutorial: Managing Dimensions in Cloud EPM.

2. Large block size has a major impact on performance.
   Block size is determined by the number of dense dimensions and the stored members in those dense dimensions. Optimal block size is between 8 KB to 500 KB. Reduce the number of dense dimensions to a maximum of 3. Level 1 and above should be label only or dynamic calc for dense dimensions.

3. Text, Smart List, Date, and Stored Percentage type accounts should be set to Never for the consolidation property.

Copyright © 2025, Oracle and/or its affiliates.

Aggregating these values, unless the accounts are explicitly excluded in an aggregation script, will create useless data and make the cube unnecessarily large, which likely will degrade performance.

4. All generation two members should be set to Ignore.
   You cannot include these root members in your forms because you cannot define security for those members. So there is no point in aggregating the generation two members to the root. This will also increase the number of blocks in your application.
5. Long or flat dimensions will lead to an issue with the performance of aggregation.
   If there are more than 200 members under a parent member, add intermediate parents.
6. Enabling dimension members for multiple cubes will create dynamic X-Refs and lead to performance issues.
   Use the HSP_NOLINK UDA to avoid creating dynamic X-Refs. Use data maps or Smart Push for moving data between cubes.
7. For simple calculations, leverage outline math instead of writing member formulas.
   An example of a simple calculation is Account C = Account A – Account B.
8. When possible, avoid single child parent members.
   Single children parent members lead to implied shares or duplicate blocks and data on disk if the parent member is made Never Share.
9. Aggregate large dimensions in ASO instead of BSO whenever possible.
   This improves performance for example during cube refresh and maintenance time.
10. Store historical data beyond two years in ASO instead of BSO.
    If you have 5 or 10 past years of historical data, not all of the data is needed for calculations. If needed, you can have a couple of years of historical data for calculations in your BSO cube, and you can move other historical data to the ASO cube. For optimum performance, it is a best practice to keep the BSO cube light and ensure that it is focused on the calculations for data entry.

Plan the Entity Dimension

The Entity dimension represents your organizational structure, such as Cost Centers, Departments, Business Units, Divisions, and so on.

You can group Cost Centers by creating rollup members, called parents, to reflect how your organization is viewed. For example, rollups can be by business unit, division, or other functional structure. As an example, you could create Cost Centers that roll up to Business Units that roll up to Divisions.

You can also create multiple reporting structures. For example, an alternate structure could be created to support regional reporting. If you plan in multiple currencies, set the base currency of each entity.

The Entity dimension is one of the primary dimensions used for the budgeting process. Together with the Scenario and Version dimensions, the Entity dimension is used to define an approval unit, a discrete component that can be promoted or demoted for approval or review by a user's peers. Members of all dimensions outside the approval unit are promoted and demoted along with the approval unit itself. For example, all twelve months are promoted together when an approval unit is promoted. Individual months can't be promoted independently.

A secondary dimension is also supported, as described in Approvals Design Considerations. In addition, you can include phased approvals, as described in Managing Approval Phases.

After each dimension is loaded or updated, it is a best practice to refresh the application.

Copyright © 2025, Oracle and/or its affiliates.

Plan the Account Dimension

The Account dimension is the place for your chart of accounts. It should include the members to which you plan or forecast. It doesn't necessarily include every account in your chart.

For example, your Account dimension could include accounts for Income Statement, Balance Sheet, and Cash Flow. Or, it could include accounts for KPIs and Ratios. In some cases, your accounts may have sub accounts, but this is not typical.

The Account dimension includes financial intelligence. The following account types are supported:

- **Expense**: Cost of doing business
- **Revenue**: Source of income
- **Asset**: Company resources
- **Liability and Equity**: Residual interest or obligation to creditors
- **Saved assumption**: Centralized planning assumptions ensuring consistency across the application

The account type settings are used to report Quarterly and Year Total values and for variance analysis.

Planning uses a hierarchical structure to create Account grouping subtotals and totals. Each account group is assigned a consolidation operator that determines how it rolls up to its parent.

Example:

Net Income = Total Revenues - Total Expenses

In this example, the consolidation operator for Total Revenues is Addition, and the consolidation operator for Total Expenses is Minus. Intelligence determines, in part, the sign in which data is loaded, entered, or calculated.

The Account dimension can be populated either by loading data or using Smart View. To load data from a file, the file format must meet specific requirements.

After each dimension is loaded or updated, it's a best practice to refresh the application.

Best practices:

- Upper level members should be set to Dynamic Calc or Label Only.
- For member formulas used to calculate Ratios and other types of KPIs or percentages, set them to Dynamic Calc, Two Pass. The Two Pass setting properly calculates Percentages at upper levels.

Plan the Version Dimension

You can use versions to preserve different iterations of the planning process. Versions are also useful for controlling data access to Read or Write.

These two types of versions are available:

- **Standard Target**: Input data can be entered to upper levels.
- **Standard Bottom Up**: Input data can be entered to level 0 only.

Approvals and workflow functionality can be enabled only for Bottom Up versions.

As a best practice, these versions are recommended:

Copyright © 2025, Oracle and/or its affiliates.

- **Working**: Where users perform their tasks, including reviewing Actual Results and developing Plan and Forecast.
- **1st Pass**: If you want to maintain multiple iterations of your Plan, you can preserve a pass of it in this version. You can create other members if you require more than one saved iteration. You can leverage the Copy Data functionality to move data to this version. Copy data copies data and textual input.
- **What If**: Provides a placeholder where users can change assumptions and analyze the outcome.

After each dimension is loaded or updated in the build process, it's a best practice to refresh the application.

Plan the Currency Dimension

If you enabled multiple currencies for your application, you can add the currencies you use to plan and report.

You can then define exchange rates by scenario and year to be used in conversions. A calculation script is created that enables you to perform currency conversion. To enter exchange rates, follow the process in Specifying Exchange Rates in *Administering Planning*.

Best practices:

- Limit the number of reporting currencies. Typically, customers have only one.
- Enter exchange rates for each valid scenario and year combination.
- From this point on, currency conversion can be calculated by running the Calculate Currencies business rule that is associated by default with each form.

Run the currency conversion calc script prior to:

- Reviewing any updated local data in reporting currencies
- Running certain calculations that may be dependent on reporting currency data

Plan Exchange Rates

Each application has a default currency that you specify when creating the application. When you set up exchange rate tables, you enter exchange rates from all source currencies to the default. Triangulation is used to convert to all other Reporting currencies.

Exchange rates are set by Scenario by year for Average and Ending Rates.

Plan the Period Dimension

Use the Period dimension to establish the calendar’s range within a given year, for example, by month.

Best practices:

- Use substitution variables for this dimension to support reporting and calculations. Potential substitution variables are: CurrMo, CurrQtr, and PriorMo. These variables must be updated on a monthly basis.
- To use time period calculations such as Year to Date (Y-T-D) or Quarter to Date, select the dynamic time series icon in the Period dimension. You can then select which time period calculations you need to support your process.
- Summary time periods such as quarter totals and a year total should be set to dynamic calculate to reduce calculation time.

Copyright © 2025, Oracle and/or its affiliates.

- After each dimension is loaded or updated, refresh the application.

Plan Years and Substitution Variables

Years are incorporated into the application in many places, including forms, calculations, reports, and Smart View. Because you'll use the application for many years into the future, the best practice to referencing this dimension is by using a substitution variable.

Substitution variables act as global placeholders for information that changes regularly. The variable and value correspond to the year, and the value can be changed at any time.

The value of the substitution variable is displayed on forms and reports as a placeholder. This reduces maintenance for the application.

As a best practice, create substitution variables for each year that is included in your process. For example:

- CurrY, Current Year
- NextYr, Budget (Plan) Year
- PriorYr, Prior Year

Design Custom Dimensions

You can use a custom dimension to further categorize your data. For example, custom dimensions might include Product or Markets.

Keep in mind that access permissions can't be granted at the dimension level, also called generation one. For example, access permissions can't be assigned directly to the Product member for all descendants. If you enable security for your custom dimension, it is recommended that you design generation two for all custom dimensions to which security will be applied with security access assignments in mind.

After each dimension is loaded or updated, it is a best practice to refresh the application.

Additional Best Practices

Complete these tasks after you add or update dimensions.

- **Refresh the application.**
  You must refresh the application whenever you change the application structure.

  Changes made to the application are not reflected to users performing data entry and approvals tasks until the application is refreshed.

  For example, when you modify properties of an Entity member or add a Scenario, these changes are reflected to users after you refresh the application.
- **Load historical data.**
  After you load all of your structures, such as accounts and entities, you can load historical data. This can include data from prior year actual results and current year plan and budget.

  Loading historical data provides users a way to analyze results, review trends, and make meaningful comparisons.

  This also helps verify the structures that you have built into your application. For example, you can verify that data ties to previously created reports. If the data doesn't reconcile, you must verify if this is caused by a data issue or if there is an issue with the structures.

  Create an aggregation rule to see consolidated data in your application.
- **Plan valid intersections.**

Copyright © 2025, Oracle and/or its affiliates.

Valid intersections let Service Administrators define rules called valid intersection rules that filter dimensional intersections for users when they enter data or select runtime prompts. For example, you can specify that certain programs are valid only for specific departments. Leverage valid intersections to control data entry only for valid intersections.

During form design, keep these points in mind for valid intersections.

- If dimensions with valid intersections are on the Page, the user will only be presented with valid combinations in the member selector.
- If dimensions with valid intersections are on the column or row, the form designer can completely suppress invalid intersections. When the suppression option is not selected, invalid intersections are set to read only.

Best Practices for Designing Business Rules

Poorly written rules have a major impact on all aspects of an application. Below are some best practices when designing your business rules. Following best practices recommendations can lead to significant performance benefits.

Follow these key design guidelines for business rules.

- Top 10 Best Practices for Rules
- Add Business Logic Using Calculations
- Build Aggregations
- Set the Point of View
- Build Detailed Calculations
- Calculation Manager Diagnostics
- Example Rule Issues and Solutions

Top 10 Best Practices for Rules

Follow these best practice recommendations when designing your business rules. These guidelines can lead to significant performance benefits because poorly written rules have a major impact on all aspects of an application.

1. Follow these guidelines for SET commands:
   - At the top of the rule, do not use `SET CREATEBLOCKONEQ ON` or `SET CREATENONMISSINGBLK ON`.
   - Do not use administrative type commands like this in end-user rules, as this requires a restructure: `SET CLEARBLOCK EMPTY`.
   - Avoid or test rules using `SET CALCTASKDIMS`. (Oracle Essbase typically does this automatically.)
   - The Data Copy rule should include the following to prevent copying empty blocks within the Fix statement: `SET COPYMISSINGBLOCK OFF`
2. Block Creation should be done using either Datacopy or Sparse Member assignment. The functions `@createblockoneq` and `@createblock` should be used as a last resort inside a limited Fix statement.
3. Avoid missing dimension references in a Fix statement (for example, in Data Copy Rule). This can waste processing time and increase contention and create unnecessary blocks for all levels of the missing dimensions.

Copyright © 2025, Oracle and/or its affiliates.

4. Remove parallel calculation in business rules associated with forms. Calc Parallel or Fix Parallel should be used only with administrative/batch rules.
5. Do not create unnecessary zeros, as this leads to block and data explosion. Carefully review business logic and add necessary If conditions to check for zeros. Convert zeros to `#missing`. After this, a dense restructure is required to remove the blocks.
6. Eliminate multiple passes on the same blocks; instead, set a proper outer fix and move in and out as necessary. Combine If statements instead of using and re-using If on the same intersections.
7. Avoid using cross-dimensional references on the left side of an equation. This has a performance impact.
8. Aggregate dimensions in order of creation of the most blocks to the least blocks in the script, for example, Agg (1st Most Blocks Dimension, 2nd Most Blocks Dimension, 3rd Most Blocks Dimension). Agg is faster than Calc Dim and is a preferred mechanism for aggregation. Aggregation using `@ancestors` in end user rules all the way to the top of the dimension can lead to block contention.
9. Use run-time prompts instead of creating multiple rules with the same underlying logic. More rules means more maintenance.
10. Use templates for breaking down and reusing business logic. However, templates should not be fully functioning rules with Fix, EndFix. The rule combining various templates should have a proper outer Fix and move in and out of smaller pieces as necessary.

Add Business Logic Using Calculations

To incorporate your business logic into your application, you can build calculations using Calculation Manager. This lets you create, validate, deploy, and administer sophisticated calculations that solve business problems.

You typically create business rules and rulesets to:

- Perform revenue modeling
- Perform expense modeling
- Calculate KPIs
- Perform allocations

Calculation Manager includes these objects:

- Rules: Contain components and templates
- Components: Assist you in building rules
- Rulesets: Contain rules that can be calculated simultaneously or sequentially
- Templates: Include system templates that perform calculations and custom templates that can be designed by administrators

To learn more about creating calculations, see the guidelines in Designing with Calculation Manager for Oracle Enterprise Performance Management Cloud.

Build Aggregations

Aggregations roll up your application to summary-level members in the dimension, such as Entity or any other sparse dimension.

Calculation Manager includes templates to help you build aggregations. Here are some suggestions on how to use templates.

Copyright © 2025, Oracle and/or its affiliates.

Set the Point of View

When the point of view is set, the rule will run only for the selected members. Using a runtime prompt for the dimensions allows users to specify member values for these dimensions when launching the rule. This way, users can launch the rule several times for different years, scenarios, and versions without having to modify the rule in Calculation Manager.

Typical settings:

- Full dense aggregation: Complete this section if parent values in your dense dimensions are not set to dynamic calc. Typically this tab is left empty.
- Full sparse aggregation: Select the sparse dimension that needs to be aggregated. The order of the selected dimensions is not relevant.
- Partial dimension aggregation, Dense: Complete this section if parent values in your dense dimensions are not set to dynamic calc. Typically this tab is left empty.
- Aggregate the data up to the local currency: No
- Aggregate the missing values in the Database: Yes

  Be careful when using this option with a Target Version where data is entered at the parent members and descendants will be #Missing.
- Optimize the Calculation on Sparse dimension: Off
- Select a value for the calculator cache: Default
- Do you want to activate the debug mode for this wizard?: Debug Wizard On or Debug Wizard Off. Select Debug Wizard On if you want to see a script generated to display selections for some of the Design Time Prompts in this template.

Best practices:

- Leverage runtime prompts for members such as Entity, Scenario, and Version. This allows your rule to be dynamic and run based on user input.
- Typically, dense dimensions such as Account and Period don't need to be aggregated. If this is the case, you can set parent members to dynamic calc. However, if you have member formulas on dense dimensions and they are not set to dynamic calc, a Calc Dim rule is required.

Build Detailed Calculations

You use Calculation Manager to create, validate, deploy, and administer calculations that solve business problems.

There are three types of objects that can be calculated in Calculation Manager:

- Rulesets: Contain rules that can be calculated simultaneously or sequentially
- Rules: Contain components and templates
- Components: Contain formula components, script components, condition components, range components, and fixed loop components

Best practices:

- As a first step in building your rules, ensure that you understand the business logic and which entities or departments the rule applies to. For example, know the accounts that are involved in the rule.
- Be sure you know the source and destination accounts.

Copyright © 2025, Oracle and/or its affiliates.

- After you fully understand the drivers of the calculation, use the proper object component or template to build the rule. The components and templates facilitate member selection to help deploy the rules.
- Leveraging runtime prompts for members such as Entity, Scenario, and Version allows your rules to be dynamic and run based on user input.

Calculation Manager Diagnostics

Run errors and warnings before deploying rules. This provides helpful information, including:

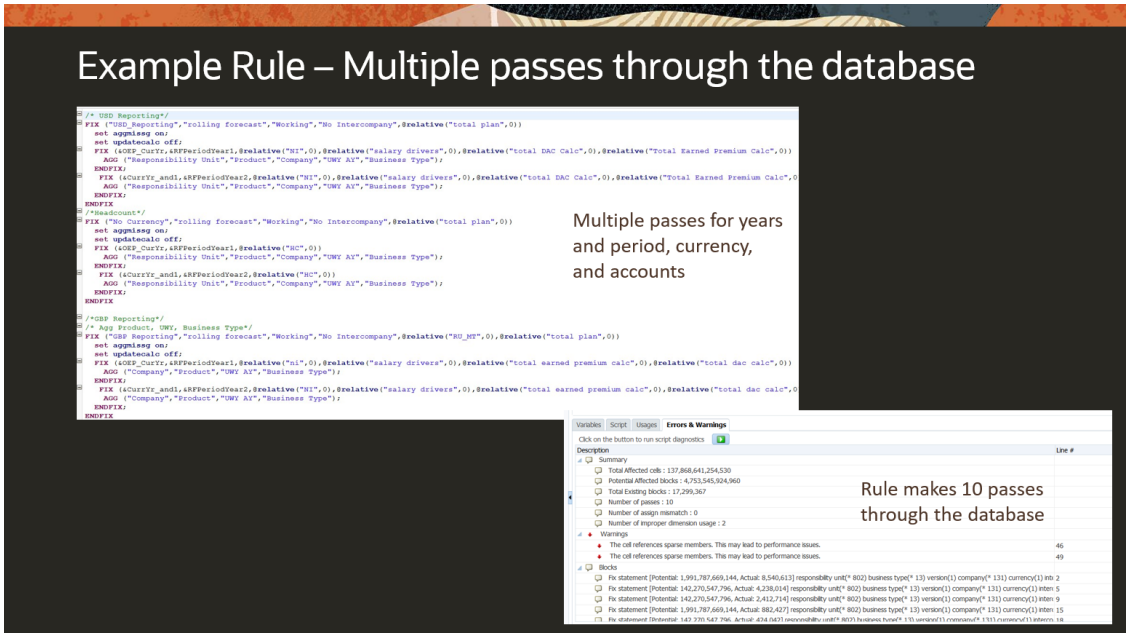
- The number of passes through the database
- Any necessary warnings
- Information on the number of blocks and if dimensions are missing
- Any rules that need to be optimized
- If any of the components of the right side of the equation contain a zero, the derived member will be 0. Then after aggregation there are many 0's.
- To address this, the rule should include an if statement such as the following. This way, zeros will not be so significant in the application.

```
if ("Earned Premium"<>0)
```

Example Rule Issues and Solutions

Example 1: Rule that Makes Multiple Passes Through the Database

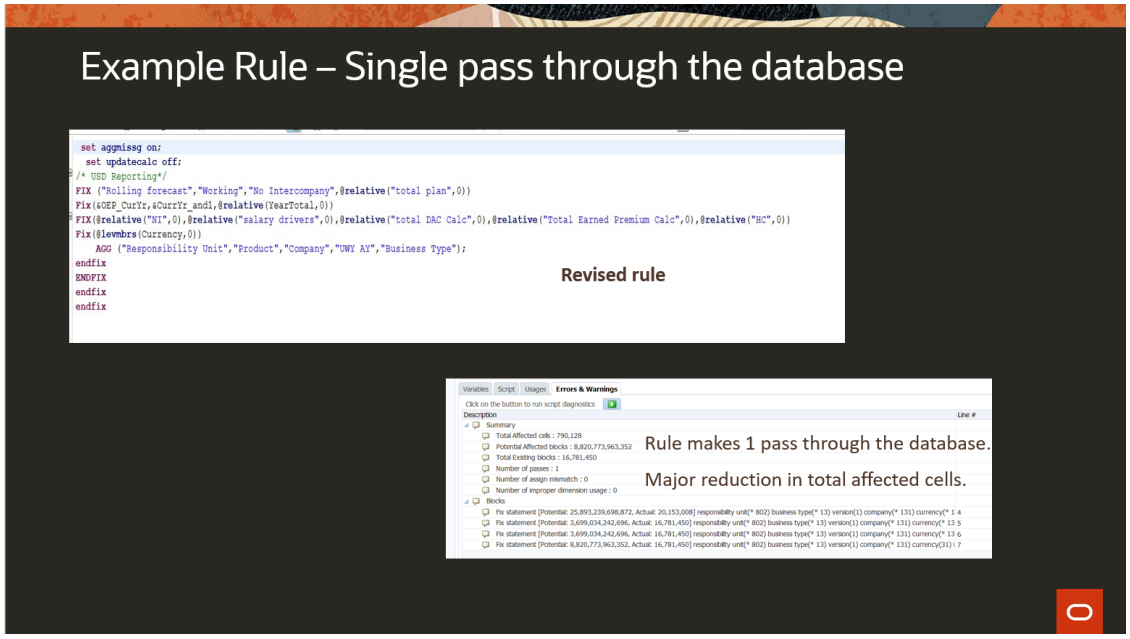
The following rule makes 10 passes through the database for years and period, currency, and accounts.



Example 2: Rule that Makes Only One Pass through the Database

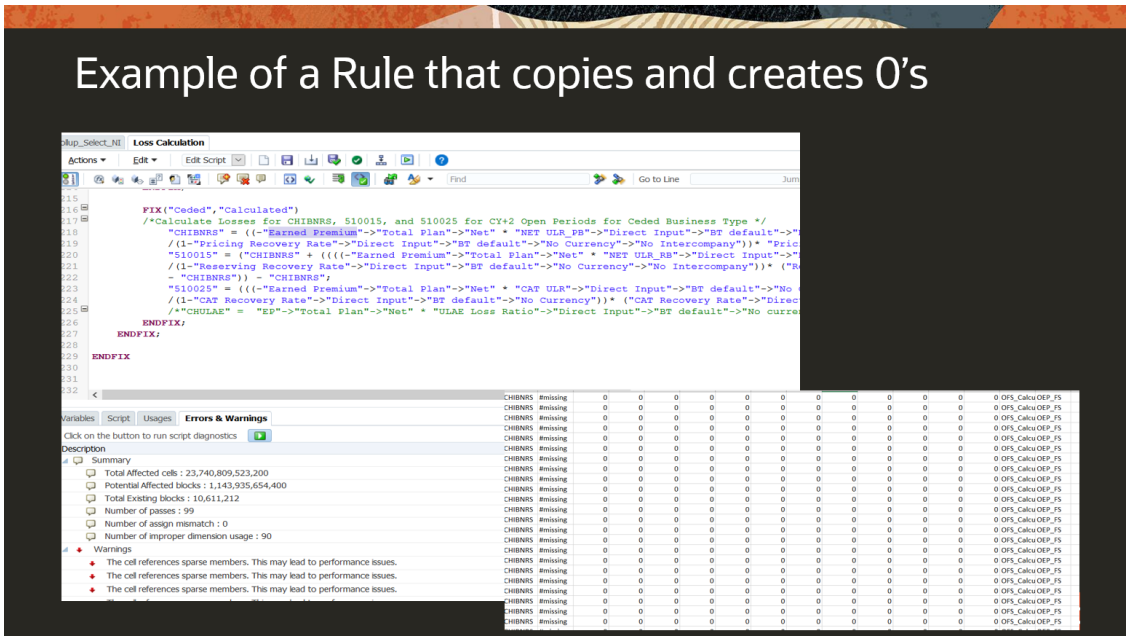
The following rule makes just one pass through the database. This is a major reduction in the total affected cells.

Copyright © 2025, Oracle and/or its affiliates.



Example 3: Rule that Copies and Creates Zeros

In this rule, if any of the components of the right side of the equation contain a 0, the derived member will be 0. After aggregation, there will be many zeros. To address this, the rule should include an if statement saying `if ("Earned Premium"<>0)`. This way zeros would will not be so significant in the application.



Example 4: Removing Zeros from BSO Cubes

The following formula results in the original value, or, if 0, will change to #missing.

```
"DenseMbr" = "DenseMbr" * "DenseMbr"/"DenseMbr";
```

```
"SparseMbr" = "SparseMbr" * "SparseMbr"/"SparseMbr";
```


Copyright © 2025, Oracle and/or its affiliates.

To address this, create a rule leveraging the formula. Use a sparse dimension that has the fewest members (such as Year, Scenario, or Version) because the calculation needs to be done for each member.

```
Fix(idescendants(Entity),@levmbrs(Accounts,0),@levmbrs(Period,0),@idescendant
s(Custom dimensions),
@levmbrs(Years,0),@levmbrs(Versions))
Actual=Actual*Actual/Actual;
   Fix("Actual")
      CLEARBLOCK EMPTY:
   Endfix
Endfix
```

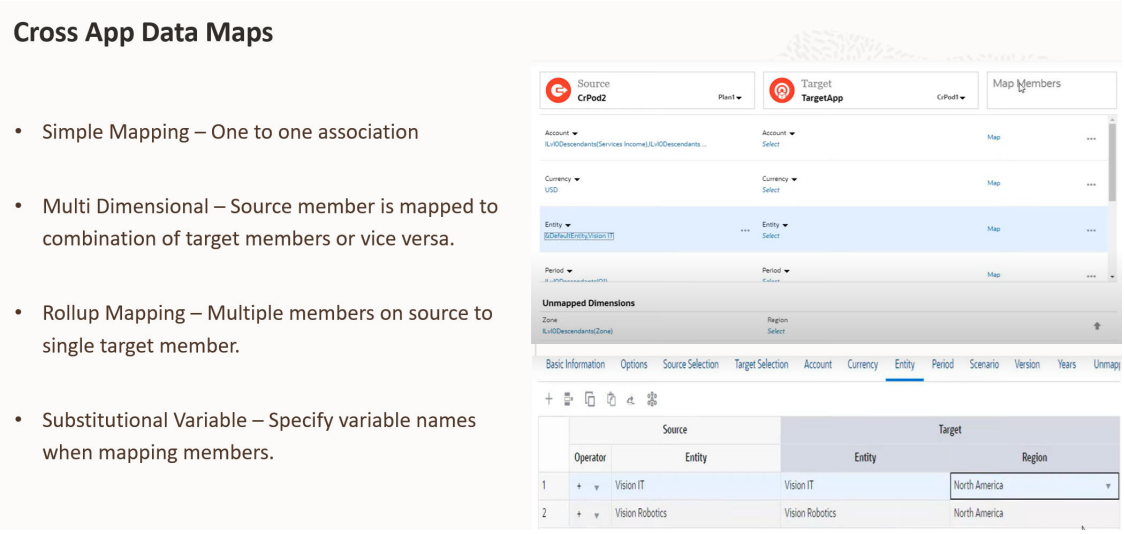
Remember to do a restructure afterward. Otherwise, the empty blocks will not be deleted.

Data Maps Design

Designing effective data maps is crucial for ensuring accurate data integration and transformation during data loads.

Include data maps in your design to realize these benefits:

- Trickle feed data for real time reporting
- Synchronize cell commentary, supporting details between cubes and applications
- Real-time data synchronization
- Instant data movement based on user changes
- Trickle feed data from one app to another across instances
- Support member level mapping, substitution variables, and cross-dimensional combination members
- Smart List as a source while doing member mapping



Follow these best practices for effective data map design, as described in [Application Design for EPM Planning (Part III): ASO, Data Maps, and Smart Push]:

- Trickle feed into ASO with small slices of data for best performance.

Copyright © 2025, Oracle and/or its affiliates.

- Smart Push enforces a limit of data movement for optimal performance. Work within this limit.
- Large data movement using Smart Push should be limited to only a few concurrent users.
- Use the database suppression option to suppress missing data at the Essbase level.
- Perform large slices of data movement using data maps.
- Use Smart Push to move data during user activity .
- Smart push gets precedence:
  - A high degree of concurrency for data push and Smart Push will cause contention
  - Data push jobs will wait
- Use Groovy to set overwrite selection to only modified data for grids with sparse members on rows.

Get additional design guidelines for data maps:

- Understanding Data Maps
- Defining Data Maps
- Moving Data

Smart Push Design

Designing effective Smart Push strategies enhances data integration and ensures seamless data transfer between applications.

Watch this Customer Connect presentation to understand how to design for Smart Push: Application Design for EPM Planning (Part III): ASO, Data Maps, and Smart Push.

Review these guidelines for best practices in designing Smart Push.

- About Smart Push
- Moving Data from One Cube to Another Cube Using Smart Push
- Instantly Push Data for Reporting in Planning

Form Design

Use these design guidelines to plan effective forms.

Form Design Overview

You'll build a number of forms to support data entry and summary-level reports. The form content is similar to the templates you use to collect and calculate data. The layout may differ from what you are accustomed to in spreadsheets.

To enhance usability, group forms within major categories such as Revenue, Compensation Expense, Other Expenses, and so on. You can create some forms to support data entry, and others for summary and review. You can also include charts to help users analyze results.

Form performance is based on several factors, including network and environmental factors, structure, and layout.

Copyright © 2025, Oracle and/or its affiliates.

Design Considerations

Design forms to enable users to enter information such as revenue, expenses, and assumptions.

Best practices:

- Group accounts logically, but don't include too many accounts on a single form.
- Limit the number of entry forms to an amount comfortable for end users. A delicate balance needs to be achieved between the number of accounts on a single form and the number of forms required to support your process.
- Use detail forms to enable users to enter all related information. All accounts that require input should be found on a form. The accounts can be broken down into several different forms.
- While building forms, ensure that you select all appropriate options to enhance the design of your form. For example, use settings to control precision, display, and menus, and to associate the proper rules with your form.
- Use Substitution Variables to reference dimensions such as Years.
- Suppress invalid Scenario/Time Period option, set Periods in row or column on the form to the Start and End Period set for the Scenario. Leveraging this functionality can be used instead of substitution variables for Years.
- Consider setting valid intersections to set relationships between different dimensions. Suppressing invalid combinations can be set in row or column to make only valid intersections available to end users. By default, only valid intersections will be available to the end users when the dimensions are set in the Page selection.
- Use relationships to incorporate members onto the forms instead of picking members individually.
- Consider using User Variables for dimensions such as Entity and Scenario to help reduce the dimension selection for end users.
- If your application supports multiple currencies, consider setting a User variable so users can define their base currency.
- Organize forms into folders.
- Use substitution variables to reduce the maintenance on forms.
- Put dense dimensions such as Account and Period on the row and column of a form. Put sparse dimensions such as Entity on the Page axis.
- Dimensions such as Scenario or Version and Year can reside on the POV, column, or row. It's important to properly gauge how columns or rows will be returned when a user opens the form.

Associate Rules with Forms

Associating rules with forms enables users with appropriate access to launch associated business rules from the form to calculate and derive values.

You can associate multiple business rules with a form by cube. Business rules associated with a form can be set to launch automatically when the form is opened or saved. You can select Use Members on Form to populate runtime prompts from the current form instead of prompting users for input when rules are launched.

Best practices:

Copyright © 2025, Oracle and/or its affiliates.

- For rules that take longer to run, set them to launch from an Action menu or simply through association with the form.
- If a business rule has runtime prompts, limit the number of prompts to keep the user's job simple.

Add Menus to Forms

You can associate menus with forms. Action menus allow users to click rows or columns in forms and select menu items. For example, they can launch a business rule, with or without runtime prompts, or move to another form.

Menus are context-sensitive. The menus that display depend on the form settings and where users right-click in the form.

Best practices:

- When designing forms, use Other Options to select menus available for Form menu item types.
- As you update applications, update the appropriate menus. For example, if you delete a business rule referenced by a menu, remove it from the menu.

Build Data Validation Forms

Data validation can serve as a visual clue to users that business policies have been met. You can add conditional color coding to forms, and validation messages can be generated if entered data violates validation rules or if a condition is met.

Defining data validation rules involves these main tasks:

- Identify the data cells or location that you want to display with validation messages or in different colors when conditions are met. Be mindful of accessibility concerns if color is the only differentiator.
- Identify the cell, column, or row that needs to participate during rule evaluation, and define the rule accordingly.
- Create the data validation rule at the location identified.

Organize Forms into Folders

Use folders as a way to organize the forms in your application. Forms can be grouped in folders by process or user type, or simply to help users readily find forms. You can move forms into folders, and you can create a folder hierarchy. Creating folders also simplifies assigning access because all forms in the folder will inherit the access permissions assigned.

Build Summary Level Forms

Summary level forms typically bring together all of the pieces of a user's plan or forecast. They enable users to review and analyze their results.

Using dashboards can also be an effective way to help users analyze their results.

Forms and Cubes

When you create a form, you associate it with a cube, which determines the form's valid members. For example, if you assign a form to the Revenue cube, you can add only accounts that are valid for the Revenue cube. Entered data is saved to the selected cube's database.

Copyright © 2025, Oracle and/or its affiliates.

> **Note**
>
> - You can't change a form's cube after assigning it.
> - You can edit form accounts only if their source cube matches the form's cube.
> - If you add an account to a form associated with a cube other than the account's source cube, the account is read-only on that form.

Forms and Permissions

Assign permissions to a form to determine which users can modify its design (for example, layout and instructions) and input data. Users can input data on forms only if they have permission to one secured dimension's member. For example, if users have read-only permission to the Europe entity, the rows and columns that include the Europe entity are read-only. Users can change data only for members to which they have write permission.

Forms and Versions

For bottom-up versions, rows and columns with level 0 members allow data entry. Rows or columns set to a parent member are read-only. The point of view must also be set to the level 0 member to allow data entry on a bottom-up version. Target versions allow data entry in parent and children members set to Store.

Filtering Form Members by Attributes

You can select members by using attributes. For example, on the Entity dimension you can select members by a specific Region such as South. The resulting grid will only contain members that have the South attribute (for example, TX, NM, and so on). Values can be entered and saved into rows and columns filtered by attributes.

Forms and Shared Members

Because you can't select shared members individually, select them using a relationship function. For example, select an alternate functional rollup to include all members under that rollup. Users can enter values in rows or columns that display shared members, and data is saved to the base members in the database.

Forms and Calculations

To optimize calculations, select row members using relationships (such as Descendants or Children) instead of selecting individual children. For example, calculating individual parent-level totals could take several passes, so use a relationship instead.

Dashboards Design

Dashboards allow you to display information graphically or to display several forms simultaneously. Designing effective dashboards requires a thoughtful approach to ensure that they are user-friendly, insightful, and aligned with business objectives.

Follow the design best practices in these topics for successful dashboard design:

- Concepts in Designing Dashboard 2.0 Dashboards
- Considerations for Dashboard 2.0
- Dashboard Design Guidelines

Copyright © 2025, Oracle and/or its affiliates.

- Recommendations for Improving Dashboard Design

Also consider these high-level guidelines when designing dashboards:

- As a best practice, balance the number of components on the dashboard to ensure that it is visually pleasing to the user.
- You can design interactive multi-chart dashboards to enable users to analyze their plan or forecast data.
- You can also display a grid and a graph together, or you can combine multiple grids.
- Use dashboard settings to combine dimensions into a common POV.

Figure 3-3 Example Dashboard with a Grid and Charts

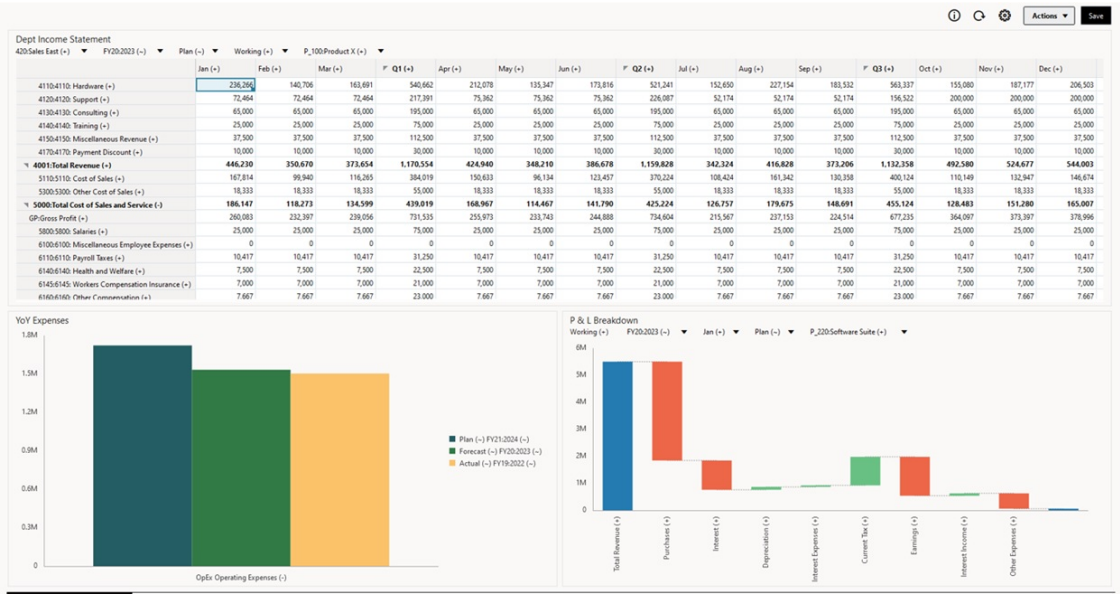
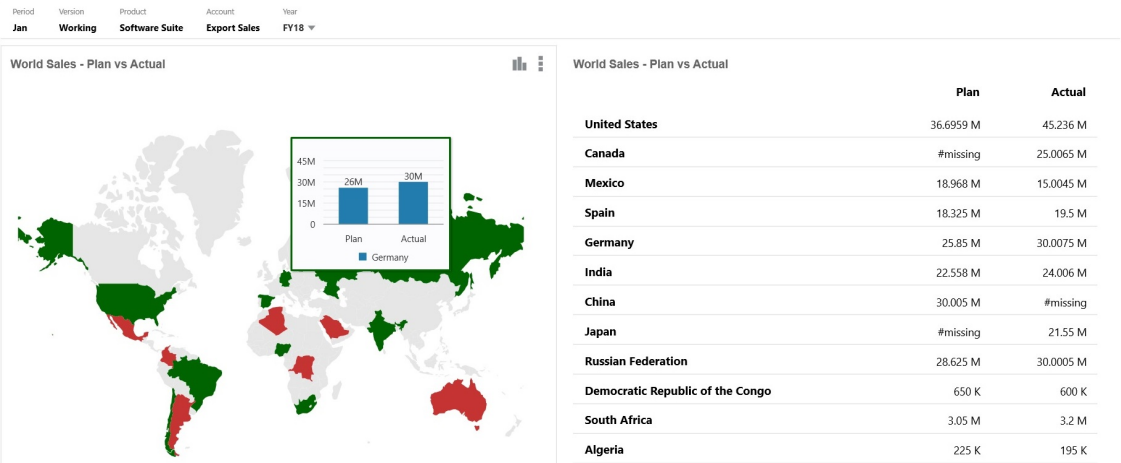


Figure 3-4 Example Geomap - World Sales Using Color Highlight



Copyright © 2025, Oracle and/or its affiliates.

Approvals Design

Designing dimensions effectively is important to ensure accurate reporting, analysis, and performance management.

Follow these practices when designing approvals.

Build Approvals

Use approvals to track budgets and review status, planning unit ownership, and process issues. This reduces the time required for the planning cycle.

Set up the approval path independent of organizational structure to reflect the path a plan or forecast must follow for approval.

Users can provide annotations and comments for their submissions.

Set Up the Approvals Hierarchy

Setting the approvals hierarchy defines the promotional path used in approvals. The basis of the hierarchy is the Entity or any part of the Entity dimension in combination with any secondary dimension.

The secondary dimension can be a mix of several dimensions, depending on where you are in the workflow. For example, for some entities, you can combine the Entity dimension with the Products dimension in the promotional path, and for other entities you can use the Channels dimension in the promotional path.

You can directly assign the approvals unit to owners and reviewers. You can create validation rules to handle conditional promotion paths that are dependent on data conditionals. You create different hierarchies to support review processes within your organization.

The hierarchy is then assigned to the appropriate Scenario and Version combination.

Planning units are combinations of scenario, version, and entity or part of an entity. Scenarios and versions are the basis of the review cycle. The hierarchy contains approvals units and any other dimensions that are part of the review process.

Things to know about approvals:

- The review process follows the promotional path you set up when you select the owner and reviewers for a planning unit, unless an event triggers a change in the promotional path.
- Parent/child relationships between members affect the review process
- When users promote or reject a parent, the parent’s children are promoted or rejected unless they are approved. The owner for the parent becomes the owner of the children.
- When users approve a parent, its children are approved.
- After all children are promoted to the same owner, the parent is promoted to the owner.
- When the status of all children changes to one status, for example Signed Off, parent status changes to the same status.
- Users can't change the status of a parent if its children have different owners.
- If the children are promoted to, submitted to, or signed off by different users, the parent has no owner and only Service Administrators can change its status.

Copyright © 2025, Oracle and/or its affiliates.

- The approval unit moves from one reviewer to another until the budget process is complete.

For additional information, see Managing Approvals.

You can also include phased approvals, as described in Managing Approval Phases.

Tasks Design

Design tasks to guide users through the planning process with Task Manager to provide assignments, instructions, and due dates.

Consider these design best practices for tasks.

- Design tasks to guide users through the application to ensure that the process is followed and that all of the proper data has been collected.
- Use tasks to guide users through the planning process.
- Build tasks to support the different types of users and process flows, for example:
  - Open a form
  - Launch a business rule that you specify
  - Start the review process with a specified scenario and version
  - Copy a version of the current form's data
  - Open a specified URL

Navigation Flow Design

Designing an effective navigation flow ensures that users can easily access and utilize the features and data they need.

Use these best practices to design helpful navigation flows.

Navigation flows set the clusters or cards available at the top of the user screen. The cards are typically associated with actions in your business process, such as Plan Revenue and Plan Expenses. Within each card, vertical tabs can be created to lead the user through the process for that business area. Forms can be linked to a vertical tab to guide the user in the process. Vertical tabs can have one or many horizontal tabs linking to either forms or dashboards.

Your application comes with a default navigation flow. To customize the cards and flow for your organization, copy the default, and then you can use it to make your own.

You can create a cluster to represent an entire business process that can contain cards for the actions, or you can create new cards. Cards can be designed to be single page or can have multiple tabs. For a card that is set up to be tabular, you can have multiple tabs that allow you to have content visible to the end user as horizontal tabs. You specify the content type for each of the tabs and link to an artifact.

For example, you can associate cards with:

- Dashboards
- Forms
- Rules
- Approvals

When customizing navigation flows, keep these guidelines in mind:

Copyright © 2025, Oracle and/or its affiliates.

- Predefined Navigation Flows
- Editing a Navigation Flow
- Customizing Labels for Cards, Tabs, and Clusters

It is a best practice to associate navigation flows with groups. You can then tailor a navigation flow to apply to a particular group. Keep in mind that users can be part of more than one group, so they can have access to more than one navigation flow.

Periodically validate navigation flows. See Using Validate to Find Missing Artifacts in Navigation Flows.

You can create translation labels for clusters, cards, or tabs in the Artifact Labels page. See Customizing Labels for Cards, Tabs, and Clusters.

For more information, see Understanding Navigation Flows and Designing Navigation Flows in Cloud EPM.

Access Permissions Design

Setting up access permissions effectively is important to ensure secure access and efficient management.

Follow these design best practices.

Set Up Access Permissions

Access permissions determine a user's privileges after the product launches. Most often, groups are established to help organize users. By definition, a user group is a set of users with similar access permissions.

For details on the topics in this section, see Managing Users and Roles.

You can assign access permissions for groups and individual users to these application elements:

- Scenarios
- Versions
- Accounts
- Entities
- Custom dimension members
- Forms
- Business rules and templates
- Dashboards
- Reports

Users can be in a group:

- Service Administrator
- Power User
- User
- Viewer

Best practices:

Copyright © 2025, Oracle and/or its affiliates.

- For dimensions secured by default, modify access permissions as needed.
- Assign access permissions to application elements such as dimension members, forms, and rules. Users can view or use only those application elements to which they have access.

Set Up Users and Groups

Your company's users must be added to the Oracle Identity Management System prior to gaining access permissions to any of the elements in your application. Access permissions determine a user's privileges after the product launches.

By definition, a user group is a set of users with similar access permissions. The use of groups as a way to organize users and assign access permissions is a best practice.

Add Users

Users must be added to your environment, assigned privileges, and granted access to the application.

Users' roles will be defined as one of these types:

- Service Administrator: Creates and manages applications, including dimensions, forms, Calculations, and so on. The Service Administrator manages access permissions and initiates the budget process
- Power User: Creates and maintains forms, Oracle Smart View for Office worksheets, and Financial Reporting reports. Enables you to manage business rule security. Can perform all User tasks.
- User: Enters and submits plans for approval, runs business rules, uses reports that others have created, and views and uses task lists. Leverages Smart View to enter data and do ad hoc analysis.
- Viewer: Views and analyzes data through data forms and any data access tools for which they are licensed. Viewers can't modify any data in the application. Typical View users are executives who want to see business plans during and at the end of the budget process.

Users, Power Users, and Viewers can access forms, task lists, and business rules based on permissions assigned by the Service Administrator.

Create Groups

It's highly recommended to leverage groups when assigning access permissions to users. Having groups of similar users eases security maintenance on an on-going basis. As users are added to groups, they inherit the access permissions of the group. Assigning group access permissions to elements such as dimension members, forms, and task lists means that you don't need to assign those access permissions individually for each user.

Best practices:

- If an individual user is assigned to a group, and the access permissions of the individual user conflict with those of the group, the individual user's access permissions take precedence.
- The use of groups for sets of users with similar access permissions should be well defined prior to implementing user access.
- Individual permissions override group permissions.
- If an individual is assigned to multiple groups, the group with the most restrictive access takes precedence.

Copyright © 2025, Oracle and/or its affiliates.

Access permissions assigned to a user directly override access permissions inherited from groups that the user belongs to. For example, if you have inherited read access to Plan from a group but are assigned write access to Plan directly, you get write access to Plan.

Assign Users to Groups

As a best practice, leverage groups as a way to reduce maintenance and assign similar access to users. Give users access to the appropriate groups.

Assign Access to Dimensions

In order for users to read or write data, assign access permissions to these dimensions:

- Account
- Entity
- Scenario
- Version

If security is enabled on custom dimensions, you must assign security to users to those dimensions as well. For dimensions secured by default, modify security access as needed.

Assign Access to the Account Dimension

Give users read or write access only to those accounts they are allowed to see. You can assign access privileges as Read, Write, None, or Display.

Best practices:

- Relationship functions should also be leveraged whenever possible to reduce on-going security maintenance. The relationship functions are: Member, Children, iChildren, Descendant, and iDescendant. For example, assigning Write access to Descendants of Net Income for a group allows all users of that group to have Write access to all accounts that are descendants of Net Income. This way, you don't need to assign access individually to each account.
- To take full advantage of the rules of precedence and inheritance, use an exception-based method for managing security. Primary assignment of security should be by group and relationship. Assign group rights to parent level members, and use relationships to push the assignments down to the children or descendants. Assign individual user rights to children on an exception basis.

Assign Access to the Entity Dimension

Give users read or write access only to those entities they are allowed to see. You can assign access privileges as Read, Write, None, or Display.

Assign Access to the Scenario Dimension

Access to Scenario is typically set to Read or Write. For example, you may want to assign access to Actual and Variance scenarios as Read, and to Plan and Forecast as Write.

Assign Access to the Version Dimension

Access to Version is typically set to Read or Write. For example, you may want to assign access to Final Version as Read, and to Working as Write.

Copyright © 2025, Oracle and/or its affiliates.

Assign Access to Custom Dimensions

If security is enabled on any custom dimension, you must assign security to the dimension in order for users to have access.

Assign Access to Forms

Before users can open forms, they must be assigned access permissions.

Users who are assigned access to a form folder can access the forms in that folder unless they are assigned more specific access.

Users and Power Users can view or enter data only into forms to which they have access. They can work only with members to which they have access.

Tips:

- To simplify assigning access to forms, organize forms into folders and assign access at the folder level instead of the individual form level. Access permissions can be set to Read, Write, or None. Write access to a form is only valid for a Power User and enables editing the structure of the form. Read allows users to enter data based on their dimensional permissions.
- When you assign access to a folder, all folders under it inherit that access.
- If you assign specific access (such as None or Write) to a form folder, that access permission takes precedence over its parent folder's access permissions. For example, if a user has Write access to Folder1 that contains Folder2 to which the user has None access, the user can open Folder1, but doesn't see Folder2.
- If a user has None access to a form folder called Folder1 that contains a form called Form1 to which the user has Write access, the user can see Folder1 and Form1.

Assign Access to Business Rules

Before users can launch business rules, they must be given access permissions to the rules.

As a best practice, organize business rules into folders that have similar user access, and apply security to the folders. You can also give access permissions to individual business rules, although this is a little more time-consuming.

Users have Launch access to Calculation Manager business rules in folders to which they are assigned access, unless they are assigned more specific access

Assign Access to Task Lists

In order to navigate through the application, users must be assigned access to individual task lists.

As a best practice, assign access using groups. This is more efficient than applying access to individual task lists.

Assign Access to Dashboards

Assigning access to dashboards allows you to control with users can view or interact with specific dashboards. This ensures that users see the right insights based on their requirements.

Assign Access to Reports

Users must be assigned access to a report before they can use it.

Copyright © 2025, Oracle and/or its affiliates.

As with other artifacts, it's recommended that you organize reports into folders and assign access at the folder level. This limits the amount of maintenance required for security. As reports are added to the folder, access is inherited from the folder.

Security Design

Setting up users, groups, and access permissions effectively is crucial for ensuring secure access and efficient management.

Understand access permissions, including user and role management, application artifacts that can be assigned permissions, types of access permissions, and managing and reporting on access permissions. See Setting Up Access Permissions and Defining Cell-Level Security in *Administering Planning*.

Learn about managing users and configuring security settings in *Administering Access Control for Cloud EPM*:

- Managing Users and Roles
- Configuring Security Settings

Get an understanding of access control, managing role assignments, and generating reports in *Administering Access Control for Cloud EPM*:

- Overview of Access Control
- Managing Role Assignments at the Application Level
- Generating Reports

Watch the videos on Customer Connect:

- Approval Groups and Phased Approvals
- Cell Level Security for Cloud EPM Applications.

Complete the tutorials:

- Setting Up Security in Planning
- Setting Up Security in Cloud EPM

Reports Design

Designing effective reports helps support useful insights and decision-making. Here are some best practices for report design and management.

Follow these general design guidelines:

- Before building reports, plan how to report on your financials for management and determine how many different report formats are required. Report formats specify the layout of the report, such as which elements are on the rows and columns. Report formats can be used to create many different reports, such as by cost center or division.
- To simplify building reports, specify a report format for each type of report you need. For example, build your income statement and other detailed reporting with the formatting that your management team is accustomed to reviewing.
- First set the report dimension layout and member selections. Next, get the report to capture the data. Finally, apply formatting.

Answer these questions to determine the most effective design:

Copyright © 2025, Oracle and/or its affiliates.

- How many individual reports are required, where a single report (such as an Income Statement) can be run for multiple POVs?
- What is the maximum number of users that will be running reports frequently within the same time frame at peak times? What is the frequency of running different sets of reports?
- What types of cubes are going to be the data source for the reports? For example is an ASO reporting cube going to be utilized or BSO/BSO? If you are using BSO/BSO, consider the query performance impact of using dynamic calc storage on sparse dimension parent and formula members. Do any of the cubes contain a large amount of dimension members that need to be reported on?
- Try to avoid writing relational-type reports, where possible. A good indication of relational-type report is one with multiple row dimensions that are expanded using member selection functions such as Descendants or Bottom Level which return a large number of cells. A report is considered big when the number of cells starts getting into the tens of thousands and above. Consider using Suppress Missing Blocks with this type of use case.
- How many books and bursting definitions are required?

  Are larger books required, which contain many reports or reports run for a large number of members? Consider the book governor limitations and potentially longer book processing times when creating large books.

  What is the number of members bursted across a dimension for larger bursting definitions?

  How many bursting definitions are planned to be run at peak times? Do you plan on running bursting definitions using EPM Automate or REST APIs? If so, consider the performance impact when doing this.

For additional information, refer to Designing Reports in *Designing with Reports for Oracle Enterprise Performance Management Cloud*.

Integration Design

Integrating Oracle Fusion Cloud EPM with other systems helps streamline operations and ensure accurate data flow.

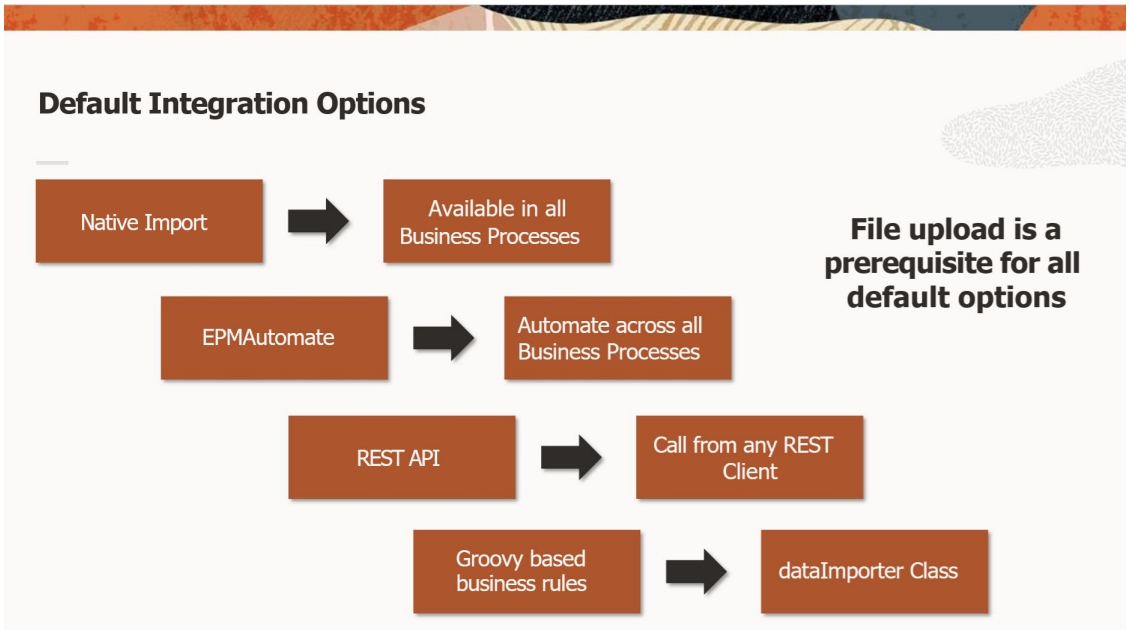
Use these resources to design effective integration:

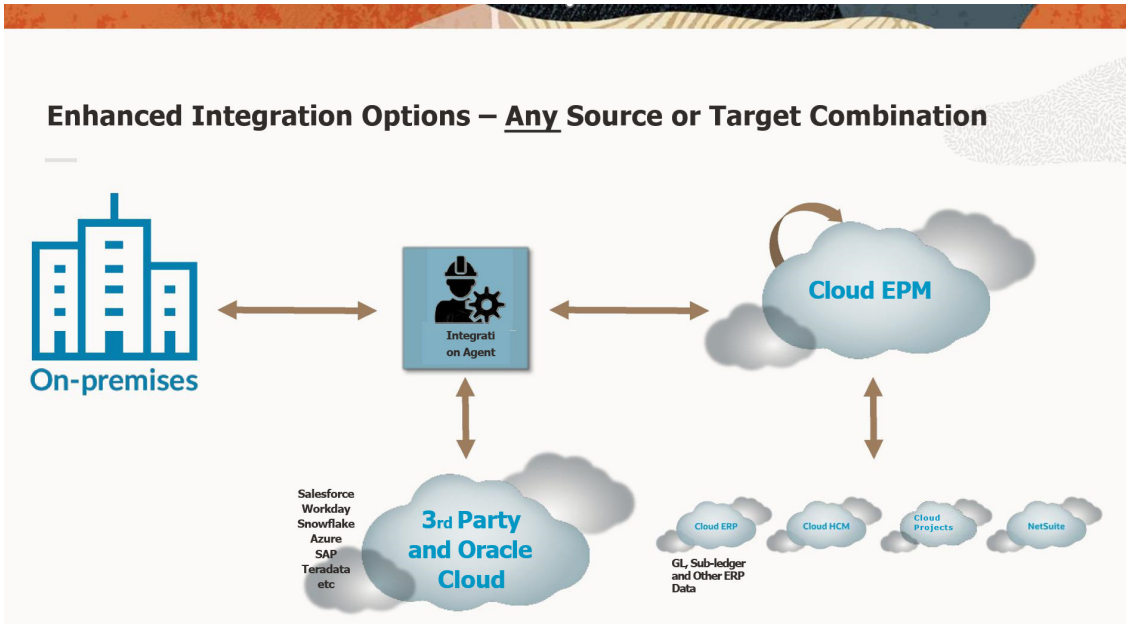
- Integrating Data in Cloud EPM
- Data Integration
- EPM Integration Agent
- Data Integration Best Practices
- About Integration for Cloud EPM

In addition, implement the integration guidelines in this webinar on Cloud Customer Connect, EPM Data Integration Best Practices and Roadmap:

- Start integration work early in the project.
- Discuss integration as data flows: Source, Target, Transformations, and Service Level Agreements.
- Recognize that first-time load performance may not be optimal.
- Use log level 5 during your development work.
- Make frequent backups, and include snapshots. Save load and log files separately.
- Understand how to read log files: Performance Tuning in Data Integration.

Copyright © 2025, Oracle and/or its affiliates.





Copyright © 2025, Oracle and/or its affiliates.

4

Addressing Common Requirements

Addressing common requirements involves strategic planning and best practices to ensure optimal performance and user satisfaction.

Here are some key common requirements and the associated best practices.

Table 4-1 Addressing Common Requirements

Requirement	Design Best Practices
Understand your detailed business requirements before designing the application.	Getting Started with Your Design: • Gathering Design Information • Planning Your Application • Applying Your Design
Plan your application design: single or multiple applications based on your use case.	Decide on a Single Application or Multiple Applications
Choose the right application type based on your business requirements.	Cube Types Design
Plan for efficient model design. For example, optimize sparse and dense dimensions.	Dimension Design
Enable the Hybrid BSO application type, and consider the number of Hybrid BSO and ASO cubes.	• Hybrid BSO Design • ASO Design
Plan for security and access control, and address requirements for audit and compliance.	Security Design
Design your application for scalability and flexibility.	Application Design
Have a plan to manage metadata.	Design Checklist for Dimensions: Table 2
Include performance tuning options. For example: • Minimize complex dynamic calc members • Set the aggregation properties and data storage type • Optimize calculation scripts • Follow effective form design guidelines	Design Checklist topics, including: • Business Rules and Calculation Design, Table 4 • Forms and User Interface Design, Table 3
Design the appropriate approval workflow.	Approvals Design
Implement integration best practices.	Integration Design
Plan for disaster recovery and backup.	Design checklist for Maintenance: Table 9
Incorporate change management.	Creating and Running an EPM Center of Excellence
Create and run an EPM Center of Excellence (CoE).	Creating and Running an EPM Center of Excellence

Copyright © 2025, Oracle and/or its affiliates.

5
Use Case Examples

Designing best practices for common use cases can help organizations maximize the platform's capabilities and improve overall performance.

Review these topics to get use cases that help with your design:

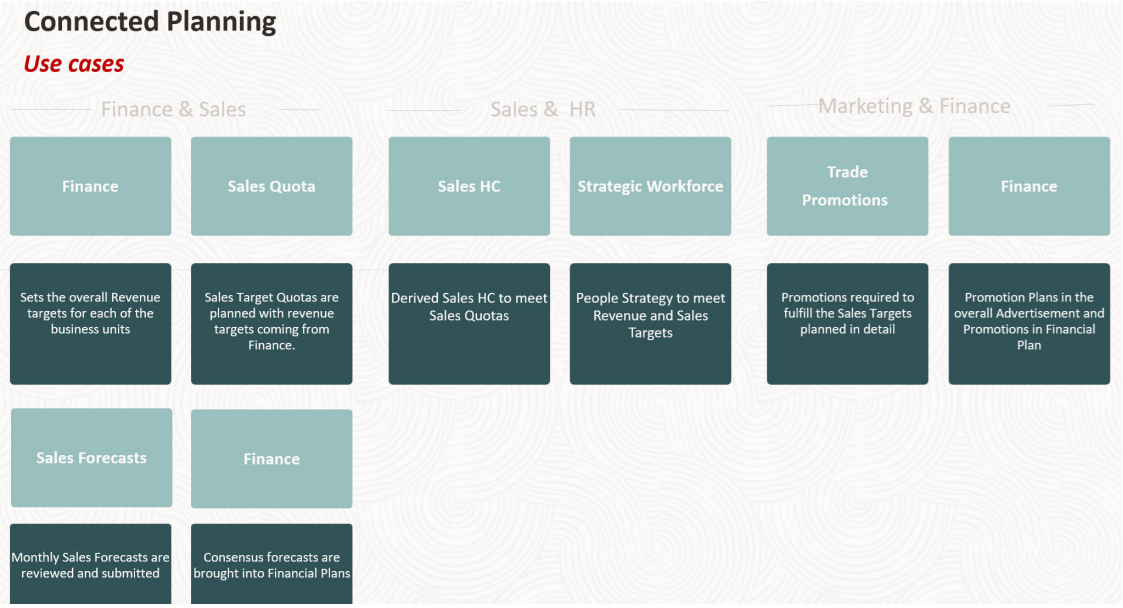
- Connected Planning
- Financial Planning and Operational Planning
- Industry Use Cases

Connected Planning

These are some common use cases for connecting planning with finance, sales, HR, and marketing.

- **Finance and Sales**:
  - Finance: Sets the overall revenue targets for each of the business units
  - Sales Quota: Sales target quotes are planned with revenue targets coming from finance
  - Sales Forecasts: Monthly sales forecasts are received and submitted
  - Finance: Consensus forecasts are brought into financial plans
- **Sales and HR**:
  - Sales HC: Derived sales headcount to meet sales quotas
  - Strategic Workforce: People strategy to meet revenue and sales targets
- **Marketing and Finance**:
  - Trade promotions: Promotions required to fulfill the sales targets planned in detail
  - Finance: Promotion plans in advertisement and promotions in the financial plan

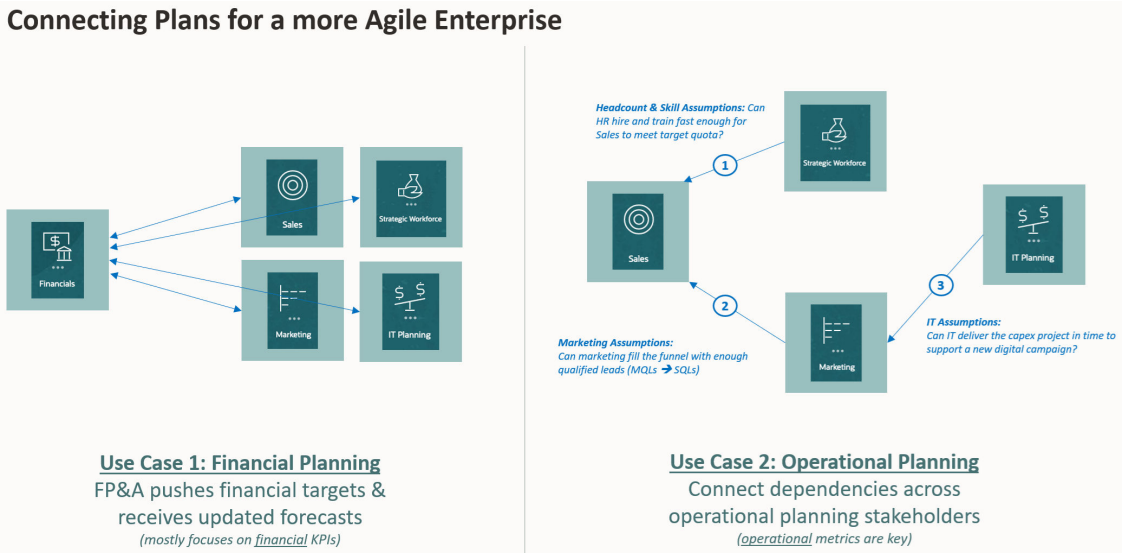
Copyright © 2025, Oracle and/or its affiliates.



Financial Planning and Operational Planning

Two typical use cases are financial planning and operational planning:

- For financial planning, FP&A can push financial targets to Sales, Strategic Workforce, Marketing, and IT Planning. It can then receive updated forecasts, focusing on financial KPIs.
- For operational planning, dependencies can be connected across stakeholders. In this case, operational metrics are key, such as asumptions for headcount and skills, marketing, and IT.



Industry Use Cases

Here are some industry-specific use cases and common business practices along with the associated best practices.

For additional use cases, review these Business Scenarios.

Copyright © 2025, Oracle and/or its affiliates.

Table 5-1 Industry Use Cases and Best Practices

Industry	Use Case and Best Practice
Retail	• Sales and Margin Planning–at the store, brand, division, or product level • Discount planning • Cost of sales planning • Fixed and rolling forecast, long range planning • Direct and indirect overhead planning • Staff cost, Capex, and project financial plans • Marketing, travel, and business trips plans • New stores plan and refit maintenance • Rent plan, shrinkage plan, and depreciation plan • Support division budget • Indirect OH budget • Driver-based OH allocations
Healthcare	• Ability to plan by entity (hospitals), service line (such as oncology or cardiology), or payer (insurance providers) • Patient volumes and global stats planning (inpatient or outpatient) • Patient revenue planning (volumes and average revenue rates per patient) • Allocations at the payer level • Contractual deductions and reimbursements • Other deductions (such as bad debts or charity) • Non patient revenue (such as pharmacy, gifts, bouquets, and parking revenue) • Bed capacity planning • Opex planning–fixed and variable cost (consumable items like medical supplies, lab supplies, oxygen and gases, and IV solutions.) • Opex–cost per unit of service for fixed and variable costs • Fixed and variable staff–FTE planning, cost planning • UOS, productive and non-productive hours planning • Benefits planning, FICA calculations • Hospital level P&L statements • Actual vs. Budget with comparative and variance analysis (Complete P&L) • Dashboards for both operational and financials
Manufacturing	• Production planning • Maintenance planning (spares consumption) • Sales operational planning at the product level • Sales forecasting–monthly or weekly • Quota planning and target setting by sales rep • Maintaining different prices for revenue models • Sales revenue and discount planning • Material future prices planning • Material consumption planning • Route level planning • Logistics and supply chain planning • Factory overheads planning • Overheads allocation planning • COGM and COGS Planning • Gross Margin Analysis • Detailed planning for OPEX accounts • Detailed marketing planning • Non-operating income and expenses planning • Budget control, transfers, adjustments, and write back

Copyright © 2025, Oracle and/or its affiliates.

Table 5-1 (Cont.) Industry Use Cases and Best Practices

Industry	Use Case and Best Practice
Banking	• Detailed mid-term projections based on trends and drivers • Rolling forecast planning based on trend and drivers • Detailed operational planning–loans and deposits • Planning at the OCR, product, and client segment level • Interest income and expenses planning • Non loan fee income planning • Upfront fee calculations and amortizations • PAR, early repayment, and loan penalty fee calculations • Other income and expenses planning • Capital projects • Existing assets planning • New assets planning and project assets planning • Gross value, net value, depreciation, and accumulated deprecation plan • IFRS 16–leased assets planning • Corporate office expenses planning • Headcount planning by department, position, and levels • Manpower cost planning–existing and new staff • Interest expenses on borrowings • Borrowed funds projection • New borrowing schedule • Commission and borrowing fee expenses • Treasury-related fee expenses • Various operational reports • Annual budget reporting pack and monthly financials • Rolling forecast • Balance sheet and cash flow budgeting
Airline	• Financial and strategic planning based on drivers, supporting top-down and bottom-up planning models • Revenue planning by flight number, route level, flight type, region, and business segment. Integration with airline revenue systems for the actual data. • Inflight sales planning, duty free sales planning • Opex planning at the GL account level. Detailed Opex planning for fuel cost, repairs, maintenance, training, and so on. • Budget financial statements–profit and loss, cash flow, balance sheet, annexures
Hospitality	• Detailed operational planning–room revenue, F&B revenue • Day-wise planning capability for the number of guests, room revenue • Events, weekend-based pricing, and automated revenue calculations • Predictive planning capability, demand planning • Automated ancillary services revenue, calculations based on drivers • Other income, expenses planning • Fixed and variable cost planning, direct and indirect cost planning • A&P planning at the initiative level • Actual vs. budget for A&P at the initiatives level • Development cost planning • Actual vs. budget tracking for development cost • USALI Reports–butterfly reports with MTD and YTD comparison

Copyright © 2025, Oracle and/or its affiliates.

Table 5-1 (Cont.) Industry Use Cases and Best Practices

Industry	Use Case and Best Practice
Telecom	• Operational planning–subscriber type, product, site, and division level planning • Plan for subscriber base, minutes of usage, average revenue per user, revenue by product, site, and subscriber type • Driver-based plan to calculate revenue for postpaid, prepaid, value-added services, handsets, internet, wireless, broadband, and digital services • Network operations and maintenance planning, depreciation planning • Capex–existing and new assets, leased assets, and IFRS 16 compliance • Capital projects planning–balance carry forward to next year • Marketing campaign planning • Approved budgets write back to ERP, budget transfers, and adjustments • What if scenarios and models
Higher Education	• Planning, budgeting, and forecasting at colleges, locations, departments, student category, faculty level, and study program • Integration with the student information system and financial system • Faculty planning, number of students planning, student accommodation planning • Monthly and weekly forecasting • Revenue and cost planning for student tuition fees, library fees, transportation fees, and book fees for new and existing students • Other revenue (non-operational revenue) planning • Budgetary control, transfers, adjustments and write back to ERP • Approval workflow to get the plan approved by rector and management
Insurance	• Insurance premium revenue–% growth (renewals and new) • Re-insurance planning–% on gross return premium • Loss ratio, net premium earned • Product and channel-level planning • IFRS 17 and IFRS 4 compliance for insurance planning • Commission planning • Investments planning • Integration with front end systems • IFRS 17 and IFRS 4 compliance reports

Copyright © 2025, Oracle and/or its affiliates.

Table 5-1 (Cont.) Industry Use Cases and Best Practices

Industry	Use Case and Best Practice
Real Estate Projects	• Real estate contractual projects planning–such as malls, shops, office, residential, commercial, complex, and warehouse • Plan for detailed project construction cost, development cost, labor cost, and material and equipment cost • Enter the % of completion by period so the revenue will get budgeted in the period of completion • Allocate project overheads to the project, such as direct and indirect overheads • If the project runs for multiple years, do cost adjustments during every budget cycle to adjust the overall project cost • Calculate the funds flow and cash flow for the project to see the nature of the funding, such as equity or loan • Enter the equity and debt % to calculate loan requirements to meet the project expenses • Budget for loan, interest, and principal amount repayments • Budget for loan amount utilization for the project year on year • Projects department budget for maintenance cost required to maintain properties year on year–this can also be considered to derive revenue budgeting • Revenue budgeting can be derived based on total operational cost, overheads, adjustments, borrowing costs, and profit margin • Total revenue divided by total rental area to derive the revenue per square meter • Rental area by tenant and revenue per square meter can provide the rental revenue by property and by customer for revenue budgeting • Provision to increase rent by a certain % considering inflation and maintenance costs, and accordingly derive revenue budgets for each property by tenant
Oil and Gas	• Sales and marketing planning–crude, petroleum, and gas • Region and product-level planning for oil and gas products • Sales volume and revenue planning for crude, petroleum, and gas • Marine planning and vessels planning • Oil and gas production planning at the refinery level • Allocation of oil and gas products to local government, local sales, and export sales • Product, region, and country-level profitability reports
Common Business Processes	• Long-term business planning • Rolling forecast and projections • Capex planning and leased assets plan • Projects and program planning • Financials–revenue and expenses planning • Automated income statement • Budget reports and dashboards • Cash flow and balance sheet planning • Driver and trend-based planning • Predictive planning • Actual vs. budget, actual vs. forecast comparative analysis • Approval workflows for each process • Budget transfers and adjustments • Scenario analysis

Copyright © 2025, Oracle and/or its affiliates.

6
Design Checklist

Here is a comprehensive design checklist for implementing and managing Oracle Fusion Cloud Enterprise Performance Management.

This design checklist provides information on planning, configuration, data management, reporting, and user training.

- **Applications**: Table 1
- **Dimensions**: Table 2
- **Forms and user interface**: Table 3
- **Business rules and calculations**: Table 4
- **Integrations and data movement**: Table 5
- **Reports and maintenance**: Table 6
- **Dashboards design**: Table 7
- **Navigation flow**: Table 8
- **Maintenance**: Table 9
- **Security**: Table 10
- **Planning Modules and other best practices**: Table 11

Copyright © 2025, Oracle and/or its affiliates.

Table 6-1 Applications Design Checklist

Task	Check Off After Completed
Decide if you need one application or multiple applications after considering the following scenarios: Single application scenario: • The planning process requires real-time seamless integration. • The same administrator can manage all the processes with no security concerns. • The planning process remains the same across regions or countries, and maintenance down time does not fall during business hours. • Performance and scalability can be handled within the same application. • The planning process can be implemented within the maximum number of cubes supported in an application. Multiple application scenario: • Customers with a diversified business that has multiple regions, legal units, and business processes where a single application cannot meet all the business requirements. Separate planning processes are required across different regions, for example, for revenue and workforce. • A separate administrator is required for each planning process. For example, workforce data should not be visible for the finance administrator. • There are time zone differences, regulatory issues, or application maintenance work downtime falls during normal business hours. • The number of cubes supported in an application is not sufficient to address all of the planning business processes. • There are requirements for performance and scalability of the application.	
While creating the application, enable the sandbox feature if you want to do simulations on the fly while working on planning. Enabling the sandbox feature creates separate sandboxes and a parent version to store all sandbox versions. This also allows users to create what-if versions on the fly while working on data forms. Selecting this option also provides the Enable Sandboxes option for Version members.	
While creating the application, enable the EPM Task manager feature. Task manager provides centralized monitoring of all tasks and provides a visible, automated, repeatable system of record for running a business process.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-1 (Cont.) Applications Design Checklist

Task	Check Off After Completed
Review the number of supported cubes and module cubes to determine what meets your needs. For more information, see Adding Cubes in *Administering Planning*. ** Total of 12 Hybrid BSO or ASO* *** Targeting 23.09* *+ Expanding to 5 per Product Roadmap* EPM Cloud Enterprise: Modules Based 3 / 4 / 5 / 2 = 14 EPM Cloud Enterprise: Custom 6 / 6 / 0 / 0 = 12 EPM Cloud Enterprise: Free Form* 12* / 12* / 0 / 0 = 12 EPM Cloud Enterprise: Sales Planning** 0 / 1 / 3 / 2 = 6 EPM Cloud Enterprise: SWP** 0 / 1 / 2 / 1 = 4 EPM Cloud Enterprise: Predictive Cash Forecasting (PCF) 0 / 1 / 2 / 1 = 4 EPM Cloud Standard: Modules 1 / 1 / 5 / 2 = 9 Legacy: EPBCS 3 / 4 / 5 / 2 = 14 Legacy: PBCS 3 / 4 / 0 / 0 = 7 NetSuite Planning: Standard 1 / 1 / 1 / 1 = 4 NetSuite Planning: Premium 3 / 1 / 1+ / 1 = 6	
Review this webinar on Customer Connect on application design for Planning. It provides detailed information on application deployment strategy, dimensions, and forms: EPM Application Design for EPM Planning Part I: App Deployment, Dimensions, and Forms.	
Review this webinar on Customer Connect for best practices to ensure high-performing ASO cubes in an application. It also covers best practices for data maps and Smart Push to synchronize data between BSO calculation cubes and ASO cubes: EPM Application Design for EPM Planning Part III: ASO, Datamaps, and Smart Push.	
Review this webinar on Customer Connect for best practices to ensure high performing applications: EPM Application Design for EPM Planning Part IV: Application Types and Migration from Legacy SKUs.	

Table 6-2 Dimensions Design Checklist

Task	Check Off After Completed
Identify which dimensions are required for each planning process, and map the process to standard (mandatory) dimensions. Most high-level and common financial planning use cases can be addressed with standard dimensions that can be renamed. Extend the dimensionality with custom dimensions which can be added and enabled or renamed per your business requirements.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-2 (Cont.) Dimensions Design Checklist

Task	Check Off After Completed
Identify the dimensions for planning and reporting. Some of the reporting dimensions can be handled using attribute dimensions if the relationship does not change over time. Attribute dimensions can be used in forms, business rules, reports per your reporting requirements. Examples of attribute dimension include product category, product size, and product UOM. These attributes can be associated with the product dimension. This helps in filtering products based on their categories, such as UOM and size in product-level planning reports.	
Combine two or more unrelated dimensions into a single dimension to avoid inter-dimensional irrelevance. You can have multiple hierarchies within the same dimension. For example, job, equipment, and material are unrelated in the projects cube, so they are created as a single dimension called Resource Class that has multiple hierarchies for job, equipment, and material.	
For seeded modules, enable the custom dimensions for appropriate business processes such as Revenue, Expenses, and Balance sheet according to the relevance of that dimension. This will help improve overall system performance.	
The ASO cube can have up to 32 dimensions. The ASO cube is mostly used for historical data reporting. The ASO cube also supports Groovy rules if calculations are required.	
Account and period should be defined as dense dimensions, and the remaining dimensions should be sparse. Leverage outline math or member formulas for simple calculations. Using outline math and member formulas can help achieve quick calculations without the need for business rules. Business rules and Groovy can be leveraged for more complex calculations.	
Parent levels and member formulas in the dense block should be set to Dynamic Calc or Label Only to avoid the requiring dense aggregations. Setting parent levels to dynamic calc enables automated aggregation to parent levels based on consolidation operators.	
Set consolidation operators to avoid creating unnecessary upper-level blocks. Do not use dynamic calc and store.	
Use alternate and shared hierarchies when members can be grouped under different parents for top-down allocation or reporting purposes. This can be leveraged in the case of top-down allocation requirements or multiple reporting requirements such as GAAP (for example, US GAAP and Local GAAP).	
Move archived versions into the ASO reporting cube to keep the size of BSO plan types down.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-2 (Cont.) Dimensions Design Checklist

Task	Check Off After Completed
Do not create too many custom dimensions or future dimensions just because they exist in ERP. Analyze the business requirement and determine if some of the custom dimensions can be merged together logically. Having the optimal number of dimensions can improve the overall application performance.	
For member formulas used to calculate ratios and other types of KPIs or percentages, set them to Dynamic Calc, Two Pass. The Two Pass setting properly calculates percentages at upper levels. For example, to calculate the ratio between Sales and Margin, the system must calculate Margin as a parent member based on its children, including Sales. To ensure that the ratio calculation is based on a newly calculated Margin figure, tag the Margin % ratio member as a two pass calculation. Essbase calculates the database and then recalculates the Margin % member. The second calculation produces the correct result.	
Use Label Only in sparse dimensions when reasonable. This will help avoid the creation of unnecessary data blocks.	
For best performance, the number of child members should not exceed 200 members for dynamic parents and 500 members for the store parent type. Avoid having flat hierarchies in any dimension. Setting intermediate parents helps achieve better performance by speeding up aggregation time.	
Enabling dimension members for multiple cubes creates dynamic X-Refs and leads to performance issues. It is recommended to add UDA HSP_NOLINK to all Account members to stop automatic @XREF generation across cubes where account members are the same. Use data maps or Smart Push to move data across cubes.	
Note which dimensions are aggregating and which are non-aggregating to ensure proper member properties and unary operators are assigned to each dimension.	
Use Dynamic Calc in sparse dimensions where the impact on query performance and data exports is minimal to eliminate unnecessary data blocks. In general, use store or never share for sparse dimensions and aggregate using business rules when required. Setting all to store improves the size of the application for the page and index file, thus improving overall performance.	
Where reasonable, use Unary Operator ~ or ^ to eliminate unnecessary aggregated data blocks. This will help minimize the data block size.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-2 (Cont.) Dimensions Design Checklist

Task	Check Off After Completed
Avoid single child parent members to reduce block size or duplicate blocks. A single child parent member leads to implied shares or duplicate blocks and data if the parent member is made never share.	
Do not have too many parents with a single child. Extra layers are not necessary, and this will impact reporting.	
It is a best practice to store historical data in ASO cubes. Have only up to 1 to 2 years of data in the BSO cube for trend-based planning, and load the remaining years into an ASO cube for historical data reporting. Consider using Predictive Planning that can use an ASO cube as a source of data. Aggregate large dimensions in ASO instead of BSO whenever possible.	
Do not create QTD, YTD members in the period dimension. Use the Dynamic time series capability in the period dimension to report periodic QTD, HTD, and YTD data. Summary time periods such as quarter totals and a year total should be set to dynamic calculate to reduce calculation time.	
The dimension order should follow either hour-glass or modified hour-glass order. Hour-glass order is best used in cases where most calculations are performed in batch and contain full sparse dimension aggregations. Modified hour-glass order (non-aggregating sparse dimensions are placed after the aggregating dimensions) is best used in cases where most calculations are performed on save in web forms with only partial aggregations. Dimensions should be in the proper order, starting from most blocks to least blocks order. This is fastest and produces the fewest empty blocks while doing aggregations.	
Make use of Attributes and UDAs for capturing additional information related to the dimension. For example, Product size, ProductType, and Product UOM can be captured as attributes instead of having additional dimensions in the system. This is useful when relationships do not change over a period of time.	
The Data Type Evaluation Order setting is predefined for out-of-the-box cubes and cannot be changed. For custom cubes, it is recommended to set the proper evaluation order. Data types can have conflicting results on the face of a data form, depending upon the cell intersections defined for the data form. For example, the intersection of a percent data type and a currency data type need to be resolved based upon a defined order set by a Power User.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-2 (Cont.) Dimensions Design Checklist

Task	Check Off After Completed
Aggregate only those dimension members that are not dynamic calc.	
Automate metadata integration using the import metadata process, and scheduled it to run automatically using EPM Automate or REST APIs. Metadata extraction from ERP can also be automated using a BIP report. This not only helps automate metadata integration, it also helps avoid data integration issues and errors while importing data.	
Smart Lists and multi-cube BSO dimensions are useful when the planning dimension is not a primary dimension for a process and there is a need to break it into sub processes.	
Text, Smart List, Date, and Stored percentage type accounts should be set to Never or Ignore for consolidation. Aggregating these values, unless the accounts are explicitly excluded in any agg script, creates useless data. Configure the dimension model to address this for better application performance.	
Dynamic member formulas should have conditional logic and meaningful formulas for optimum system performance. Member formulas should be written to leverage NOT conditions to avoid returning #missing values.	
Follow the rules for naming conventions for applications, dimensions, members, and aliases. For example, the name should be within 80 characters and should not use special characters. Review the complete list of guidelines: Naming Restrictions in *Administering Planning*.	
For very large applications with a large number of reporting dimensions and instantaneous aggregation or reporting requirements, you can leverage Hybrid BSO-ASO with Smart Push and Aggregate Views in ASO.	
Dimensions should be grouped together into appropriate cubes. Avoid having many intersections set to the No Member.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-3 Forms and User Interface Design Checklist

Task	Check Off After Completed
Avoid long-running rules on Form Save since that has an impact on performance. Instead, use an action menu for long-running rules so that users can trigger the rule only when required. This will provide a better user experience while improving performance for forms. Guidelines: • On Save should only include limited rules that run very quickly. • Rules with dimension aggregation should be run explicitly. • On Load should be very limited.	
If business rules are configured for the form, execution time of these rules needs to be checked since this directly affects the loading and saving time of the form.	
To reduce maintenance, use member functions instead of picking individual members. This will ease form administration in the long run.	
Control the number of dimensions on the row. Having multiple dimensions on the row can negatively impact performance.	
Put dense dimensions such as Account and Period on the row and column of a form. Put sparse dimensions such as Entity on the Page axis or POV. This will help achieve better performance.	
It is recommended to use the Suppress Missing Blocks setting to improve performance. Consider either reducing the number of accounts or adding a form accessed by a right-click menu option to add new account members. Turn on suppress missing blocks whenever you have sparse dimensions on rows. Suppress missing blocks and data should be used on read-only forms	
Run the Web Form Diagnostics Tool on the environment with a full set of data. The diagnostics tool uses the last used run-time-prompt (RTP) and user variables, so this should be done by the implementer. This will help guide form design if a given form does not have acceptable performance.	
Use the outline to add member formulas instead of using formula columns in the form.	
Do not use large forms for ad hoc grids. Instead, break them up into smaller forms.	
Do not have too many hidden columns or rows in the form definition as this can lead to performance issues. Instead of formula columns, the best practice is to use outline member formulas.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-3 (Cont.) Forms and User Interface Design Checklist

Task	Check Off After Completed
If Smart Push is enabled on forms, follow these best practices for Smart Push: • Only push level 0 cells that can potentially be changed by an end user. The size of the pushed data should be small. • One or more data maps can be associated to a form. Use the form context by allowing the system to pick page, POV, rows, or the column selections union of members from a multi-segmented form. • Use database suppression to compress missing data while extracting from the source. This provides better performance in Hybrid. • Enable *Run smartpush in the background* to release the UI thread immediately to the user. • Run on save runs immediately upon form save, and the user can also invoke Smart Push from an action menu in the form. • Automatic push should be limited because users tend to save data in a web form before finishing their data entry. Do not over-use Smart Push. • Large data movement using Smart Push should be limited to only a few concurrent users. Limit large slices of data movement using data maps and data import during user activity that uses Smart Push to move data. • Create summary level 0 forms in the BSO cube so that users carry out one Smart Push operation on all data after all in-progress edit and save operations are completed • Smart Push is intended for moving small amounts of data from BSO to ASO cubes. It is not intended as a method of moving all or large amounts of data. • If Smart Push is used, data push should only be done in batch to avoid contention issues with Smart Push processes. • Use Groovy to set overwrite selection to only modified data for grids with sparse members on row.	
Use dynamic user variables, substitution variables, and valid combinations instead of creating too many forms.	
Dimensions such as Scenario or Version and Year can reside on the POV, column, or row. It is important to properly gauge how columns or rows will be returned when a user opens the form.	
Do not have too many rows on a single form. Use relationships to incorporate members onto the forms instead of picking members individually. This helps improve maintenance.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-3 (Cont.) Forms and User Interface Design Checklist

Task	Check Off After Completed
While building forms, ensure that you select all appropriate options to enhance the design of your form. For example, use settings to control precision, display, and menus, and to associate the proper rules with your form.	
Use substitution variables to reference dimensions such as years, period, scenario, version,and currency. This reduces maintenance of forms.	
Select appropriate options in settings: • Use settings to control precision, display, and menus, and to associate rules with your form. • Select the Suppress Scenario/Time Period option. • Set Periods in the row or column on the form to the Start and End Period set for the Scenario.	
Assess form performance: • Forms should open in under 5 seconds. • Forms should save in under 30 seconds. If improvements are needed, implement the guidelines in this checklist.	
Consider using user variables for dimensions such as Entity and Scenario to help reduce the dimension selection for end users.	
Organize your forms in logical folders. Separate input-type forms from reporting forms. This improves form administration, and it also helps users access forms from Smart View.	
Consider using dashboard-based forms wherever possible so that users can input their plan numbers and see the results immediately.	
Business rules associated with a form can be set to launch automatically when the form is opened or saved. You can select Use Members on Form to populate runtime prompts from the current form instead of prompting users for input when rules are launched.	
Set the total number of rows in a form to be less than 100 to provide better performance and user experience.	
Forms are used by multiple users, so rules set to run on save should not contain CALCPARALLEL or FIXPARALLEL.	
Data should not be submitted through ad hoc forms because ad hoc entries can potentially override data validations or calculations.	
Data validation rules in forms help to implement business policies and practices. You can specify cell colors and data validation messages that are generated on the form if entered data violates a validation rule.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-4 Business Rules and Calculations Design Checklist

Task	Check Off After Completed
Rule-related issues are the cause of many escalations. The impact of poorly written rules includes: • Increased size of the database through the number of page and index files • Decreased performance of calculations • Increased maintenance time	
Avoid having many rules that run on form save. This can lead to slower form performance.	
Rules should not have missing dimensions in fix or member block statements. Errors and warning should be run on all rules to ensure dimensions do not go outside fixes.	
SET Commands: • Do not use SETCREATEBLOCKONEQ ON and SET CREATENONMISSINGBLK ON at the top of the rule. • Do not use administrative type commands like SET CLEARBLOCK EMPTY in end user rules as it requires a restructure. • Avoid or test rules using SET CALCTASKDIMS. Essbase does this automatically, so it is best to let Essbase to decide. • Data copy rules should include SET COPYMISSINGBLOCK OFF to copying empty blocks within the Fix.	
Block creation should be done using either DATACOPY or sparse member assignment. Functions (@CREATEBLOCK ON EQ and @CREATEBLOCK) can be used as a last resort inside a limited Fix statement.	
Rules should have conditional logic to prevent constant values from being created where formulas have more than one member. Conditional logic will prevent the creation of zeros or other nonsensical values.	
Missing dimension references in Fix (such as a data copy rule) can waste processing time, increase contention, and create unnecessary blocks for all levels of the missing dimensions.	
Using @ancestors in a fix statement to aggregate other dimensions is not a best practice. This is a slow way to aggregate and creates a lot of blocks.	
Remove parallel calc in business rules associated to forms. CALC Parallel or FIX Parallel should be used only with administrative or batch rules.	
Creating unnecessary zeros will lead to increased blocks and data. Carefully review business logic and add necessary IF conditions to check for zeros. Convert zeros to #MISSING.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-4 (Cont.) Business Rules and Calculations Design Checklist

Task	Check Off After Completed
Eliminate multiple passes on the same blocks. Instead, set an outer fix and move in and out as necessary. Combine IF statements instead of doing several IFs on the same intersections.	
Avoid using cross-dimensional references on the left side of the equation, as this has a performance impact.	
Aggregate dimensions in order of creation of most blocks to least blocks in the scripts. Agg is faster than CalcDim and is a preferred mechanism of aggregation. Aggregation using @ANCESTORS in end user rules all the way to the top of the dimension can lead to block contention issues.	
Use RTPs instead of using multiple rules with the same logic. More rules means more maintenance. Changes to dimension properties, incorporating conditional logic, removing set createblockeq, and performing calculations at the proper level instead of every level will improve the lev0 to upper level block ratio and improve overall performance.	
Use templates for breaking down and reusing business logic. However templates should not be used for fully functioning rules with FIX, ENDFIX. The rule combining various templates should have a proper outer fix and move in and out of smaller pieces as necessary.	
Ensure that each end user business rule is only calculating unique blocks. No two users should calculate the same blocks concurrently to avoid block contention.	
End users should not be fully aggregating sparse dimensions.	
When the point of view is set, the rule will run only for the members chosen. Using a runtime prompt for the dimensions will allow users to specify member values for these dimensions when launching the rule. This way, users can launch the rule several times for different years, scenarios, and versions without having to modify the rule in Calculation Manager.	
Use of CALCPARALLEL and FIXPARALLEL for concurrent processing in business rules by many users may result in slow performance because such rules consume more resources compared to business rules that are processed serially.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-4 (Cont.) Business Rules and Calculations Design Checklist

Task	Check Off After Completed
A single invocation of a business rule that uses CALCPARALLEL and FIXPARALLEL may perform acceptably. However, overall performance will deteriorate if multiple concurrent users execute such business rules. As more users concurrently run calculations with CALCPARALLEL and FIXPARALLEL, resource usage increases and may reach capacity thereby decreasing the overall performance. Do not use CALCPARALLEL and FIXPARALLEL for business rules run by end users. Also, do not use CALCPARALLEL and FIXPARALLEL for business rules run in batch that are run concurrently with end user business rules.	
Review which rules are end user and batch calc, and identify the frequency of admin batch calc runs, such as nightly or every 2 hours during end user processing.	
Identify how each end user rule is scoped to avoid block contention with concurrency.	
Utilize Calculation Manager diagnostics and log messages to optimize all rules. This provides information on: • The number of passes through the database • Warnings • The number of blocks and if dimensions are missing • Rules that need to be optimized	
Use Groovy for some of the calculations in custom cubes. Groovy can be launched independently, using action menus, run on load or save of form, or through EPM Automate and REST APIs. Here are some use cases where Groovy can be used: • Calculate only on edited or changed cells or rows • Data validation on forms before save • IF ELSE before FIXes if some criteria is met • Change cell color based on data validation • Move data to reporting cube only for edited cells • Move data across cubes (also using data maps) • Make selected rows or columns read only based on conditions • Increase performance • Automate EPM Automate tasks like backups and import files • Integrate strategic models using Groovy rules Learn more about the Groovy object model, syntax, java classes, and packages: Overview (Oracle Enterprise Performance Management Cloud, Groovy Rules)	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-4 (Cont.) Business Rules and Calculations Design Checklist

Task	Check Off After Completed
Rules for currency conversion are not needed as this functionality is built in. Just enable the currency while creating the application, and the system automatically creates an exchange rates form and seeds the rule for currency conversion.	
Leverage runtime prompts for members such as Entity, Scenario, and Version. This allows your rule to be dynamic and run based on user input.	
Typically, dense dimensions such as Account and Period don't need to be aggregated because parent members are set to dynamic calc. However, if you have member formulas on dense dimensions and they are not set to dynamic calc, a Calc Dim rule is required.	
After you fully understand the drivers of the calculation, use the proper object component or template to build the rule. The components and templates facilitate member selection to help deploy the rules.	
As a first step in building rules, ensure that you understand the business logic and which entities or departments the rule applies to. For example, know the source and target accounts that are involved in the rule.	
Poorly written rules have a major impact on the application, including: • The size of the database - the number of page and index files • Performance of calculations can cause block contention issues, which impacts end user rule performance • Maintenance time increases and impacts rule performance and maintenance time. For optimal application performance, it is strongly recommended to follow best practices while writing business rules.	
Avoid costly functions such as @CurrMbr, for example: `->@MEMBER(@PREVSIBLING(@CURRMBR( "Years" )))` Use @Prior instead.	
Nest Fix statements where possible to reduce the number of passes on the database. Each full FIX requires a pass on the database. For example, use an outer FIX for Version, Scenario, and any other dimension selections that are static throughout the business rule.	
Run Errors and Warnings in Calculation Manager to see the number of passes, missing dimensions, and issues with the rule.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-4 (Cont.) Business Rules and Calculations Design Checklist

Task	Check Off After Completed
Schedule frequently used rules as jobs to run automatically on a periodic basis. This could be, for example, a Rollup income statement rule or another aggregation rule that reflects all planning inputs in financial statements. Even data or metadata integrations can be scheduled as jobs as required.	
General best practices: • All Essbase functions and commands should be in capital letters, for example, FIX, ENDFIX, IF, ENDIF, and SET. • Dimension members should always be used within double-quotes, for example, "Oct". • Obsolete code should be deleted from within scripts and not simply commented-out • Any changes to the code post go-live should be commented-out with the corresponding change request ID to enable traceability.	
Be sure that you don't create too many 0 and 1 values in the application with your rules. If conditional logic is implemented in the rules, it will prevent 0s from being copied or aggregated.	
Review this webinar on Customer Connect to understand best practices for writing calcscripts and common mistakes to avoid when creating rules. It also covers design of Hybrid BSO cubes, which helps optimize performance: Application Design for EPM Planning Part II: Calculation Manager Rules and Hybrid BSO.	

Table 6-5 Integrations and Data Movement Design Checklist

Task	Check Off After Completed
Use data maps for cube-to-cube integration instead of XRef or XWrite commands. Using XWRITE with a large number of users and increased data can lead to performance issues. Do not use XREF and XWRITE across members in different cubes.	
Data load for both YTD and Periodic is supported. However, loading YTD data is overhead as YTD data will be dynamically calculated.	
Do not execute business rules, reports or scheduled tasks while data is being loaded.	
While integrating actual data from source systems, eliminate rows with 0 or null values for optimal performance.	
Ensure that no two users execute data load to the same intersections.	
For file based integrations, use EPM Automate and task scheduler to automate the whole integration process on a periodic basis.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-5 (Cont.) Integrations and Data Movement Design Checklist

Task	Check Off After Completed
The EPM Integration agent can be leveraged to establish a seamless connection with the SAP Data source and extract the data into Cloud EPM. Seamless integration is supported as long as the source DB supports type 3 or type 4 compliant JDBC URLs.	
If the source database does not support type 3 or type 4 JDBC URLs, file based data integration can be automated using EPM automate and task scheduler. Alternatively, it can be loaded to the database using REST APIs and then using the integration agent to connect with the database.	
If there are multiple applications where metadata needs to be maintained, you can use Oracle Fusion Cloud Enterprise Data Management for centralized master data management. It provides built-in workflows, and different nodes can be created and maintained per business requirements.	
EPM Automate can be run directly on the cloud using Groovy rules. Routine activities like daily backups and integrations can be executed using EPM Automate commands using Groovy scripts. Review this presentation on Customer Connect for more details: What's New in EPM Automate.	
Review this presentation on Customer Connect for in-depth information on inbound and outbound integrations in Cloud EPM, including the EPM integration agent: Cloud EPM Integration.	
The end-to-end integration process can be automated and orchestrated using the pipeline feature. The pipeline functionality allows you to conduct a sequence of operations such as Load Data or Metadata, Clear Data, Aggregation, Batches, Business Rules, and so on. This allows users to plan all of their jobs in one location and perform them in parallel or serial mode. Users save time by running each job one after the other. Pipeline also assists in remotely running jobs from another instance via connections. Review this presentation on Customer Connect: Data Exchange Pipeline Feature.	

Table 6-6 Reports and Maintenance Design Checklist

Tasks	Check Off When Completed
Review the design guidelines in Reports Design.	
Use Reports features, including charting, report viewing, books, and bursting.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-6 (Cont.) Reports and Maintenance Design Checklist

Tasks	Check Off When Completed
Determine the frequency and distribution of reporting. For example, for frequency, if budgets are annual, are reports run annually or on another frequency? For distribution, will users run reports and books on the web or in Smart View? Will users need to receive report output in email through bursting?	
Before building reports, determine how many different report formats are required.	
Begin building reports by arranging dimensions properly and selecting members. Next, get the report to capture the data and apply formatting. FInally, utilize other Report features, such as conditional formatting and text, zoom, drill to content and drill through, and grouping.	
Schedule Restructure Cube to be run every weekend during off times. This will help clear empty blocks in the application.	
Define regular maintenance processes for ASO cubes: • Merge data slices and remove zeros. • Compact the outline. • Review the dimension hierarchy types. • Build the optimal number of aggregate views.	

Table 6-7 Dashboards Design Checklist

Task	Check Off When Completed
Use dashboards at the start of the planning process to provide users with an overview of their process by showing summary data.	
Use interactive dashboards so that changes are reflected as soon as updates are made. For example, users can change a driver in the form and immediately see that reflecting in charts and other summary forms.	
Use dashboards 2.0 to take advantage of spark charts, geo maps, bubble charts, waterfall charts, gauge charts, and so on. Use appropriate chart types based on the available data to provide the best visualizations to users.	
Use a global POV bar to display selections at the top of the dashboard. After the user selects a member from the global POV bar, all the forms and charts will be updated based on the selection. For example, if a user selects the entity Vision, all the charts and forms are updated based on this filter.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-7 (Cont.) Dashboards Design Checklist

Task	Check Off When Completed
Enable summary-level dashboards with the ability to drill down into further details based on dimension hierarchies. Include explanations and comments on data and charts by inserting a commentary section.	
Use logarithmic scales, hierarchical labels, and a secondary Y axis in relevant charts and create interactive dashboards to enhance the user experience.	

Table 6-8 Navigation Flow Design Checklist

Task	Check Off When Completed
Use navigation flows to provide an easy way for users to complete their planning and reporting processes.	
Customize the navigation flow based on user personas, on business processes, or a mix of both. This enables designers to control how roles or groups interact with the business process.	
Create clusters and cards, and arrange them horizontally or vertically with icons based on business processes. Customize them to provide a helpful user experience.	
You can hide any clusters and cards that are not required, and arrange them in the order of user preference per business requirements. For example, you could move business process cards up and move down applications, tools, or academy.	
Categorize navigation flows and apply access permissions as required by the business.	
If a user has access to multiple navigation flows, they can switch navigation flows from the home page based on the task they need to perform.	

Table 6-9 Maintenance Design Checklist

Task	Check Off When Completed
Schedule Restructure Cube to be run every weekend during off times. This will help clear empty blocks in the application.	
Regular maintenance processes should be defined for ASO cubes, including: • Merge data slices and remove zeros. • Compact the outline. • Review the dimension hierarchy types. • Build the optimal number of aggregate views. See Optimizing Aggregate Storage Option Cubes.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-9 (Cont.) Maintenance Design Checklist

Task	Check Off When Completed
Schedule automated daily backup of all the artifacts and have a disaster recovery plan for test and production environments. Daily backups can be easily automated using EPM Automate or REST APIs and organized in separate folders. See Backing Up and Restoring an Environment Using the Maintenance Snapshot.	

Table 6-10 Security Design Checklist

Task	Check Off When Completed
Understand access permissions, including user and role management, application artifacts that can be assigned permissions, types of access permissions, and managing and reporting on access permissions. See Setting Up Access Permissions in *Administering Planning*.	
Learn about managing users and configuring security settings in *Administering Access Control for Oracle Enterprise Performance Management Cloud*: • Managing Users and Roles • Configuring Security Settings	
Get an understanding of access control, managing role assignments, and generating reports in *Administering Access Control for Oracle Enterprise Performance Management Cloud*: • Overview of Access Control • Managing Role Assignments at the Application Level • Generating Reports	
Watch the webinars on Customer Connect: • Approval Groups and Phased Approvals • Cell Level Security for Cloud EPM Applications.	
Complete the tutorials: • Setting Up Security in Planning • Setting Up Security in Cloud EPM	

Table 6-11 Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
If you want to choose your own chart of accounts, do not enable accounts under Financials.	
If not necessary, do not enable custom and future dimensions.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Leverage standard business processes such as Financials, Workforce, Projects, Capex and Strategic Modeling, and build configurations on top of that if there are custom requirements. Watch this webinar on Customer Connect to get an overview: Planning Modules Refresher: Financials, Capital, and Projects	
Modules provide significant out-of-the-box content with hooks for customization. Using standard modules helps with maintenance of your application. Modules also track customizations to out-of-the-box-content, and custom changes are retained during upgrade. You also have the ability to restore the modified content to the original.	
Customer dimension members should be seeded under the parent members provided with the module. Forms automatically incorporate customer dimension members that are embedded within the seeded parent members.	
Review these Customer Connect webinars to learn about the business processes: • Financials • Capital • Projects	
The Predictive Cash Forecasting solution enables operational cash flow planning for short- and medium-term time horizons. It covers several forecast methods such as smart drivers, predictive planning, driver methods, and trend-based planning methods. Companies can forecast for cash inflows and outflows accurately with data driven approach to ensure that they have enough liquidity to meet its financial obligations and strategic goals. Review this Customer Connect webinar to learn more: Announcing Predictive Cash Forecasting in Cloud EPM.	
Use valid intersections to filter dimensional intersections for users when they enter data or select runtime prompts. Valid intersections restrict data input, but not data loads or calculated results. They must contain a dimension unique to the database to restrict the intersections for that database. Otherwise, the valid intersection applies to all databases that share the intersections. Planning Modules include some predefined valid intersections.	
Use Groovy scripts to improve the overall performance as it can run only on edited cells in a particular form. Groovy can also be leveraged for data validations in forms.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Schedule Restructure Cube to be run every weekend during off times. This will help clear empty blocks in the application.	
Predictive Planning and Auto Predict functionality can be used for generating what-if scenarios based on actual data and seasonality patterns. System prediction can be used to compare drive- based planning results. Alternatively, prediction scenarios can also be copied to a plan scenario. Review this Customer Connect webinar to learn about auto predict functionality: Reimagine Your Planning and Forecasting Process Using Auto Predict.	
The Financials business process includes standard functionalities such as drivers and trend based planning, rolling forecast, balance sheet and cash flow planning, integrated financial summary, weekly planning, P13 support, budget adjustments, and write back. To save time and effort, it is recommended that you use these standard business processes in the Financials framework instead of customizing. Review this Customer Connect webinar to learn more about integrated Financial Statement Planning: Integrated Financial Statement Planning using Cloud EPM Planning Modules.	
If you want to implement IFRS16 and leased assets planning functionality, it is available out-of-the-box in the Capital business process. You can plan for new assets, existing assets, or leased assets, and the financial impact is integrated seamlessly with the ability to drill through back to the source details.	
For capital, indirect projects, and contractual projects, Projects addresses project requirements and has seamless integration with Oracle Project Portfolio Management. Projects also integrates with Capital, Workforce, and Financials. It is recommended to leverage these standard capabilities rather than creating customizations.	
Workforce provides resource planning, including headcount, manpower cost, and integration with Oracle Cloud HCM and Payroll. You can leverage workforce planning functionalities for your manpower planning requirements.	
Strategic Planning provides long term business planning, scenario modeling, model consolidation, predictive modeling, and integrated financial statements. Strategic planning also integrates seamlessly with BSO cubes to bring in data required for long term planning. Leverage Strategic Modeling for any long-term planning requirements.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Use pre-built templates available out-of-the-box for scenario modeling capabilities, model consolidations, and strategic modeling. You can build long-term integrated business plans with the capabilities to goal seek, review the audit trail, and do mergers acquisitions planning, debt planning, investment planning, and predictive analysis. This helps management quickly model strategic plans with free form and standard calculations without having to write technical scripts or rules for calculations.	
Best practice solutions are available for ITFM (IT Financial Management) and Marketing campaign planning. You can raise an SR to leverage and benefit from these best practice solutions. Learn more: Overview: IT Financial Management.	
Rules such as aggregation, copy, and clear data are available out-of-the-box. It is recommended to use the provided rules.	
Enable the audit trail feature to track all changes in the application for data, metadata, and artifact. This helps track who changed what and when the updates were made.	
It is recommended to leverage the standard KPIs, metrics, and dashboards that are available in each business process. They enable top management to get quick insights from dashboards to inform key decisions.	
Dashboards 2.0 can be leveraged to generate advanced charts like geo maps, spark charts, bubble charts, waterfall charts, and gauges. Make interactive dashboards to provide an improved user experience for day-to-day use and planning as well as integrated views. For example, when you update a form, charts dynamically refresh to provide an interactive user experience while planning.	
Leverage Free Form to migrate custom Essbase cubes to the cloud. This gives you the ability to create up to 12 custom cubes, with a mix of hybrid BSO and ASO cubes. You can use any of the 29 dimensions in any combination for each cube based on your use cases. Watch this Customer Connect webinar to learn more: Announcing Multi-Cube Free Form Applications.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
It is recommended to review the activity report diagnostics that can highlight if there are any issues with form design or business rules that take a longer time to run. You can take correction action to rectify the forms and business rules based on this report. The activity report also provides information regarding: • User information, such as who signed in to the application by day or time for the last 30 days • The percentage of requests that took more than 10 seconds to complete • CPU and memory usage • The number of application design changes • A ranking of requests by duration In addition, use Calculation Manager diagnostics and log messages to optimize all rules. Watch this video to learn more about the activity report.	
Sales Planning can be leveraged if you have requirements to manage operational quota planning, target setting for sales reps, promo or non promo planning, or advance sales forecasting. Sales Planning includes features such as top-down and bottom-up planning processes, key KPIs, sales compensation planning, and ramp-up profiling. It can be leveraged and configured as per your specific business requirements.	
Use flex forms to add rows on an ad hoc basis while planning. Flex forms also provide the option to sort, filter data, and add and move rows.	
Ad hoc grids allow planners to slice and dice data and do everything that they currently do in Smart View. This functionality can be leveraged to prepare ad hoc analytical reports with slice and dice capability.	
For historical actual data, use an ASO cube. You can have 1-2 years of historical data in a BSO cube, but if there are more years, it is best to move the historical data to an ASO cube for better system performance.	
It is not recommended to do a Smart View bulk update. The best practice is to use the application by giving secured access to all users.	
Use the sandbox feature to do simulations on the fly. If users are satisfied with the sandboxing results, they can publish the data in the sandbox.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Strategic Workforce Planning can be leveraged if you want to plan for long-term workforce planning, supply vs. demand analysis, or skills analysis. Strategic Workforce Planning has seamless integration with Workforce, which addresses compensation planning.	
You can start with minimal configurations for out-of-the-box modules, and then incrementally enable each section as per business requirements.	
If there are multiple applications in use, consider setting up connections. This provides a seamless experience for users to access all of the functionality in same application.	
Use data maps and smart push to move data across applications instead of using XRef or XWrite functions.	
Using the standard best practice framework for optimal block size and best practice design for dimensions, forms, and business rules. It will also ensure optimal performance of the application.	
For ASO cubes, run Merge Slices to address large data slices. This can be scheduled to run on a periodic basis. Run Compact Outline if you have large metadata changes. This can also be run as a scheduled job.	
Be sure to check the activity report on a regular basis to identify and fix any calculation scripts and user requests that impact the overall performance of the application.	
Use navigation flows rather than task lists for planning processes, management reports, and dashboards to facilitate ease of access for users.	
Use context-sensitive right click menus to provide drilldown capabilities from forms. For example, users can right-click an account and drill down to open another form to see cost center balances and compare actual, plan, and forecast.	
Use Smart Forms in Smart View to for custom business calculations using Excel features and functions. The business calculations that you create and save in the Smart Form can then be executed in both Smart View and the provider web interface. In Smart View, the formulas are evaluated by Excel; in the web interface, the formulas are evaluated by the provider. You can even create a sandbox from a Smart Form.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Flex forms provide flexible row management in Smart View. Smart View users can rearrange row members and sort or move rows. Modified row order is maintained on refresh and during submits. Smart View users can also filter data using the Excel filtering functionality. Flex forms are used only in Smart View and not in the web interface.	
Use the reports functionality to enable a more dynamic user experience by providing interactive reporting functionality and improved reporting capabilities with charts, drilldown capability, and administrative design.	
Leverage the IPM Insights functionality to generate analytical insights for the management team. IMP Insights provides these types of insights, using historical and predicted data: • Forecast variance and bias insights - reveals hidden bias in forecasts submitted by planners • Prediction insights - helps you uncover significant deviations in forecasts compared to predicted values • Anomalies insights - detect unusual patterns in data that deviate from expected results For more information, see About IPM Insights.	
It is recommended to follow proper naming conventions for forms and business rules to quickly identify the key characteristics. This also helps ease administration. For example, rule names can follow a convention like this: *R-CubeName-RuleNumber-RuleName*, using *R* for rules and *RS* for rule sets.	
Use Compact Outline regularly for ASO cubes. As dimension members are updated and moved, compacting the outline is required, similar to how a restructure is required in BSO.	
Use the budget revisions feature to create revisions in an approved budget, such as to process midyear changes to the budget. This feature integrates with budgetary control in Oracle ERP Cloud. Watch this Customer Connect webinar to understand the budget revision feature: Overview of New Budget Revision Feature in Cloud EPM Planning Modules.	
Use the Quick Start checklists to do initial tasks on your first day such as setting up your environment, log in, enter data, analyze forms, use reports, etc. This checklist is also organized based on roles like service administrator, power users, users and viewers. Review the Quick Start checklists.	

Copyright © 2025, Oracle and/or its affiliates.

Table 6-11 (Cont.) Planning Modules and Other Best Practice Recommendations Design Checklist

Task	Check Off When Completed
Review this Customer Connection webinar for best practices on how to wow your users and make your EPM solution shine.	

Copyright © 2025, Oracle and/or its affiliates.