

Oracle® E-Business Suite

Oracle Enterprise Command Centers の拡張
リリース 12.2
部品番号 F24028-01

2019 年 8 月

Oracle E-Business Suite Oracle Enterprise Command Centers の拡張 リリース 12.2

部品番号 F24028-01

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

原著者: Nashwa Ghazaly、Muhannad Obeidat

協力者: Mildred Wang

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency- specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIX は、The Open Group の登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様と Oracle Corporation との間の契約に別段の定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様と Oracle Corporation との間の契約に定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

サポートをご契約のお客様には、My Oracle Support を通じて電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

ご意見をお寄せください	vii
はじめに	ix
1 概要	1-1
Oracle Enterprise Command Center Framework の概要	1-1
2 Oracle Enterprise Command Center Framework のアーキテクチャ	2-1
Oracle Enterprise Command Center Framework のアーキテクチャの概要	2-1
3 データ・ロード・プロセス	3-1
データ・ロード・プロセスの概要	3-1
データ・ロード・プロセス定義	3-2
4 セキュリティ	4-1
Oracle Enterprise Command Center Framework のセキュリティの概要	4-1
Oracle Enterprise Command Center Framework の認証	4-2
ページ・レベルのセキュリティ	4-2
データ・レベルのセキュリティ	4-3
5 Oracle Enterprise Command Center Framework のメタデータ	5-1
Oracle Enterprise Command Center Framework のメタデータの概要	5-1
アプリケーション	5-1
ページ	5-2
データ・セット	5-4

メタデータ.....	5-6
6 動的メタデータ.....	6-1
動的属性	6-1
記述オプション	6-1
実行オプション.....	6-3
動的優先順位ルール.....	6-5
7 Oracle Enterprise Command Center Framework の UI コンポーネントの構造...	7-1
構成モデル	7-1
ナビゲーション・コンポーネント	7-10
ビジュアル化コンポーネント.....	7-21
詳細インサイト・コンポーネント.....	7-44
レイアウト・コンポーネント	7-63
共通のコンポーネント・コントロール	7-65
8 Oracle Enterprise Command Centers の拡張方法.....	8-1
概要	8-1
既存のダッシュボードを拡張するシナリオ	8-2
既存のデータ・セットの拡張.....	8-21
A サンプル PL/SQL パッケージ.....	A-1
Oracle iProcurement の例	A-2
Order Management の例.....	A-14
B データ・ロードおよびメタデータ・ロードのシグネチャ	B-1
データ・ロードおよびメタデータ・ロードのシグネチャ.....	B-2
C サンプル・コンカレント・プログラム.....	C-1
サンプル・コンカレント・プログラム	C-2
D データ・セット ECC_SPEC_ID の構造	D-1
概要	D-1
Enterprise Asset Management	D-1
Projects.....	D-3
Payables	D-6

Receivables.....	D-7
Assets.....	D-9
Order Management.....	D-11
Service	D-12
Service Contracts (OKS)	D-13
Incentive Compensation	D-15
iProcurement	D-17
Contract Lifecycle Management for Public Sector	D-18
Discrete Manufacturing.....	D-23
Outsourced Manufacturing.....	D-26
Process Manufacturing.....	D-29
Logistics (Inventory)	D-33
Channel Revenue Management	D-34
Cost Management	D-36
Procurement	D-41
Depot Repair.....	D-43
Human Resources	D-44

索引

ご意見をお寄せください

Oracle E-Business Suite Oracle Enterprise Command Centers の拡張 リリース 12.2 部品番号 F24028-01

このマニュアルの品質および利便性について、お客様からのご意見、ご提案をお待ちしております。お寄せ頂いたご意見は、弊社製品をご利用のお客様により満足していただくための参考とさせていただきます。例:

- 実装手順が正確に漏れなく記載されていましたか?
- 手順の脈略やつながりを理解できましたか?
- 記載内容に誤りはありませんでしたか?
- 情報の構成はタスクの実施に役立ちましたか?
- 異なる情報や画像が必要でしたか? その場合、それはどのような形式の情報ですか?
- 正しい例が使用されていましたか? 他の例が必要でしたか?

記載内容に誤りを見つけた場合またはご意見をお寄せいただく際は、お客様のお名前、製品のライセンス所有者名(会社名)、マニュアルのタイトル、部品番号、章番号、項番号およびページ番号(該当する場合)をお知らせください。

注意: ご報告いただく前に、マニュアルの最新版で問題箇所が訂正されていないかどうかをご確認ください。My Oracle Support および www.oracle.com にある新しい Oracle E-Business Suite リリース・オンライン・ドキュメント CD をご利用ください。ここには、最新のドキュメント・ライブラリと最近リリースまたは改訂されたすべてのドキュメントが収録されています。

ご意見、ご提案は電子メール・アドレス appsdoc_us@oracle.com にお寄せください。

お客様のお名前、住所、電子メール・アドレス、電話番号(任意)をご記載ください。

Oracle ソフトウェアに関するサポートをご希望の場合は、お客様のサポート担当者または Oracle サポート・サービスにお問合せください。

Oracle ソフトウェアのトレーニングまたは指導をご希望のお客様は、Oracle University 製品について弊社の現地事業所にお問合せください。オラクル社の事業所一覧については、弊社 Web サイト www.oracle.com をご覧ください。

はじめに

対象読者

Oracle E-Business Suite Oracle Enterprise Command Centers の拡張リリース 12.2 によるこそ。

Oracle E-Business Suite の製品情報の詳細は、xi ページの「関連情報の入手元」を参照してください。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc> を参照してください。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は、(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup>)を参照してください。

構成

- 1 概要
- 2 Oracle Enterprise Command Center Framework のアーキテクチャ
- 3 データ・ロード・プロセス
- 4 セキュリティ
- 5 Oracle Enterprise Command Center Framework のメタデータ
- 6 動的メタデータ
- 7 Oracle Enterprise Command Center Framework の UI コンポーネントの構造
- 8 Oracle Enterprise Command Centers の拡張方法
- A サンプル PL/SQL パッケージ

- B データ・ロードおよびメタデータ・ロードのシグネチャ
- C サンプル・コンカレント・プログラム
- D データ・セット ECC_SPEC_ID の構造

関連情報の入手元

このマニュアルは Oracle E-Business Suite ドキュメント・ライブラリに含まれています。このマニュアル内に Oracle E-Business Suite の他のドキュメントに関する言及がある場合は、最新のリリース 12.2 バージョンのドキュメントのみを参照してください。

オンライン・ドキュメント

すべての Oracle E-Business Suite ドキュメントはオンラインで入手できます (HTML または PDF)。

- **オンライン・ヘルプ** - オンライン・ヘルプのパッチ (HTML) は My Oracle Support で入手できます。
- **Oracle E-Business Suite ドキュメント・ライブラリ** - Oracle E-Business Suite ソフトウェアのディストリビューションに含まれており、各リリース時点における PDF 版のドキュメントを提供します。
- **Oracle E-Business Suite ドキュメント Web ライブラリ** - Oracle Technology Network (https://docs.oracle.com/cd/E26401_01/index.htm) から入手可能で、Oracle E-Business Suite リリース 12.2 ドキュメントの最新更新版を提供します。ほとんどのドキュメントで PDF 版と HTML 版が用意されています。
- **リリース・ノート** - このリリースでの変更点、新機能、既知の問題およびその他の詳細は、対象製品のリリース・ノートを My Oracle Support から入手してご確認ください。
- **Oracle Electronic Technical Reference Manual** - Oracle Electronic Technical Reference Manual (eTRM) には、データベース図と各 Oracle E-Business Suite 製品のデータベース表、フォーム、レポートおよびプログラムに関する詳細が記載されています。これらの情報は、既存のアプリケーションのデータを変換し Oracle E-Business Suite データを非 Oracle アプリケーションに統合して、Oracle E-Business Suite 製品のカスタム・レポートを作成する際に便利です。Oracle eTRM は My Oracle Support から入手できます。

関連ガイド

次の関連ドキュメントがお手元にあることをご確認ください。インストール固有の要件によっては、その他のマニュアルまたはガイドが必要になることがあります。

Oracle E-Business Suite の概念

Oracle E-Business Suite リリース 12.2 のデプロイまたは大幅な構成変更を計画しているお客様向けのドキュメントです。Oracle E-Business Suite のアーキテクチャとテクノロジー・スタックについて説明した後に、戦略的なトピックへと話を進めます。特定の目標を達成するのに必要な作業の概要およびインストールと構成の選択肢が記載されています。

Oracle E-Business Suite ユーザーズ・ガイド

Oracle E-Business Suite のユーザー・インタフェース(UI)を使用した製品のナビゲーション、データの入力と問合せおよびコンカレント要求の実行方法を説明します。プリファレンスの設定や UI のカスタマイズ方法など、基本的な情報も記載されています。Oracle Enterprise Command Centers に関する簡単な説明も含まれています。また、巻末には Oracle E-Business Suite のアクセシビリティ機能とキーボード・ショートカットに関する説明があります。

統合リポジトリ

Oracle Integration Repository は Oracle E-Business Suite アプリケーションで公開されているサービス・エンドポイントに関する情報をまとめたものです。Oracle E-Business Suite のすべてのビジネス・サービス・インタフェースをまとめたカタログを提供します。このツールを使用すると、システム、アプリケーションまたはビジネス・パートナーとの統合に適したビジネス・サービス・インタフェースを簡単に見つけてデプロイできます。

Oracle Integration Repository は Oracle E-Business Suite の一部として提供されます。インスタンスにパッチを適用すると、このリポジトリのコンテンツは使用環境のインタフェースのリビジョンに合わせて自動的に更新されます。

データベース・ツールを使用して Oracle E-Business Suite データを変更しない

Oracle E-Business Suite のデータを変更する際は、SQL*Plus や Oracle Data Browser、データベース・トリガーなどのツールを使用しないでください。

オラクルでは、Oracle データベースでデータを作成、保存、変更、取得および保守するための強力なツールを各種提供しています。しかし、SQL*Plus などの Oracle ツールを使用して Oracle E-Business Suite データを変更すると、データの整合性が失われ、データ変更の監査ができなくなる恐れがあります。

Oracle E-Business Suite の表は相互に関連しているため、Oracle E-Business Suite フォームを使用して変更を加えると、他の多数の表も同時に更新されます。しかし、Oracle E-Business Suite 以外で Oracle E-Business Suite データを変更した場合、1つの表で行を変更し、その変更に関連する表に反映しないという状況が発生する可能性があります。表間の同期が失われると誤った情報を取得する可能性があり、Oracle E-Business Suite 全域で想定外の結果が生じる恐れがあります。

Oracle E-Business Suite を使用してデータを変更すると、その変更の有効性が Oracle E-Business Suite によって自動的にチェックされます。また、だれが情報を変更したかの履歴も維持されます。データベース・ツールを使用してデータベース表に情報を入力した場合、不正な情報を格納してしまう可能性があります。また、SQL*Plus などのデータベース・ツールでは変更記録が保持されないため、だれが情報を変更したのかを追跡できなくなります。

Oracle Enterprise Command Center Framework の概要

Oracle Enterprise Command Center Framework は、情報の検出、ビジュアル化および探索機能を Oracle E-Business Suite ユーザー・インタフェースに埋め込む形で提供します。Oracle Enterprise Command Center Frameworkを使用すると、サプライ・チェーン、会計、生産、調達といった様々な機能領域にビジネス・ダッシュボードを作成できます。

Enterprise Command Center ダッシュボードでは、ユーザーが対話型のビジュアル・コンポーネントやガイド付きの検出機能を使用してトランザクション情報をナビゲートし、探索型のデータ分析を実行できます。詳細なデータ・インサイトドリルダウンする機能が処理主導型のダッシュボードを実現します。

各 Oracle Enterprise Command Center に備わる一連の機能により、ユーザーは重要な質問に対する回答を得たり、ビジネス・インサイトを発見したり、関連する処理にドリルダウンしたりできます。素早く直感的にデータを分析できるため、データに基づく確かな意思決定が可能になります。

Oracle E-Business Suite にはすぐに使用できるダッシュボードが付属しており、様々な製品ファミリおよび機能領域で使用できます。機能性に優れたダッシュボードはそのテクノロジーの特性から、あらゆるユース・ケースやシナリオに対応する対話性の高いユーザー・エクスペリエンスを提供します。また、拡張性も備えているため、機能セットの用途を拡張し、実装特有の要件に対応することもできます。

このマニュアルでは、Oracle Enterprise Command Center Framework アーキテクチャの主要な側面と、Oracle Enterprise Command Center ダッシュボードの構造について詳しく説明します。実現可能な拡張シナリオについて詳述します。パッチ適用後も拡張機能を維持するためのベスト・プラクティスも紹介します。

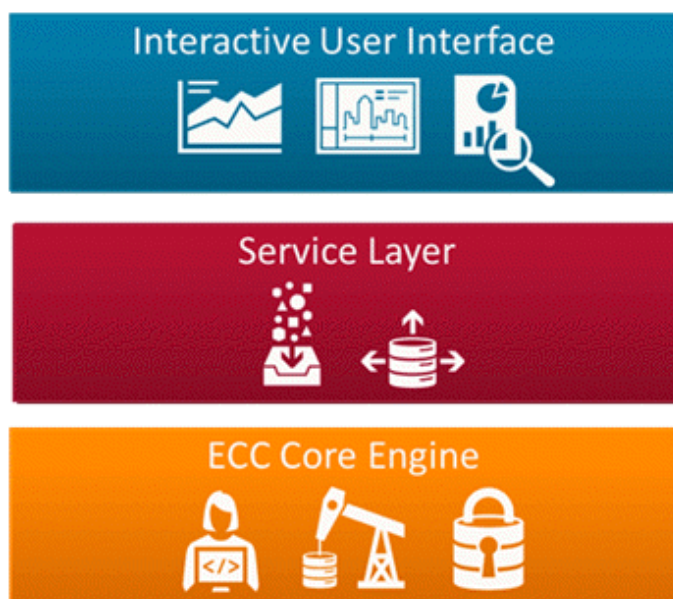
Oracle Enterprise Command Center Framework のアーキテクチャ

Oracle Enterprise Command Center Framework のアーキテクチャの概要

Oracle Enterprise Command Center Framework は、対話型 JavaScript フロント・エンドと Java バック・エンド・エンジンで構成されるミドルウェア・テクノロジーです。このアーキテクチャには 3 つの主要コンポーネントが含まれています。

1. 対話型ユーザー・インタフェース・レイヤー。オープン・ソース・テクノロジー Oracle JavaScript Extensions Toolkit(JET)上に構築されています。Oracle Enterprise Command Center Framework 開発フレームワークでは、ドラッグ・アンド・ドロップ型のレイアウト設計、すべてのコンポーネントの宣言的構成およびダッシュボードの様々なセクション間で自動的に行われる対話機能を通じ、コードを記述することなくダッシュボードを作成できます。
2. サービス・レイヤー。Oracle Enterprise Command Center Framework から提供されるサービスを使用して、クライアントはデータ・セットを定義し、データ・セット内にデータを取り込み、メタデータを通じて高度な動作を制御できます。ダッシュボードで新たなサービスを公開することで、データに対して高度な問合せを実行して集計値や積上値、関連データ要素などを取得することが可能になります。
3. Oracle Enterprise Command Center Framework コア・エンジン。Oracle Enterprise Command Center Framework の中核を成すコンポーネントです。ソース・システムつまり Oracle E-Business Suite から取得したデータはここに格納されます。データの保管時に分析が実行されるため、すばやい高度なデータ・アクセスが可能となり、様々なパターンのユーザー問合せに対応します。オプションで高度な言語処理機能を導入して、辞書ベースのデータ解析と取得をサポートすることもできます。

Oracle Enterprise Command Center Framework のアーキテクチャ



Enterprise Command Center ダッシュボードを作成するには、次の手順を実行する必要があります。

1. アプリケーションと関連データ・セットを定義します。ページをレンダリングし、ユーザー・インタフェース上のすべての属性の動作を制御するのにメタデータが必要になります。高度な関連、計算属性、動的な UI 対話を必要に応じて指定します。
2. ソース・システムつまり Oracle E-Business Suite から前の手順で定義した Oracle Enterprise Command Center Framework データ・セットへのデータの取り込み。
3. 取り込んだデータを公開するページ、および Oracle Enterprise Command Center Framework コンポーネント・パレットにある対話型コンポーネントを利用して、ビジネス・ダッシュボードを作成します。

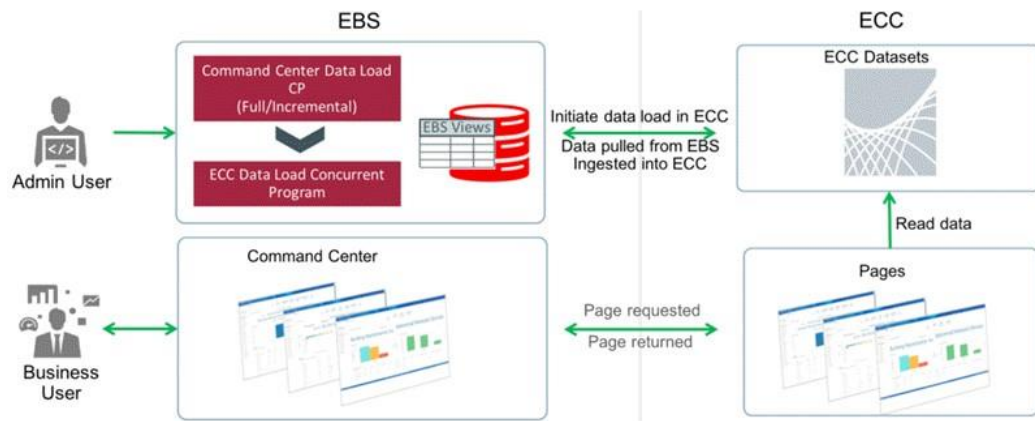
データ・ロード・プロセス

データ・ロード・プロセスの概要

データはすべてのダッシュボードの基盤です。データはソース・システムで定義されたルールと形状に従って、ソース・システムから Oracle Enterprise Command Center Framework エンジンに取り込まれます。データはダッシュボードで解決する様々なユース・ケースに合わせてモデル化されます。真のデータのソースは Oracle E- Business Suite システムということになります。Oracle Enterprise Command Center Framework アーキテクチャは Oracle E-Business Suite データベースの能力を利用して、Oracle E-Business Suite から Oracle Enterprise Command Center Framework へのデータ・ロード・プロセスを実行します。複雑なツールやテクノロジーを新たに導入するのではなく、SQL と PL/SQL の能力および柔軟性を利用して、ダッシュボードを動かすのに必要な形状でデータを準備し取得します。

次の図は、データ・ロード・プログラムのアーキテクチャの概要です。コンカレン ト・プログラムが Oracle Enterprise Command Center Framework でデータ・ロードのトリガーとなり、Oracle E-Business Suite データを引き出す様子を示しています。図の下半分は、ビジネス・ユーザーが Oracle Enterprise Command Center Framework に取り込まれたデータをダッシュボードで利用する様子を示しています。

データ・ロード・プログラムのアーキテクチャ概要



データ・ロード・プロセス定義

フル・ロード

初めて Oracle Enterprise Command Center Framework をインストールし、特定のビジネス・エリアに対しアプリケーションをインポートするときには、最初の手順としてフル・データ・ロードを実行する必要があります。このプロセスを実行すると、Oracle Enterprise Command Center Framework エンジン内に関連データ・セットが作成され、Oracle E-Business Suite データベースから Oracle Enterprise Command Center Framework にデータを取り込むプロセスが開始されます。データ・セットがすでに作成されている場合は、フル・ロードを通じてデータ・セットを作り直し、新しいデータを移入することもできます。

ビジネス・ダッシュボードへのフル・ロードは同じビジネス・エリアで所有されるコンカレント・ジョブがトリガーとなります。このプログラムは Oracle Enterprise Command Center Framework で定義された関連するデータ・セット・ロード・ルールを作動させます。各データ・セットにはデータをロードするためのルールを 1 つ以上指定できます。各ルールには Oracle E-Business Suite データベースに対する問合せが 1 つ以上格納されています。Oracle Enterprise Command Center Framework は Oracle E-Business Suite との接続に EBS SDK を使用するため、データ・ロード・コードでデータの準備、整備およびラングリングのロジックを実行したうえで、Oracle Enterprise Command Center Framework にデータを取り込むことができます。

増分ロード

増分データ・ロードのロジックと手順はフル・データ・ロードとほぼ同じですが、大きな違いが 2 点あります。まず、増分ロードはデータ・セットがすでに作成済でメタデータも取得済であることを前提としているため、これらの条件を満たしていない場合、データ・ロードは失敗します。また、増分データ・ロードでは、最後のフル・ロードまたは増分ロードが成功した時点以降に変更されたデータのみが Oracle E-Business Suite から選択されます。

メタデータ・ロード

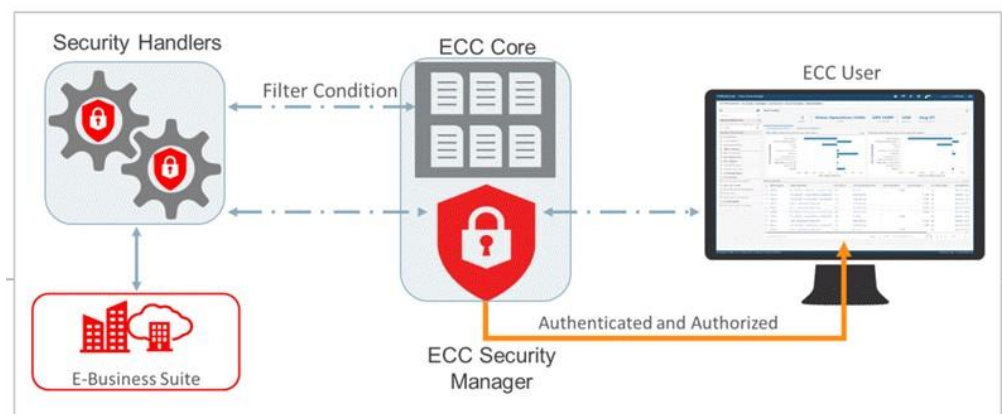
メタデータ・ロードは実行時の動的メタデータ定義を可能にする高度な機能です(ダッシュボード定義には含まれていません)。この機能の好例が Oracle E-Business Suite の付加フレックスフィールド(DFFs)です。これらは顧客のインスタンスによって違いがあるため、設計時には定義できません。そのため動的メタデータ定義フェーズを利用し、このようなメタデータを実行時に指定できるようにして、データを後からフル・ロードや増分ロードを通じて移入します。他のタイプのロードと同じ手順とルールに従うため、このロードは Oracle E-Business Suite のコンカレント・プログラムがトリガーとなり、Oracle E- Business Suite データベース・セッションを実行時環境として利用します。

セキュリティ

Oracle Enterprise Command Center Framework のセキュリティの概要

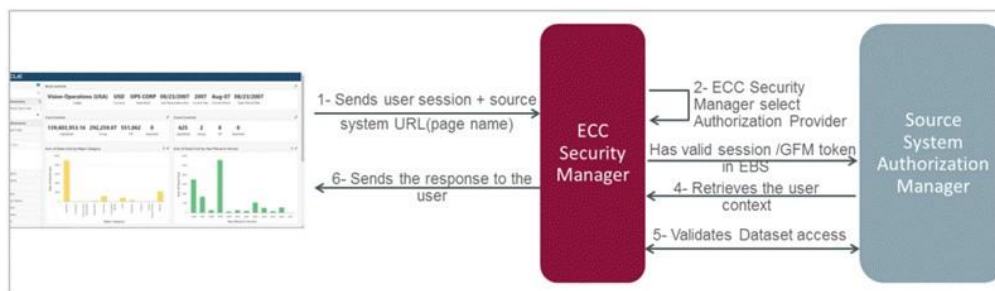
Oracle Enterprise Command Center Framework は Oracle E-Business Suite が提供するセキュリティ・モデルを利用して、ダッシュボードとその元データへのアクセスを保護します。Oracle E-Business Suite からのユーザー・アクセス権限を継承し、認証を受けたユーザーに対し Oracle Enterprise Command Center Framework ダッシュボードへのアクセスを許可します。

Oracle Enterprise Command Center Framework のセキュリティ



Oracle Enterprise Command Center Framework の認証

Oracle Enterprise Command Center Framework の認証および認可



この図は、Oracle Enterprise Command Center Framework が埋込みダッシュボードへのアクセスを認証および認可するとき使用するプロセスを説明しています。

実行されるチェックについて説明します。

1. Oracle Enterprise Command Center Framework 認証: ビジネス・ユーザーが埋込みダッシュボードにアクセスすると、Oracle Enterprise Command Center Framework は認証済の有効な Oracle E- Business Suite セッションがあるかどうかを確認し、その後に他のチェックを実行します。次に認可コントロールが適用され、ユーザーに公開するダッシュボードについて詳細な制御が行われます。
2. Oracle Enterprise Command Center Framework 認可: Oracle E-Business Suite セッション経由での認証が済むと、Oracle Enterprise Command Center Framework データ・セット権限名と FND フォーム機能の照合をとって認可確認が行われます。

ページ・レベルのセキュリティ

- Oracle Enterprise Command Center Framework では、Oracle E-Business Suite で定義されている FND_FORM_FUNCTION を使用してページが保護されます。
- 提供された Oracle E-Business Suite ロール(Role Based Access Control: RBAC)を使用して、Oracle Enterprise Command Center Framework ダッシュボードへのアクセス権を各職責に割り当てます。

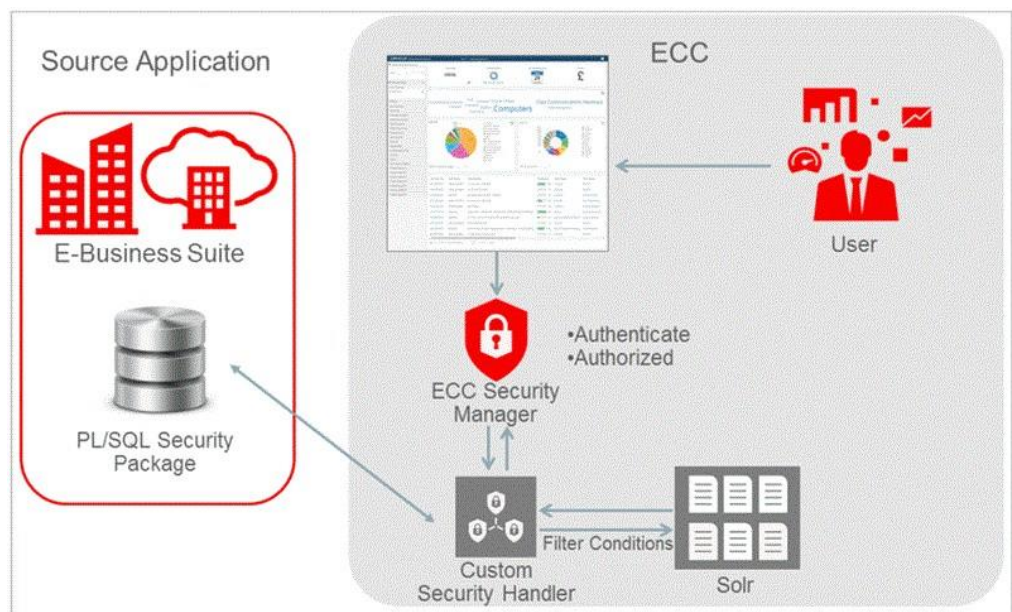
Oracle Enterprise Command Center Framework では ECC 開発者職責が提供されます。この職責を使用して Oracle Enterprise Command Center Framework ホーム・ページにアクセスし、様々な開発者機能や管理者機能を使用できます。

ECC 開発者職責を持つユーザーとしてアクセスしたダッシュボード・ページでは、ページ・レイアウトやコンポーネントの構成を変更できます。

データ・レベルのセキュリティ

Oracle Enterprise Command Center Framework データ・セット内のデータにアクセスするには、ユーザーは少なくとも 1 つのデータ・セット権限(Oracle E-Business Suite フォーム機能で定義および制御される)を付与されている必要があります。データ・セットに対するアクセス権が確認されると、そのデータ・セットに対する以降の問合せにセキュリティ・ハンドラが適用されるため、ユーザーは自身に許可されているデータのサブセットにしかアクセスできなくなります。これは次のレベルのセキュリティであり、ダッシュボードに表示されるデータにデータ・アクセス制限を適用するものです。この制限は Oracle E-Business Suite の設定に基づき決定されるため、他のビジネス・ダッシュボードにも適用される場合があります(営業単位アクセス、在庫組織アクセス、資産台帳アクセスなど)。

データ・セキュリティ



Oracle Enterprise Command Center Framework のメタデータ

Oracle Enterprise Command Center Framework のメタデータの概要

Oracle Enterprise Command Center Framework ダッシュボードの動作や、構成コンポーネント、含まれるデータ要素はメタデータをととして制御されます。メタデータは開発者/管理 UI の様々な場所で指定します。一部のメタデータは設計時に定義されて、メタデータとして単純に参照されます。また、実行時に動的に定義されるメタデータもあり、これは設計時には不明または指定できない動作を制御するのに使用されます。これは動的メタデータと呼ばれるもので、このタイプのメタデータについては以降の章で説明します。

アプリケーション

コマンド・センターは複数のダッシュボードで構成されており、機能領域の様々な側面を公開します。Oracle Enterprise Command Center Framework アプリケーションはコマンド・センターを構成するアーティファクト(ダッシュボードと関連データ・セットなど)を論理的にグループ化したものです。

アプリケーション定義

Applications
Define applications

* Application Short Name

* Application Name

Data Sets

- Reconciliation (fa-rec) X
- Pipeline (fa-clr) X
- Assets (fa-asset) X
- Mass Transactions (fa-masstrans) X

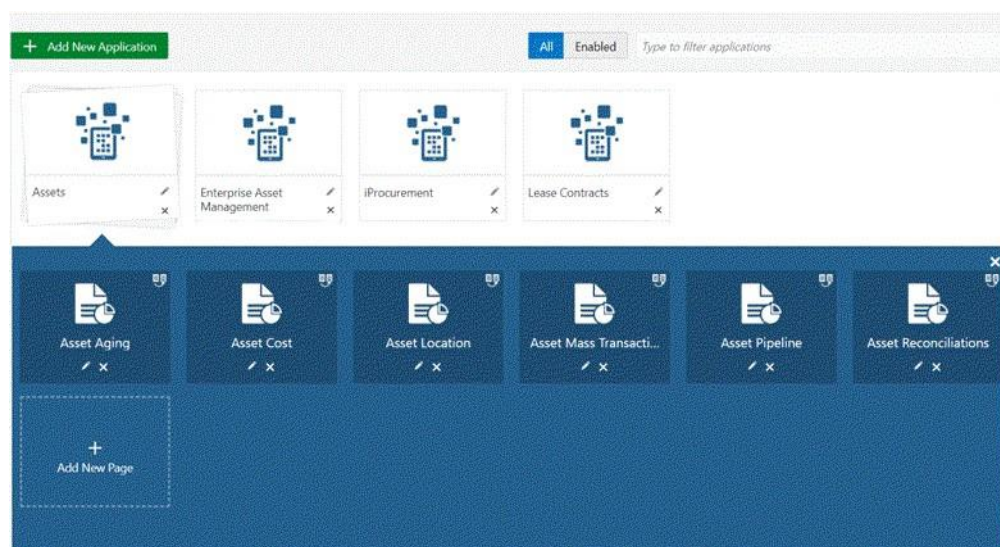
Enabled ☒

アプリケーションはダッシュボードを動作させるために必要なすべての要素をカプセル化したものです。アプリケーションはデータ・セットを参照し、これによりロード・ルールを通じてデータの移入方法が決定され、メタデータを通じて UI 上のすべての属性の動作が決定されます。あるアプリケーションが他のアプリケーションが所有するデータ・セットを参照することもできます。

ページ

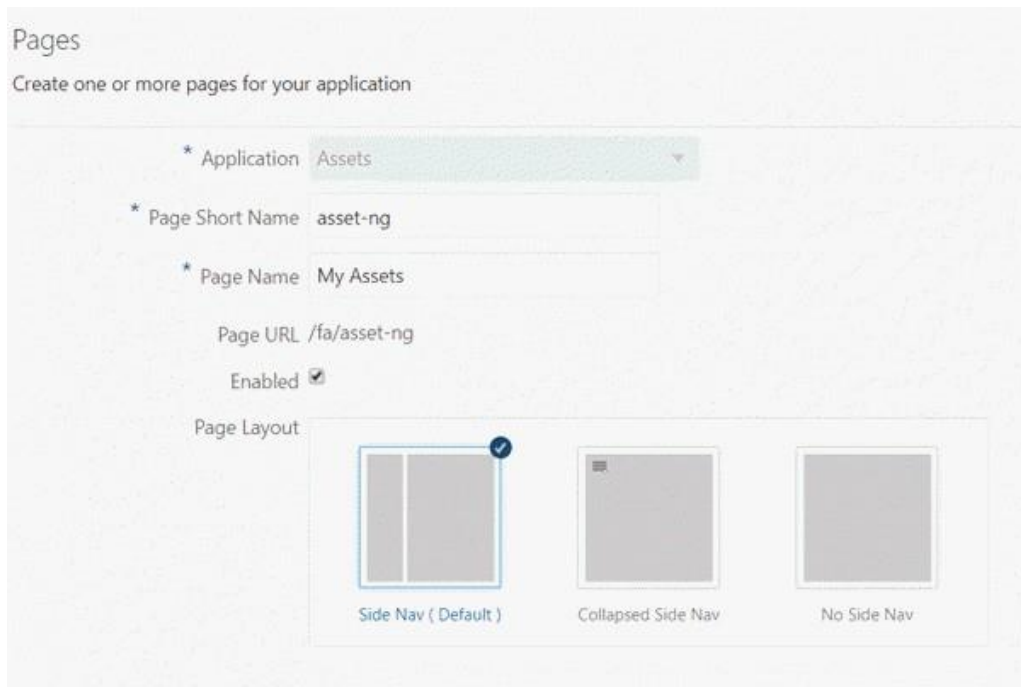
アプリケーション・ページは、ある特定タイプの機能(ページに表示されたデータのフィルタリング、視覚的なデータ表現の表示、レコードまたはレコード属性のリスト表示、特定のデータ値の強調表示など)を実現するために設計されたすべてのビジュアル化コンポーネントのためのホーム・ページです。

アプリケーション・ページ



ページはビジネス要件に従って、アプリケーションの下にグループ化されます。ページ定義ではページ名や短縮名、ページ・レイアウトなどの重要情報を設定できます。

ページ・レイアウト



ページのレイアウト構成は 3 種類から選択できます。

- サイド・ナビゲーションのあるレイアウト
- 縮小されたサイド・ナビゲーションのあるレイアウト
- サイド・ナビゲーションのないレイアウト

デフォルトのレイアウトではサイド・ナビゲーションが有効になります。ナビゲーション・パネルが非表示のとき、ナビゲーション・アイコンはページに縦方向に表示されず、かわりにハンバーガ・アイコンの形で表示されます。

サイド・ナビゲーションが縮小されているときには、選択されている絞込みはすべてファネル・アイコンで表示されます。デフォルトで同じことが、縮小サイド・ナビゲーションおよびサイド・ナビゲーションなしのレイアウトにも適用されます。ページに絞込みを適用すると、選択した絞込みファネル・アイコンが、フィルタ適用済であることを示す表示に変わります。

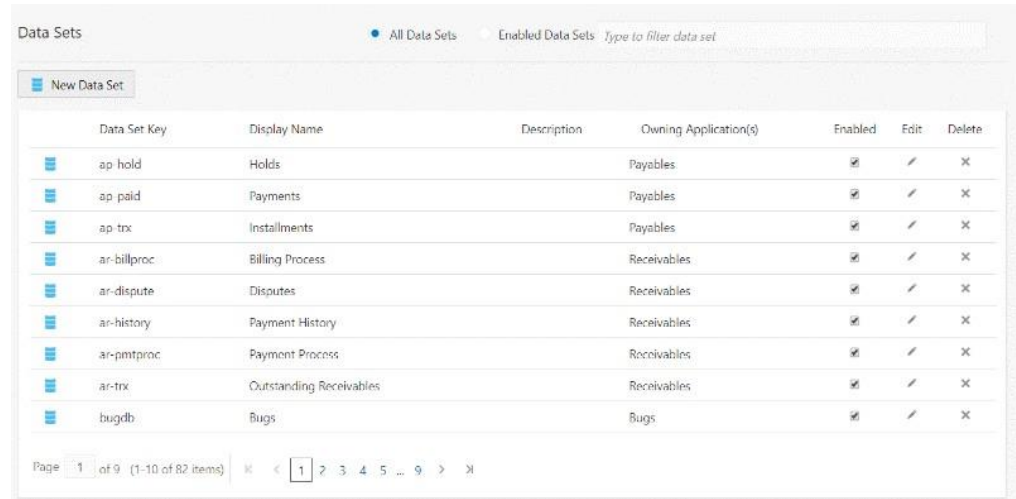
データ・セット

データ・セットは属性を論理的および物理的にグループ化したもので、ビジネス・ダッシュボードの動作とユース・ケースを下支えするものです。論理的な観点では、Oracle Enterprise Command Center Framework の 1 つ以上のダッシュボードを通じて利用される各種のユース・ケースをサポートするように設計されます。通常、元データとの制約のない対話を可能にする設計になっています。

物理的な観点では、データ・セットには 1 つ以上のレコードが、エンタープライズ・システムに格納されているエンティティの特定レベルの詳細を表す識別キーとともに格納されます。

データ・セットでは、そのデータ・セットでサポートしているデータ・ロード・ルールと、それらをどのように使用してデータ・セットにデータを移入するかが宣言されます。データ・セットは所有されているデータ・セットとしていずれかのアプリケーションに割り当てる必要がありますが、他のアプリケーションから参照することもできます。所有者であるアプリケーションがそのデータ・セットにデータを移入する役割を担います。

データ・セット・ページ



Data Set Key	Display Name	Description	Owning Application(s)	Enabled	Edit	Delete
ap hold	Holds		Payables	☒		
ap paid	Payments		Payables	☒		
ap tx	Installments		Payables	☒		
ar-billproc	Billing Process		Receivables	☒		
ar-dispute	Disputes		Receivables	☒		
ar-history	Payment History		Receivables	☒		
ar-pmtproc	Payment Process		Receivables	☒		
ar-trx	Outstanding Receivables		Receivables	☒		
bugdb	Bugs		Bugs	☒		

各データ・セットにはロード・ルール(データ・ロード・プロセス)とセキュリティ・ルール(セキュリティ)があります。詳細は、3-1 ページの「データ・ロード・プロセスの概要」および 4-1 ページの「Oracle Enterprise Command Center Framework のセキュリティの概要」を参照してください。

データ・セット・ロード・ルール

Data Sets

Define data sets and load rules

Save Cancel

* Data Set Key ap-hold

* Display Name Holds

Data Set Description

Enabled ☒

Load Rules Security Rule

Add Delete

Load Type	Package Name	Procedure Name	Connection name	Sequence
☒ Incremental Load	AP_ECC_UTIL_PVT	GET_ECC_DATA_LOAD_INFO	ebsdb	1
☒ Full Load	AP_ECC_UTIL_PVT	GET_ECC_DATA_LOAD_INFO	ebsdb	1

メタデータ

データ・セットに格納される属性は、ユーザー・インタフェース上での動作を決定するメタデータ・プロパティによって制御されます。計算、バケット、優先順位ルールといった付加価値機能も追加で指定できます。

属性メタデータ

属性メタデータでは、属性の次のような特性が定義されます。

- 名前およびタイプ。
- Display Name。
- 構成パラメータ。たとえば、属性を検索できるかどうか。
- ナビゲーション性の設定。例: 使用可能な絞込みのレコード数を表示するかどうか、複数選択、絞込みのソート方法。

属性レベルのプロパティ

次の表に、属性レベルのメタデータ・プロパティを示します。

メタデータ属性レベルのプロパティ

プロパティ	デフォルト値	説明
Attribute Key		データ・ロード問合せで定義された属性のキー。必須属性でデータ・セットごとに異なります。
Source Data Type		データ・ロード問合せで定義された属性のデータ・タイプ。
Profile		すべての属性にプロファイルが関連付けられており、プロファイルに含まれているプロパティが属性の動作を決定します。
Display Name		わかりやすい形式の属性名。 ユーザーは属性の表示名を変更できます。この変更について制約事項はありません。
Custom Display Name		提供された属性の表示名を上書きするときに使用します。
Text Searchable	Y	「Y」に設定した場合、属性がレコード検索の対象になります。 「N」に設定した場合、属性がレコード検索の対象外となります。
Refineable?	Y	属性を絞り込みとして使用できるかどうかを指定します。 「Y」に設定した場合、属性は絞り込み可能になります。 「N」に設定した場合、属性は絞り込み不可になります。
Translatable?	N	属性を翻訳可能とするかどうかを指定します。
Refinement Behavior	multi-select-or	属性に複数選択機能を設定します。次の値 (single、multi-or、multi- and) で上書き可能です。

プロパティ	デフォルト値	説明
Refinement Order	lexical	ナビゲーション・メニュー内での絞込みの表示順序。使用できる値: - lexical - 絞込みをアルファベット順または数字順でソートします。 - record-count - 絞込みを各絞込みのレコード数の降順でソートします。

Record Identifier

データ・セットは非正規化の度合いが高くなることがあるため、レコード識別子という簡単かつ強力な機能を使用することでデータ・セット・レコードを様々な粒度レベルで表示できます。SQL 問合せのグループ化キーと同じ概念を利用しています。

属性グループ

属性グループは、機能/ビジネス上の意味に基づく表示を実現するために、属性を論理的にグループ化したものです。これにより、使用可能な絞込みコンポーネントと結果表コンポーネントの雑然とした状態を解消できます。

属性グループはデータ・セット・レベルで定義されます。各データ・セットには属性グループを 1 つ以上指定でき(または 1 つも指定しない)、各属性グループには 1 つ以上の属性を指定できます(または 1 つも指定しない)。グループの設計は設計時に管理 UI で行うか、実行時にメタデータ・ロード・フェーズで行います。

属性グループ

Metadata
Manage options and precedence rules of data sets attributes

* Data Set: Assets (fa-asset)

Attributes | **Attribute Groups** | Record Identifiers | Calculated Attributes | Buckets | Precedence Rules | Associations

Add Delete Type to filter attribute groups

Sequence #	Group Key	Group Name	Custom Display Name
10	asset_details	Asset	
Attribute Name			
— Asset Number (ASSET_NUMBER) ▼ + x			
— Category (MAJOR_CATEGORY) ▼ + x			
— Category (MINOR_CATEGORY) ▼ + x			
— Asset Type (ASSET_TYPE) ▼ + x			
— Property Code (PROPERTY_TYPE) ▼ + x			
20	fin_details	Financial	
30	source	Source	

計算属性

メタデータで計算属性を定義することで、動的な動作を実現したり、取り込むデータの複雑性を低減したりできます。2 種類の計算属性があります。

- レコードベースの日付関数
 - 時間ベースのバケットを促進
 - 日付計算で `sysdate` を使用
 - 計算の結果単位として日、月または年を指定可能
 - 任意のコンポーネントでコンポーネント構成レベルのデータ条件として使用可能
- グループベースの集計関数
 - データ・ロード時には計算できない、その場での集計が可能
 - コンポーネント構成で選択したディメンションごとにグループ化
 - レコードのグループに対して作用
 - ビジュアル化コンポーネントでメトリックとして使用される(さらなる集計は不要)

計算属性

Metadata Save Cancel

Manage options and precedence rules of data sets attributes

* Data Set: Order Lines (ont-lines)

Attributes | Attribute Groups | Record Identifiers | **Calculated Attributes** | Buckets | Precedence Rules | Associations

Add Delete

☐	Attribute Key	Attribute Name	Custom Attribute Name	Formula
☐	REQUEST_DATE_FOR_BUCKET	Request Date for Bucket		((REQUEST_DATE)-(sysdate)), D...
	Attribute Request Date	Operator -	Attribute sysdate	Unit DAY
☐	ORDER_DATE_FOR_BUCKET	Order Date for Buckets		((sysdate)-(ORDER_DATE)), DAY
☐	DAYS_LATE	Days Late		((sysdate)-(SCHEDULE_SHIP_D...
☐	years_ago	Years Ago		((sysdate)-(ORDER_DATE)), YEAR

バケット

バケットを使用すると、指定の範囲に収まった特定のメトリックに基づいてデータ・レコードをビンニングできます。制御メトリックとしてデータ・セット内の通常の属性と計算属性を使用できます。

- 表示順序は連番によって決まります。
- 含める範囲の開始値と除外する範囲の終了値を指定できます。
- 空の開始値/終了値は正の無限大か負の無限大を表します。
- ギャップと重なりも許可されます。

バケット

Metadata

Manage options and precedence rules of data sets attributes

SaveCancel

* Data Set Order Lines (ont-lines) ▼

AttributesAttribute GroupsRecord IdentifiersCalculated AttributesBucketsPrecedence RulesAssociations

AddDelete

☐

Bucket Key	Bucket Name	Custom Bucket Name	Attribute
☐ REQUEST_DATE_BUCKET	Request Date		Request Date for Bucket ▼

Seq #	Label	Start	End	
1	In the Past		0	+ x
2	Today	0	1	+ x
3	1-3 Days	1	4	+ x
4	4-7 Days	4	8	+ x
5	7+ Days	8		+ x

バケットを使用することで、レコード上に動的ディメンションを効率良く作成できます。作成したディメンションは、表示や絞込み目的でチャート上で使用できます。

バケットを使用したチャート



優先順位ルール

優先順位ルールは、使用可能な絞込みコンポーネント内で追加の属性を開示するかを動的に決定します。トリガー属性であるデータ条件が満たされた場合にのみ、追加の属性を表示させることができます。

優先順位ルールではトリガー属性とターゲット属性を定義し、トリガー属性によって追加のターゲット属性がユーザーに開示されます。

優先順位ルール

Metadata
Manage options and precedence rules of data sets attributes

* Data Set: Assets (fa-asset)

Attributes | Attribute Groups | Record Identifiers | Calculated Attributes | Buckets | **Precedence Rules** | Associations

Add | Duplicate | Delete

Type to filter precedence rules

Trigger Attribute	Trigger Value	Target Attribute	Enable
Category (MAJOR_CATEGORY)	*	Category (MINOR_CATEGORY)	☒

関連

関連を使用すると、共通の属性を持つデータ・セット間にリンクを定義することで 2 つのデータ・セット間に明示的な接続を設定できます。Oracle Enterprise Command Center Framework

はこの情報を使用して、宛先データ・セット内のデータに対しソース・データ・セットの絞込み状態に基づいた連携的なフィルタリングを適用します。

また、Oracle Enterprise Command Center Framework は、同一のデータ・セットに関連付けられていながら直接のリンクが設定されていない 2 つのデータ・セットに対し暗黙的な関連を設定します。この高度な機能により、直接関連付けられていないデータ・セット間でも高度な絞込み状態が継承されます。

関連

Metadata

Manage options and precedence rules of data sets attributes

SaveCancel

* Data Set: Assets (fa-asset)

AttributesAttribute GroupsRecord IdentifiersCalculated AttributesBucketsPrecedence RulesAssociations

AddDelete

Type to filter associations

	Source Data Set	Source Attribute	Target Data Set	Target Attribute	Enabled
☐	Assets (fa-asset)	Asset	Reconciliation (fa-rec)	Asset	☒
☐	Assets (fa-asset)	Book Type Code	Pipeline (fa-cl)	Book Type Code	☒
☐	Assets (fa-asset)	Book Type Code	Reconciliation (fa-rec)	Book Type Code	☒
☐	Assets (fa-asset)	Ledger	Pipeline (fa-cl)	Ledger	☒
☐	Assets (fa-asset)	Asset Number	Mass Transactions (fa-masstrans)	Asset Number	☒
☐	Assets (fa-asset)	Book Type Code	Mass Transactions (fa-masstrans)	Book Type Code	☒

Page 1 (1 - 6 of 6 items) < 1 > →

動的メタデータ

動的属性

前章で説明したように、場合によっては実行時にメタデータを定義することがあります。これにより動作を動的に制御できます。この手法はユーザー定義属性に適しています。

動的属性を定義するには、データ・セットにメタデータ・ロード・フェーズを定義する必要があります。ユーザーがメタデータ・ロードを実行するたび、Oracle Enterprise Command Center Framework がすべての動的属性をクリーン・アップして、動的に定義された新しい属性セットを挿入します。アプリケーション定義で提供される標準の定義済属性は、このリフレッシュ・プロセスの影響を受けません。

動的属性は次のいずれかの方法で定義します。

- **記述オプション:** Oracle Enterprise Command Center Framework によって解析される問合せで、取得した属性のメタデータを定義します。問合せは実際には実行されません。
- **実行オプション:** 実行される問合せで、属性のメタデータを定義します。

記述オプション

このオプションでは、SQL 問合せの `select` 列リスト・メタデータに基づいて属性リストが作成されます。`select` 文内のすべての列を取得し、各列の名前、タイプ、精度およびスケールを読み取り、対応する属性をデータ・セット内に作成します。複数言語モードで実行されるように構成した Oracle Enterprise Command Center Framework では、属性表示名はそのままの状態データベース解析出力から取得されるため、サポートしている各言語について管理者が翻訳を提供する必要があります。

'記述'関数の属性マッピング

属性名	デフォルト値
ATTRIBUTE_KEY	データ・ロード問合せで定義された属性のキー。必須属性でデータ・セットごとに異なります
DISPLAY_NAME	列名。アンダースコアは空白スペースに置換されます
CUSTOM_DISPLAY_NAME	空の文字列
PROFILE	タイプ値に基づきます。後述の注意を参照してください。
SEARCHABLE_FLAG	属性タイプ文字列「Y」または「N」
REFINABLE	属性タイプ文字列「Y」または「N」
SHOW_IN_GUIDED_DISCOVERY	動的属性の場合は「Y」、標準属性の場合は「N」
REFINEMENT_BEHAVIOUR	multi-select-or
REFINEMENT_ORDER	lexical
SHOW_RECORDS_COUNT	「N」
COLLAPSED	「Y」
TRANSLATABLE	「N」
LANGUAGE	ユーザーが管理側から実行した場合は、ログイン・ユーザーの言語。ユーザーがメタデータ・ロードから実行した場合は Oracle Enterprise Command Center Framework の基本言語
SOURCE_LANGUAGE	LANGUAGE

この表内の"**Profile**"については、タイプ値に基づいてデフォルト値が決まります。

String => String profile

Date => Date profile

```

Number => PrecisionSize > 0 => columnSize < 10 => float profile
           => columnSize > 10 => double profile
           => PrecisionSize < 0 => columnSize < 10 => int profile
           => columnSize < 10 => long profile

```

実行オプション

提供された `select` 文を実行し、返された行をループして、対応する動的属性を行ごとに作成します。提供される `select` 文内に次の列リストが含まれている必要があります。

問合せですべての必須フィールドが返される必要があります。返されなかった場合は例外が発生し、根本的な原因が示されます。

'実行関数の属性マッピング

属性名	デフォルト値	必須/オプション	オプション
NAME		必須	
Type		必須	
Precision		必須	
Scale		必須	
DISPLAY_NAME	名前。アンダースコアは空白スペースに置換されます。	オプション	
CUSTOM_DISPLAY_NAME	空の文字列	オプション	
SHOW_IN_GUIDED_DISCOVERY	動的属性の場合は「Y」、標準属性の場合は「N」	オプション	「Y」、「N」
REFINEMENT_BEHAVIOUR	multi-select-or	オプション	multi-select-or、multi-select-and、single-select
REFINEMENT_ORDER	lexical	オプション	lexical、レコード件数
SHOW_RECORDS_COUNT	「N」	オプション	「Y」、「N」

属性名	デフォルト値	必須/オプション	オプション
COLLAPSED	「Y」	オプション	「Y」、「N」
PROFILE	タイプ値に基づく	オプション	ATTR_PROFILE_KE Y(表 ECC_DATASET_ ATTR_PROFS 内) の 1 つ。
TRANSLATABLE	「N」	オプション	
LANGUAGE	ログイン・ユーザーの言語 > ユーザーが管理 サイドから実行した場合。 Oracle Enterprise Command Center Framework の基本言語 > ユーザーがメタデー タ・ロードから実行した 場合。	オプション	ecc- config.properties ファイルの ecc.languages に追 加した Oracle Enterprise Command Center Framework 言語の 1 つ
SOURCE_LANGUAGE	ログイン・ユーザーの言語 > ユーザーが管理 サイドから実行した場合。 Oracle Enterprise Command Center Framework の基本言語 > ユーザーがメタデー タ・ロードから実行した 場合。 Language > この言語が 存在し、サポートされて いる言語が存在しない 場合。	オプション	ecc- config.properties ファイルの ecc.languages に追 加した Oracle Enterprise Command Center Framework 言語の 1 つ
SEARCHABLE_FLAG	「N」	オプション	「Y」、「N」
REFINABLE	「N」	オプション	「Y」、「N」

属性名	デフォルト値	必須/オプション	オプション
SNIPPET_SIZE	0	オプション	整数値
GROUP_KEY		オプション	

動的優先順位ルール

優先順位ルールは、一致したデータ条件に基づいて追加の属性を開示するかどうかを決定します。実行オプションを使用することで、これらを動的に指定することもできます。

1 つ以上の優先順位ルールの定義を返す SQL 文を指定します。ソース、ターゲットおよびトリガー値の組合せが同じルールが存在しない場合は、新しいルールが下層のサービスによって挿入されます。存在する場合は、そのルールが更新されて有効化されます。

いずれかの必須フィールドが空の場合、サービスが例外をスローし、実行を停止して、問題の内容を明記したメッセージを提供します。

動的優先順位ルール

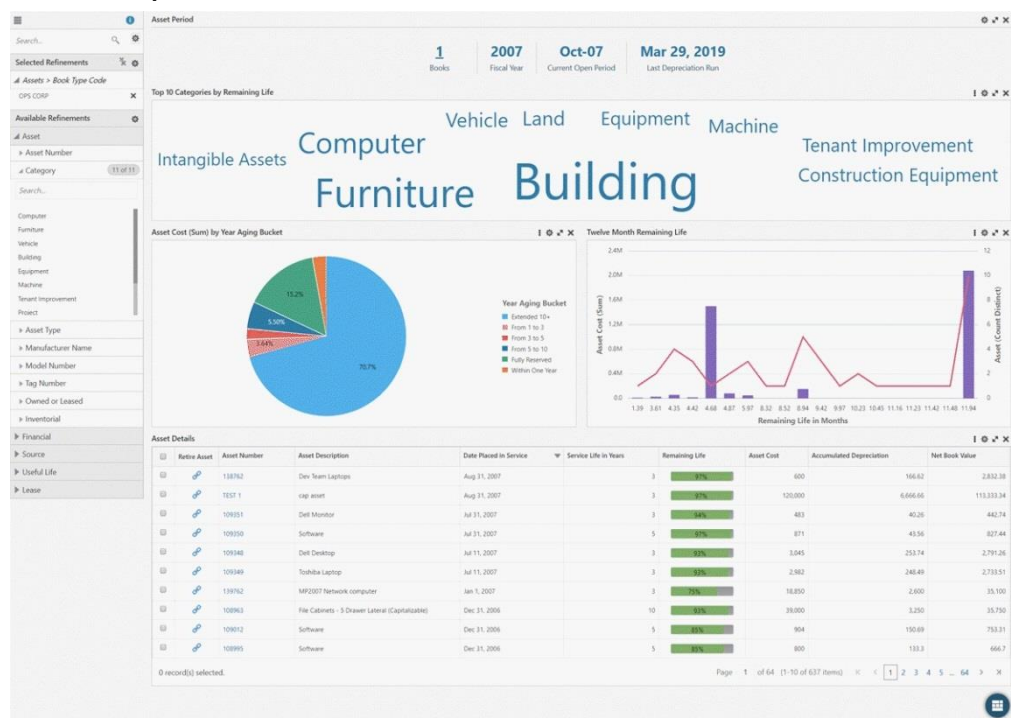
属性名	デフォルト値	必須/オプション	オプション
TRIGGER_INSTANCE_ATTRIBUTE		必須	
TRIGGER_ATTR_VALUE		必須	
TARGET_INSTANCE_ATTRIBUTE		必須	
ENABLED_FLAG		オプション	

Oracle Enterprise Command Center Framework の UI コンポーネントの構造

構成モデル

Oracle Enterprise Command Center Framework ダッシュボードは、複数の UI ビジュアル化コンポーネントの集合体です。

Oracle Enterprise Command Center Framework ダッシュボード



Oracle Enterprise Command Center Framework UI コンポーネントは、4 つの主要グループに分類されます。

Oracle Enterprise Command Center Framework UI コンポーネント・グループ

グループ	コンポーネント
ナビゲーション	検索ボックス 選択した絞込み 使用可能な絞込み
ビジュアル化	要約バー チャート タグ・クラウド 集計表

グループ	コンポーネント
詳細インサイト	結果表 グリッド
レイアウト	タブ付きコンポーネント・コンテナ

Oracle Enterprise Command Center Framework のすべてのコンポーネントは構成エクスペリエンスを向上する統合型構成モデルで提供されます。また、ユーザーに提示されるオプションのサブセットは多くのコンポーネントで共通しています。

構成モデル

Oracle Enterprise Command Center Framework 構成モデルは、SQL 問合せの構造に従った一般的な構造を使用しています。コードやカスタム式を作成することなく、すべての Oracle Enterprise Command Center Framework コンポーネントを構成できます。

Oracle Enterprise Command Center Framework コンポーネントの様々な構成オプションについて説明するため、ここでは SQL 問合せ構造を使用して、Oracle Enterprise Command Center Framework 構成テンプレート内の関連概念を紹介します。Oracle Enterprise Command Center Framework では、SQL 問合せを実行してコンポーネントにデータを表示することはありませんのでご注意ください。

たとえば、次の SQL 問合せについて考えます。

```
SELECT
  SUM(ASSET_COST),
  MAJOR_CATEGORY
```

このメトリックを追加するには、「メトリック」の下で属性"Asset Cost"と集計"Sum"を指定します。

メトリックを定義する例

Metrics

Define the metrics that will determine the series values.

+ Add Metric

Attribute	Aggregation
Asset Cost	Sum

SQL 文

```
FROM
    fa-asset
```

Oracle Enterprise Command Center Framework 構成では、選択されるデータ・セットとして `fa_asset` と表示されます。

データ・セットを選択する例

Select Data Set

fa_asset ▼

SQL 文

```
WHERE
    ASSET_TYPE = 'CAPITALIZED'
```

Oracle Enterprise Command Center Framework 構成では、属性に Asset Type Code、演算子に `=`、値に `CAPITALIZED` を指定した条件を設定します。

条件を設定する例

▲ Conditions (Optional)

Define the conditions (metrics, dimensions, and operators) that determine the chart values.

+ Add New Condition

Logical Operator: ☒ AND ☐ OR

Attribute	Operator	Value
Asset Type Code ▼	= ▼	CAPITALIZ ✕

SQL 文

```
GROUP BY
    MAJOR_CATEGORY
```

Oracle Enterprise Command Center Framework 構成では、ディメンションの下に Category がリストされます。

ディメンションを定義する例

 Dimensions

Define the dimensions for this chart series. Click in the search box to see a list, or enter text to search.

Category 

SQL 文

HAVING

SUM(ASSET_COST) > 10000

Oracle Enterprise Command Center Framework 構成では、属性に Asset Cost (SUM)、演算子に>=、値に 10000 を集計条件の下で設定します。

集計条件を定義する例

 Aggregation Conditions (Optional)

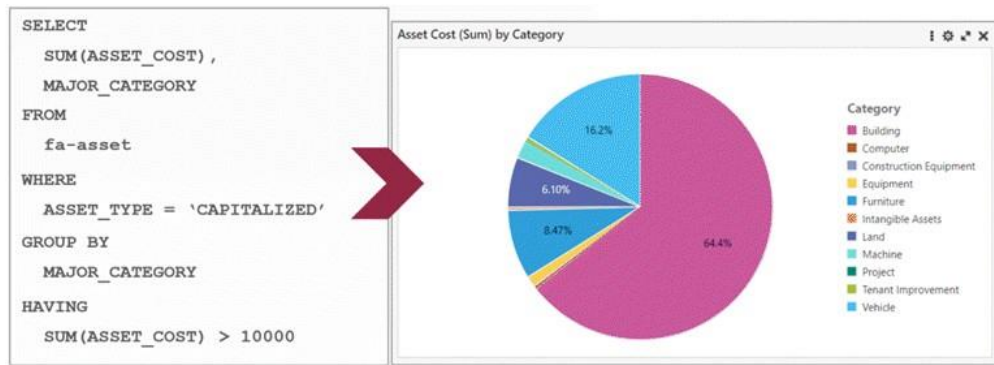
Define additional filters to be applied to the aggregated results.



Attribute	Operator	Value
Asset Cost (Sum) 	>= 	10000 

これらの例に基づいて、次のようなチャートが表示されます。

カテゴリ別の資産取得価額(合計)を表示するチャートの例



共通構成

次に、多くのコンポーネントに共通する構成オプションを示します。

- コンポーネント・タイトル
- データ・セット名
- レコード識別子
- 条件
- ディメンション
- メトリック
- 集計条件
- 表示条件

共通構成のオプション

Chart Configuration

Chart Title
Enter a title for this chart.

Data Set
Assets

Record Identifier
Asset Number

► Conditions (Optional)

► Chart Type

► Dimensions

► Metrics

► Sort Options

► Chart Cascade (Optional)

► Aggregation Conditions (Optional)

► Conditions for Display (Optional)

A. Component Title

B. Data Set Name

C. Record Identifier

D. Condition

E. Dimension

F. Metric

G. Aggregation Condition

H. Condition for Display

► Preview Save Cancel

構成の詳細

次は、様々なコンポーネントのすべての構成オプションについて説明する一覧表です。

構成オプション

構成オプション	必須/オプション	説明	例
Component Title (すべて)	必須	ナビゲーション・コンポーネント 内。タイトルは必須 選択した絞込み 使用可能な絞込み	

構成オプション	必須/オプション	説明	例
Component Title (すべて)	オプション	動的/静的: ビジュアル化コンポーネント内。 動的: デイメンションごとのメトリック属性を表示します(集計関数)。	Chart: Asset Cost (Sum) by Category
Data Set(すべて)	必須	データ・セットの表示名を選択します。	資産
Record Identifier (すべて)	オプション	データ・セット内のエンティティのカーディナリティがグレインと異なる場合に使用します。 レコード識別子を指定しない場合、レコードはそのままの状態が表示されます。 集計に影響します。	資産番号
Condition(all)	オプション	デイメンションとメトリックの両方に適用されます。 条件間には AND または OR が適用されます。	Asset Type Code = 'CAPITALIZED'
Chart Type(チャート)		様々なタイプのチャート(棒/折れ線、円、ドーナツ、散布、バブル)サポートしています。 棒、棒/折れ線および折れ線チャートの表示方法(縦/横)を制御します。 積上げチャートの表示方法(積上げ/非積上げ)を制御します。 ズームを有効化できます。	円
Visualization(タグ・クラウド)		Show Metric Value: タグ・クラウド用語を相対的なウェイトとともに表示します。 Number of items: 表示する用語の数を決定します。	

構成オプション	必須/オプション	説明	例
ディメンション(すべてのビジュアル化コンポーネント)	必須	質的データ(文字列、文字列(複数)および日付属性)。コンポーネントに表示する粒度レベルを定義します。	主カテゴリ、補助カテゴリ
メトリック(すべてのビジュアル化コンポーネント)	必須	量的データ 指定されたディメンションに基づいて集計されます。 集計表内では、 SUM/AVG/MIN/MAX/COUNT DISTINCT および COUNT がサポートされます。 実行時に 1 つのメトリックのみが表示されます(複数メトリック・チャートを除く)。	Asset Cost (Sum)
ソート・オプション(チャート、結果表、グリッド)	オプション	定義されたディメンションまたはメトリックに基づくソートを有効にします。チャートおよび結果表内での実行時ソートを可能にします。	
Cascading(チャート、タグ・クラウド)	オプション	データがディメンション値の単一値まで絞り込まれたときに、コンポーネントが更新され別のディメンションが使用されます。	主カテゴリ、補助カテゴリ
Aggregation Condition	オプション	定義されたメトリックに条件を設定できます。	Asset Cost (Sum) >= 10000
Condition for Display	オプション	コンポーネント上にデータをいつ表示するかを決定します。 1 つのコンテキストにおけるデータを表示するときに使用します。	Book code (count distinct) = 1

注意: Oracle Enterprise Command Center Framework は、コンポーネント構成について設計時に検証を強制しません。無効な構成はエラーとなり、実行時にコンポーネントのかわりに表示されます。構成を見直し、適宜修正してください。

以降の項では、ユーザーが UI コンポーネントを使用して、様々なシナリオをどのように処理して各種のインサイトを取得するかについて説明します。

ナビゲーション・コンポーネント

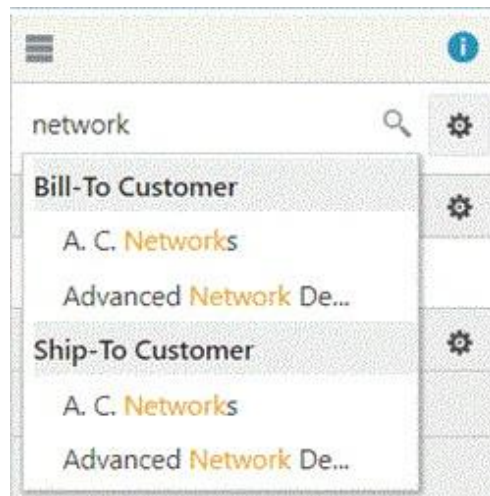
ナビゲーション・コンポーネントを使用すると、分布や特性といった事前知識がなくても、データ内を対話的にナビゲーションできます。また、特定のキーワードや用語を検索したり、フィルタを適用またはリセットしたりできます。

検索ボックス

Oracle Enterprise Command Center Framework では検索機能が提供されるため、これを使用してユーザーは特定のデータ・セット内で用語を検索できます。

基本的な検索を実行するには、検索用語を検索ボックスに入力します。Oracle Enterprise Command Center Framework によって属性値内の一致項目がリストされます。候補リスト内から値を選択すると、Oracle Enterprise Command Center Framework がその属性一致に対して特定の検索を実行します。拡大鏡(検索アイコン)をクリックした場合は、どの属性で検索基準が一致したかに関係なく、その値が含まれるすべてのレコードが取得されます。

検索ボックス

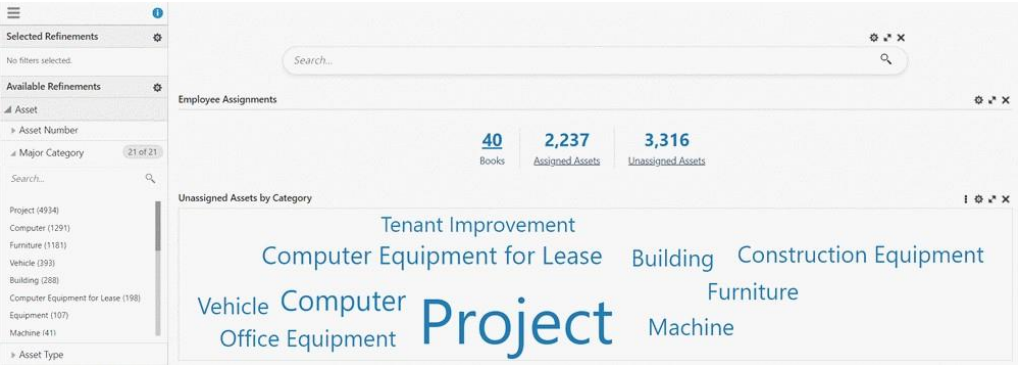


Oracle Enterprise Command Center Framework の検索ボックスでは、ワイルドカード、フレーズ、Boolean 検索といった高度な検索機能も提供します。詳細は、*Oracle E-Business Suite ユーザーズ・ガイド*の Enterprise Command Center の注目機能を参照してください。

検索ボックスをページにドラッグして、新しいコンポーネントとしてページに検索ボックスを追加できます(その逆も可)。検索コンポーネントは Add Components パネルからも追加できます。

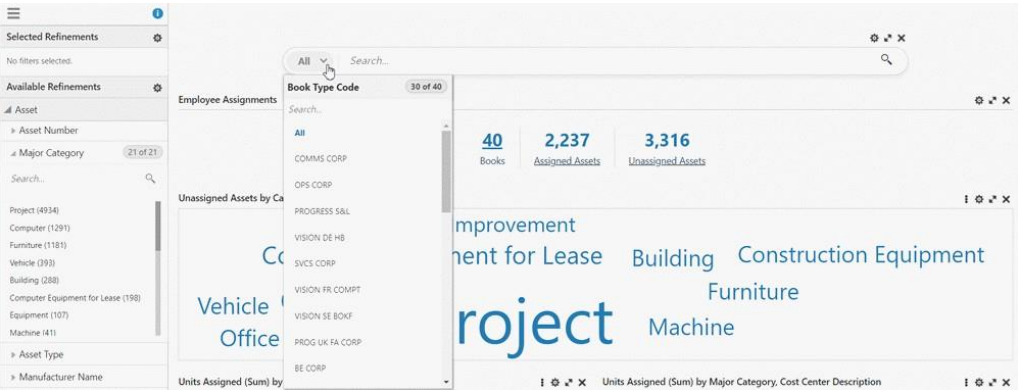
注意: 1 つのページ上に配置できる検索コンポーネントは 1 つのみです。ナビゲーション・パネルまたはページ上に配置できますが、これらの両方の場所に同時に配置することはできません。

ページ上に配置された検索ボックス



検索ボックス機能では検索カテゴリがサポートされるため、特定の属性値内を検索することもできます。

検索ボックス内の検索カテゴリ



構成

検索ボックスの構成オプション

オプション	説明
Data Set	サポートされるデータ・セットは 1 つのみです。

オプション	説明
Place Holder	検索バー内のスペース・ホルダとして使用する文字列を定義します。デフォルト: Search ...。
Show Category Search	ユーザーが属性値を選択して、その値内で検索できるようにします。
Search Category	検索する属性を選択します。実行時には、その属性の値がすべてドロップダウン・リストに表示され、そこからユーザーが値を選択できます。埋込み検索ボックスからも検索できます。
Honor Filter State	このコントロールを無効にすると、現在の絞込みは無視されて、検索で指定された新しい絞込みで上書きされます。このコントロールを有効にすると、適用されている絞込みに基づいて属性のリストが絞り込まれます。
処理	処理タイプを選択します。 Refinement: 検索値を使用して絞込みをページに適用します。 Hyperlink: 検索値で絞込みを適用した後に、指定されたページに移動します。

検索ボックスの構成

Search Configuration

×

Data Set

Assets

Place Holder

Search input place holder

☒ Show Category Search

Search Category

Book Type Code

☐ Honor Filter State

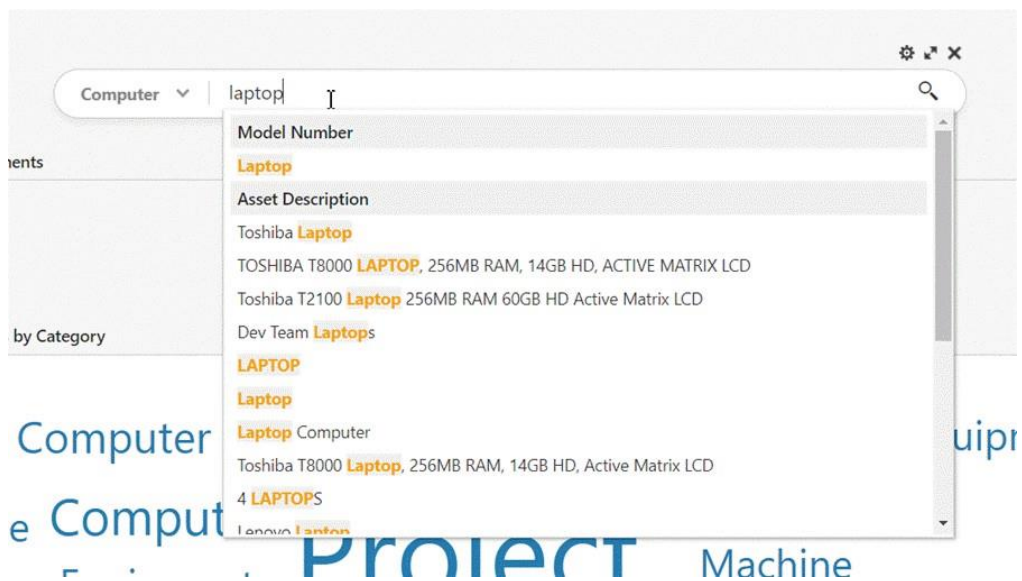
▶ Actions (Optional)

▶ Conditions (Optional)

値検索

カテゴリから属性値を選択し、検索の候補リストからさらに別の属性を選択すると、カテゴリ属性値と選択した属性値の両方が、選択した絞込みの絞込みとして追加されます。

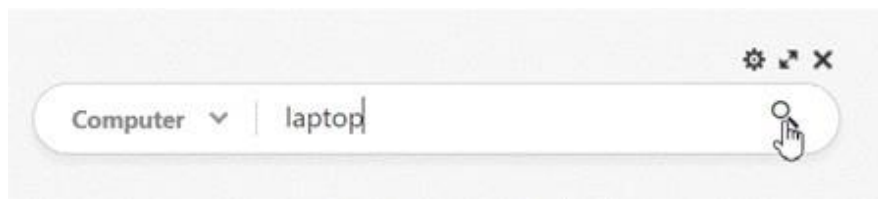
値検索



レコード検索

レコード検索は値検索と同じように動作します。拡大鏡をクリックすると、検索問合せが絞込みとして追加されます。ページの「検索」アコーディオンの下で、検索問合せが絞込みとして適用されます。

検索問合せの例



「検索」アコーディオンの例

Selected Refinements  	
▲ Assets > Major Category	
Computer	
▲ Assets > Your Search	
laptop	

選択した絞り込み

ダッシュボードにフィルタを適用する目的でユーザーが選択した値がすべて、属性名およびデータ・セットごとにまとめて表示されます。絞り可能な Oracle Enterprise Command Center コンポーネント(検索ボックス、タグ・クラウド、チャート)とユーザーが行った対話からフィルタが取得されることがあります。

選択した絞り込みコンポーネントでは連携的なフィルタリング動作も制御できるため、直接または間接的に関連付けられているデータ・セットからの絞り込み状態が反映されるダッシュボードを作成できます。これを行うには、関連付けるデータ・セットを構成で指定します。

選択した絞り込み

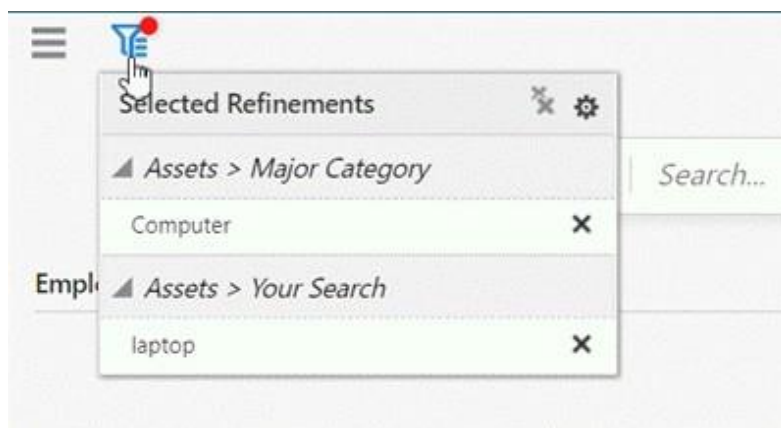
Selected Refinements 	
▲ Assets > Book Type Code	
OPS CORP	
▲ Assets > Employee Name	
Null	
▲ Assets > Category	
Computer	

バージョン 3 の選択した絞り込みはルック・アンド・フィールが異なります。

選択した絞込みのルック・アンド・フィールのオプション

ページ・レイアウト	選択した絞込み
Default (Side Navigation)	ナビゲーション・パネル内に表示されます。 サイド・ナビゲーション・パネルを縮小すると、ファネル・アイコンがページ上部に表示されます。
Side Navigation Collapsed	ファネル・アイコンとして表示されます。 サイド・ナビゲーション・パネルを展開すると、選択した絞込みがページ上部に表示されます。
No Side Navigation	選択した絞込みはページに表示されません。

選択した絞込みのファネル・アイコン

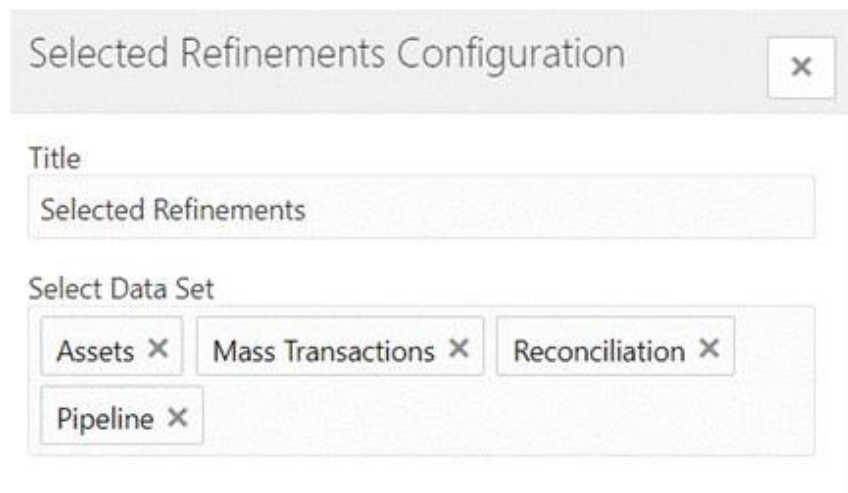


構成

選択した絞込みの構成オプション

オプション	説明
Title	デフォルトのタイトルは、Selected Refinements です
Data Set	複数のデータ・セットを選択して、定義されたデータ・セット間の関連および適用済フィルタを尊重させることができます。

選択した絞込みの構成



Selected Refinements Configuration

Title

Selected Refinements

Select Data Set

Assets X Mass Transactions X Reconciliation X

Pipeline X

使用可能な絞込み

使用可能な絞込みでは、分布や特性といった事前知識がなくても、データ内を対話的にナビゲーションできます。

ユーザーが使用可能な絞込みコンポーネントと対話したり、またはダッシュボード上の別のコンポーネント上でフィルタリング操作を実行したりすると、使用可能な絞込みの属性リストが動的に更新され、関連する属性と属性絞込みが表示されます。このナビゲーションはデータ主導型で、ユーザーのデータ内でのナビゲーション・パスに従って追加の属性を徐々に開示します。また、属性値リスト内に表示されるデータには絞込み状態が反映され、ユーザーが探索しているデータのサブセットに基づいて、縮小、展開または除外されます。

使用可能な絞込みでは、グループ化リストを使用して属性を整然と表示することができます。

使用可能な絞り込み

Available Refinements

Invoice

Invoice Type 5 of 6

Search...

Standard

Prepayment

Credit Memo

Expense Report

Debit Memo

Mixed

Invoice Number

Validation Status

Payment Status

Invoice Currency

Invoice Source

Purchase Order Number

構成

使用可能な絞り込みの構成オプション

オプション	説明
Title	デフォルトのタイトルは、Available Refinements です。
Data Set	サポートされるデータ・セットは 1 つのみです。

オプション	説明
Attribute List	Display Attribute Groups: このチェック・ボックスを使用して表紙する属性グループを選択できます。 属性リスト: コンポーネントに表示する属性と、その表示方法(リスト/範囲)を制御できます。これらのコントロールは属性のフラット・リストと属性グループの両方でサポートされます。

使用可能な絞込みの構成

The image displays two versions of the 'Available Refinements Configuration' dialog box. Both versions have a title bar with a close button (X). The 'Title' field contains 'Available Refinements' and the 'Data Set' dropdown is set to 'Assets'.

Left Screenshot: The 'Attributes List' section is expanded. The 'Display Attribute Groups' checkbox is checked. Below it is a 'Select Group' dropdown and a green '+ Add Group' button. A list of attributes is shown, each with a minus sign icon and an 'X' icon to its right: Asset, Financial, Source, Useful Life, Lease, and Additional Details. Below this list is a section for 'Conditions (Optional)'.

Right Screenshot: The 'Attributes List' section is expanded. The 'Display Attribute Groups' checkbox is unchecked. Below it is a 'Sort Attributes' button and a search filter 'Filter Attributes by Name or type'. A list of attributes is shown, each with a plus sign icon and a checkmark icon to its right: Book Type Code, Asset Number, Ledger, Asset, ECC_SPEC_ID, and Book Type Name.

Both versions have a 'Preview' button (green), a 'Save' button (blue), and a 'Cancel' button (grey) at the bottom.

Like 検索

使用可能な絞込みでは、文字列タイプの属性と、演算子'*'/'%(0 個以上の文字用)や'?'(1 文字用)を使用した検索問合せで **LIKE** 検索をサポートしているため、部分一致や完全一致の検索を実行できます。検索結果は選択した絞込みの絞込みとして、即時適用されます。

拡大鏡をクリックすると、検索問合せに部分検索演算子が含まれていない場合でも、**LIKE** 検索が実行されます。演算子が指定されていない場合、部分検索演算子が検索問合せの末尾に暗黙的に追加されます。

使用可能な絞り込み内の Like 検索

Available Refinements

Asset

Asset Number

27 of 27

1010%

101012

101013

101014

101007

101008

101040

101041

101042

Manufacturer Name

選択した絞込み内の Like 検索

Selected Refinements  	
▲ Assets > Major Category	
Computer	×
▲ Assets > Your Search	
laptop	×
▲ Assets > Asset Number	
Like 1010%	×

ビジュアル化コンポーネント

Oracle Enterprise Command Center Framework にはグラフとチャートのセットが含まれており、これらの強力な機能を使用して、意思決定に必要なデータを要約して表示できます。インサイトを取得したり、外れ値を検出したり、チャート上で直接データをフィルタリングしたりできるほか、より深いレベルの詳細にドリルダウンできます。

要約バー

要約バーには、ダッシュボード対象ビジネス・エリアの重要側面を要約したメトリックやディメンションの値が表示されるため、特定のビジュアル・エリアに関する基礎固めに役立ちます。

要約バー内のエントリは次の表に示すタイプのいずれかです。

要約バーのエントリ・タイプ

要約項目	説明
メトリック	特定のメトリックの値(売上合計や平均利益)を表示します。 宛先のダッシュボードやタブにナビゲートするときに使用でき、オプションで絞込み処理を呼び出せます。

要約項目	説明
ディメンション	関連付けられているメトリック値の最高値または最低値のいずれかに関連付けられているディメンション値。売上合計が最も高い製品カテゴリなど。
フラグ	フラグには構成されているディメンションの数が表示されます。 追加のメトリックがポップアップに表示されます。

要約バー

Open Payables						
1 Ledgers	USD Functional Currency	11,006,513.05 Supplier Balance	59,656.26 Employee Balance	63,000.00 Unapplied Prepayment	11,048,919.31 Past-Due Invoices	10,000.00 Invoices Due in 7 Days

構成

次に、一般構成オプションと項目特有構成オプションを示します。

一般構成

要約バーの一般構成のオプション

オプション	説明
+ Add Summary Item	+ Add Summary Item の横にあるドロップダウン矢印をクリックして、追加する要約項目のタイプをリストから選択します。
Title	要約項目のタイトル
Data Set	データ・セットを選択します。

注意:

- 要約項目の表示順序を指定するには、リスト内で各項目をドラッグして目的の順序の位置に移動します。
- 要約項目を削除するには、項目の'x'アイコンをクリックします。

要約バーの構成

Summarization Bar Configuration

Summarization Title

Asset Period

+ Add Summary Item

Books (flag)	—	×
Fiscal Year (dimension)	—	×
Current Open Period (dimension)	—	×
Last Depreciation Run (metric)	—	×
Total Assets (metric)	—	×

Summary Item Title

Total Assets

Data Set

Assets

Record Identifier

Asset Number

Define Conditions (Optional)

Select Metric

Conditions for Display (Optional)

Visualizations

Actions

Preview Save Cancel

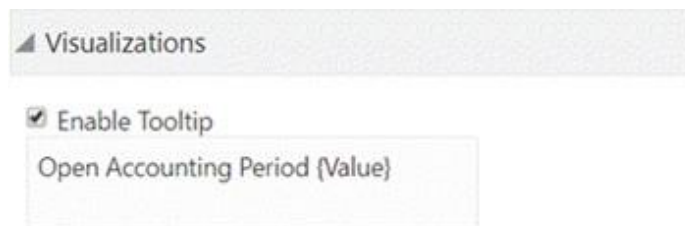
項目特有の構成

ビジュアル化:

ビジュアル化のオプション

オプション	説明
Enable Tooltip	要約項目を説明するツールチップを追加できます。
Tooltip Text	値を表示するには{Value}トークンを使用します。

ビジュアル化構成オプション



条件付き書式設定

- メトリック項目でのみサポートされます。
- 特定の条件に基づいて別の色でメトリック値を表示できます。

条件付き書式設定のオプション

オプション	説明
Add Condition	1 つ以上の条件を追加します。
Formatting	• 条件が一致したときに表示する色を選択します。 • 使用する数値演算子を選択します。 • 値フィールドに適切な値を入力します。 • between (b/w)演算子を使用する場合は、最小値と最大値を入力します。 • 条件を削除するには Delete アイコンを使用します。

条件付き書式の構成

Conditional Formatting

Define conditions for visually highlighting the summary item.

+ Add Condition

Display

When value is..

<

5

×

実行時の条件付き書式



処理

- メトリック項目でのみサポートされます。
- メトリック値をクリックして、別のダッシュボードに移動したり、絞り込み処理を適用したりできます。

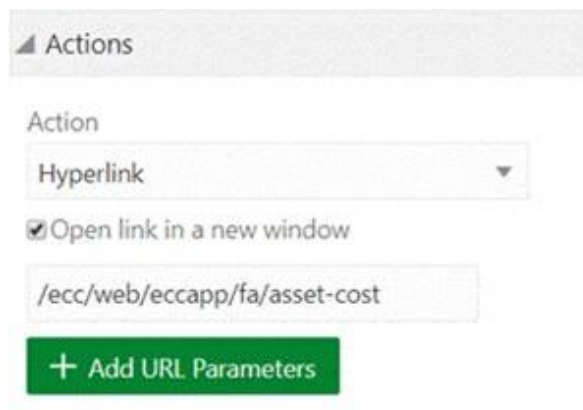
処理オプション

オプション	説明
処理	処理タイプ・リストから処理を選択します。

処理オプション

オプション	説明
Hyperlink 処理	- 値をクリックすると、ユーザーは指定された URL にナビゲートされます。 - ハイパーリンクには同じアプリケーション内の別のダッシュボードまたは外部 URL を指定できます。 - URL にはパラメータとして属性値を含めることができます。 - 新しいブラウザ・ウィンドウを開いて宛先のダッシュボードを表示するには、Open link in a new window チェック・ボックスを選択します。 - URL フィールドにハイパーリンクの宛先を入力します。 - 外部 URL に属性値を追加するには次を実行します(オプション)。 + Add URL Parameters をクリックします。 - 属性リストからパラメータ属性を選択します。 - 属性が URL にパラメータとして挿入されます。パラメータ名は属性キーで、パラメータ値は{ID}で ID はその属性の ID です。

Hyperlink 処理の構成



フラグ条件

- フラグ条件を使用して、フラグ要約項目を表示するときの条件を追加します。

- 定義された条件の少なくとも 1 つが満たされた場合にのみ、フラグ要約項目がエンド・ユーザー・ビューに表示されます。

フラグ条件オプション

オプション	説明
+ Add Condition	• 事前定義のメトリックを選択します • 演算子 • 値

フラグ条件の構成

▲ Flag Conditions (Optional)

Define flag conditions.

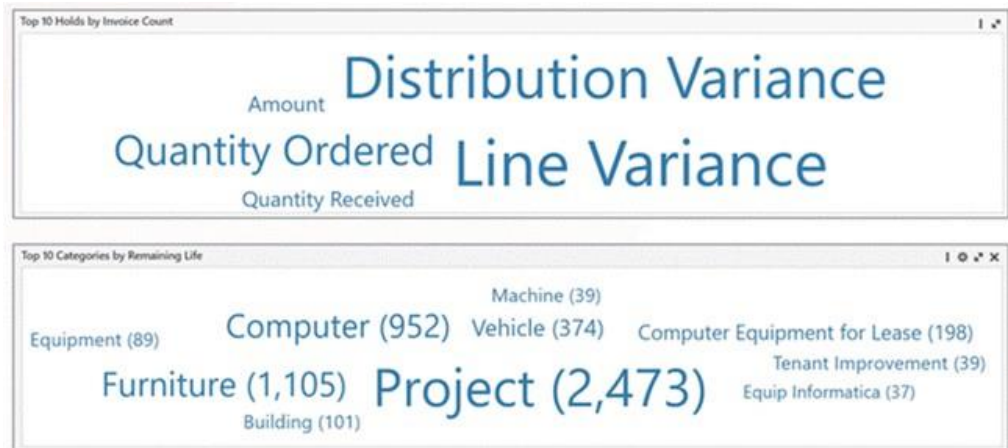
+ Add New Condition

Attribute	Operator	Value
Count of Asset N...	>	100

タグ・クラウド

タグ・クラウド・コンポーネントでは、表示されている複数の用語をそれらに関連付けられているメトリックの値に基づいて比較できます。オプションで、各用語に関連付けられているメトリック値をコンポーネントに表示できます。

タグ・クラウドの例



構成

次に、タグ・クラウドの構成情報を示します。

一般構成

タグ・クラウドに表示される用語は、選択したディメンションからの値です。エンド・ユーザーに選択を許可するディメンションのリストを構成できます。

選択されたメトリックの値によって、タグ・クラウドの用語の相対的なサイズが決まります。

タグ・クラウドの構成

Tag Cloud Configuration

Tag Cloud Title
Top 10 Categories by Remaining Life

Data Set
Assets

Record Identifier
Select record identifier for the data set

Conditions (Optional)

Visualization

Dimensions

Define the Dimensions (by clicking on the input below or typing text inside it to search) that will determine the Tag Cloud items

Category

Category

Metrics

Tag Cloud Cascade (Optional)

Aggregation Conditions (Optional)

Conditions for Display (Optional)

Preview

Save

Cancel

特有構成

次に、タグ・クラウドの特有構成オプションを示します。

ビジュアル化

タグ・クラウドの追加の表示構成オプションを提供します。

タグ・クラウドのビジュアル化オプション

オプション	説明
Show Metric Value	このオプションを有効にすると、各タグ・クラウド用語に計算済メトリック値が表示されます。

オプション	説明
Number of Items	タグ・クラウドに表示する最大用語数を入力または選択します。

タグ・クラウドのビジュアル化構成



タグ・クラウドのカスケード

タグ・クラウドのカスケード・オプション

オプション	説明
+ Add Cascading	• +Add Cascading Item をクリックします。 • 最初のディメンション(カスケードの最上位レベル)を選択します。 • さらにカスケード・レベルを追加するには、+Add Cascading Item をクリックして、次レベルのディメンションを選択します(新しいディメンションがカスケードの最後に追加されます)。 注意: カスケード・レベルを追加するには、すべてのディメンションをディメンション・セクションで再定義する必要があります。

タグ・クラウドのカスケード構成

▲ Tag Cloud Cascade (Optional)

Define the Cascade levels for the Tag Cloud

+ Add New Cascade item

1. Category ▼ ×

2. Category ▼ ×

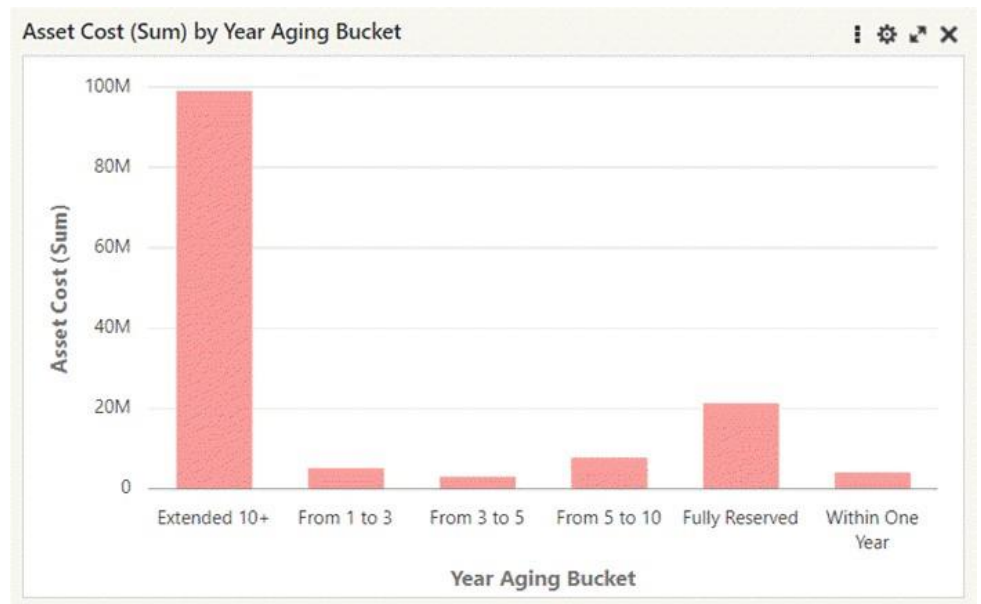
チャート

チャート・コンポーネントには、アプリケーション・データに基づいた視覚的なチャートが表示されます。数種類のサブタイプをサポートしており、特定のデータを選択して表示するオプションも含まれています。

チャートには次のサブタイプがあります。

- 棒グラフ
系列ディメンションまたはグループ・ディメンションにわたって集計した、1 つ以上のメトリック値を表示します。

棒グラフの例



1 つ以上の値を正確に比較する場合に便利です。

- 棒/折れ線グラフ

系列ディメンションにわたって集計した、2 つのメトリック値を表示します。

棒折れ線グラフの例

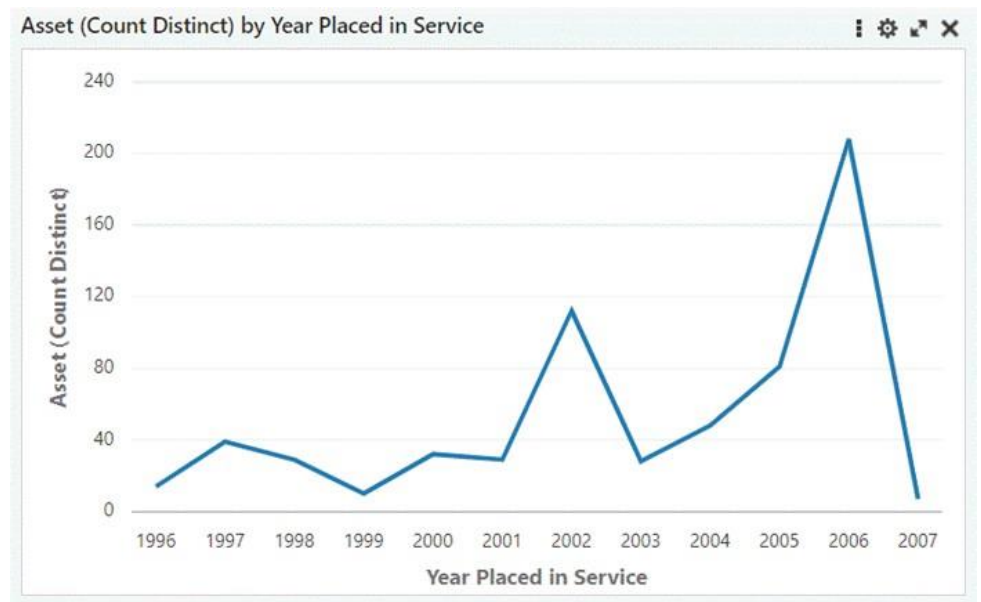


数量とその経時的な変動傾向を一緒に表示する場合に便利です。

- 折れ線グラフ

系列ディメンションにわたって集計した、1 つ以上のメトリック値を表示します。

折れ線グラフの例

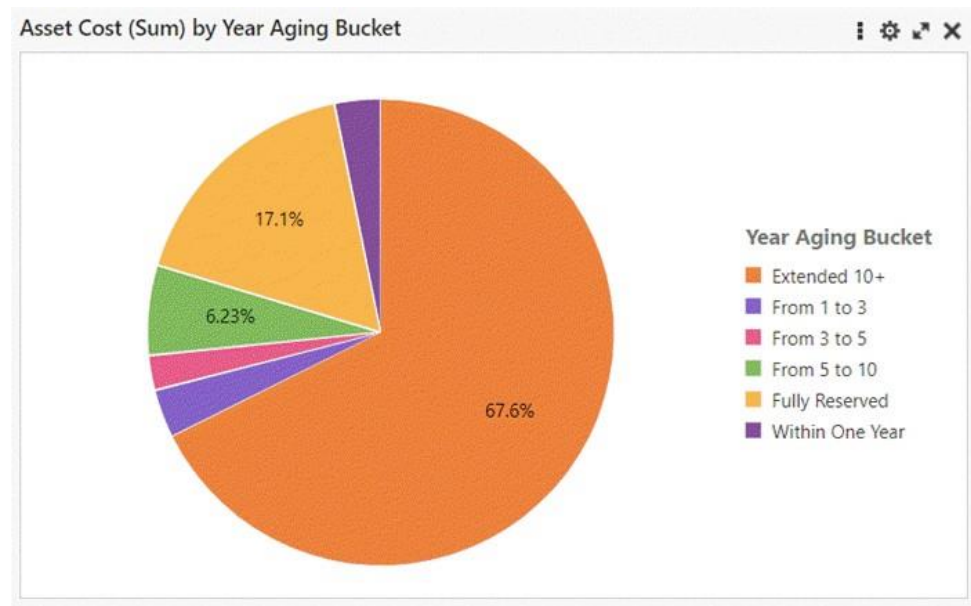


変動や傾向を表示する場合に便利です。

- 円/ドーナツ・グラフ

系列ディメンションの各値毎に集計した、1つのメトリックを表示します。

円グラフの例

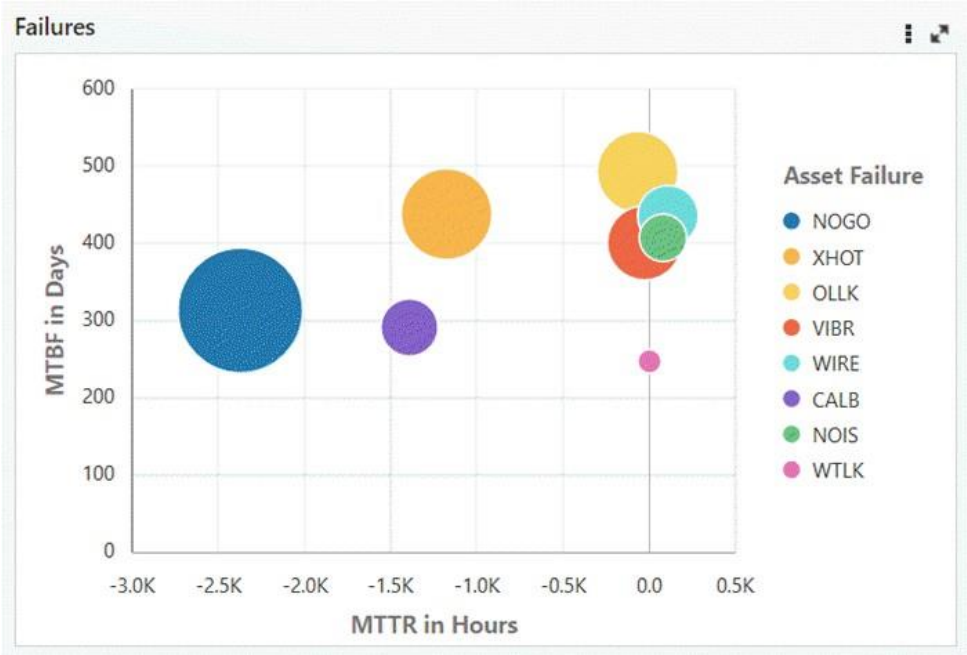


全体に対する各値の割合を表示する場合に便利です。

- 散布図/バブル・チャート

データ・ポイントを表示します。各データ・ポイントが 1 つのディメンション値を表します。ビン化散布図ではデータ・ポイントをビンに集約でき、バブル・チャートではデータ・ポイントを 3 つ目のメトリックに応じた大きさでバブル表示します。

バブル・チャートの例



これらのチャートはメトリック間の相関性を表示する場合に便利です。

7.3.3.1 構成

次に、チャートの構成オプションを示します。

7.3.3.1.1 一般構成

一般構成オプション	
オプション	説明
Title	
Data Set	チャートに関連付けるデータ・セットを選択します。
Chart Type	チャート・タイプを選択します。
ディメンション	系列ディメンション: 次の表内のほとんどのチャート・タイプで必須(M)です。

チャート・ディメンション

タイプ	グループ・ディメンション	系列ディメンション
棒		(M) カテゴリ軸
積上げ棒	カテゴリ軸	(M) 色
複数メトリック棒		(M) カテゴリ軸
折れ線		(M) カテゴリ軸
棒/折れ線		(M) カテゴリ軸
バブル	(M) 色	バブルの数
散布図	(M) 形と色	形の数
円/ドーナツ		(M) 扇形の色

チャート・ディメンションの構成

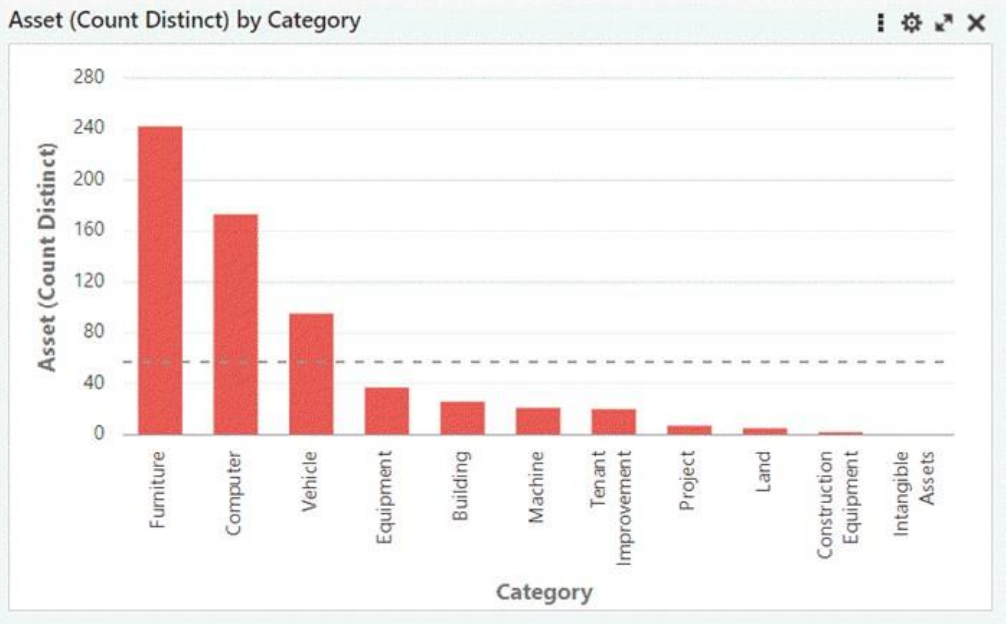
Dimensions

Define the dimensions for this chart series. Click in the search box to see a list, or enter text to search.

Category ×

Cost Center Description ×

参照線のあるチャートの例



チャートのオプション

オプション	説明
メトリック	
系列メトリックの表示 (「単一」または「複数」)	チャート上にすべてのメトリック値を一度に表示するか、またはユーザーが実行時のオプションを通じて制御するのを決定します。
Allow Multi Metric Option	ビジネス・ユーザーが実行時に、チャートの単一メトリックと複数メトリックを切り替えられるようにします。 このオプションはデフォルトで選択されています。
Show As Line	棒グラフを折れ線グラフに変換します。

オプション	説明
+ Add Metric	メトリック属性と、次に示す集計関数を選択します。 - 合計: メトリックの合計値を計算します。 - 平均: メトリックの平均値を計算します。 - 最小: メトリックの最小値を選択します。 - 最大: メトリックの最大値を選択します。 - 個別の件数: 属性の値の件数を数えます。
+ Add Y Axis Reference Line	- 参照線: 事前定義のメトリックに基づいてチャートに参照線を表示します。 例: 資産取得価額(合計)の平均 - チャートで定義されているメトリックの数だけ参照線を追加できます。 - チャートにメトリックが 1 つしか表示されていない場合、このメトリックに対応する参照線が表示されます。 - 平均、最小値、最大値、定数がサポートされています。 - 円/ドーナツ・チャート以外のすべてのチャート・タイプに適用されます。

チャート・メトリック計算

Metrics

Show Series Metric ☒ Single ☐ Multi

☒ Allow Multi Metric Option

☐ Show As Line

Define the metrics that will determine the series values.

+ Add Metric

Attribute	Aggregation	
Asset Number	Count Dis...	✖
Asset Cost	Sum	✖

Y-Axis Reference Lines

+ Add Y-Axis Reference Line

Asset Number(Count...	Average	✖
Asset Cost(Sum)	Average	✖

ソート・オプション

- ソート順およびソート属性を決定します。ソート・キーにはディメンションまたはメトリックを指定できます。アルファベット順のソートはディメンションに適用され、数値はメトリックに適用されます。
- エンド・ユーザーのソートを許可します。エンド・ユーザーが実行時にデフォルトのソートを変更できるようにします。

チャートのソート・オプション

Sort Options

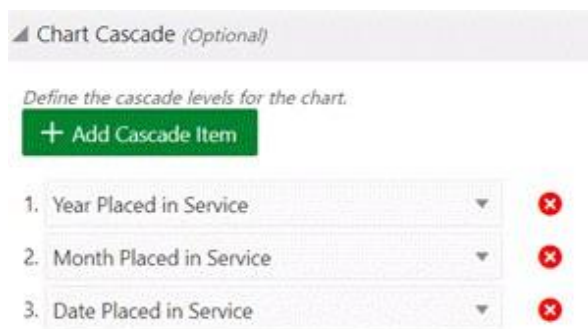
Metric Descending

☒ Allow end-user sorting

チャートのカスケード・オプション

オプション	説明
+Add Cascading Item	最初のディメンション(カスケードの最上位レベル)を選択します。 さらにカスケード・レベルを追加するには、+Add Cascading Item をクリックして、次レベルのディメンションを選択します(新しいディメンションがカスケードの最後に追加されます)。 注意: カスケード・レベルを追加するには、すべてのディメンションを系列ディメンション・セクションで再定義する必要があります。

チャートのカスケード構成



集計表

集計表では興味のあるディメンションの集計を表形式で表示できます。表示されるメトリックは設計時に定義した集計結果です。

集計表の実行時オプション

Balances			
☐	SLA Account Description	Subledger Balance	General Ledger Balance
☐	Asset Clearing	-599,189	-221,580
☐	Computers and Software	28,199	1,980
☐	Furniture	63,082	118,640
☐	Vehicles	277,121	22,370
0 record(s) selected.			
Page 1 of 1 (1-4 of 4 items) < 1 >			

ビジネス・ユーザーは集計表に表示するデータを実行時オプションを通じて変更できます。

集計表のデータを調整する実行時オプションの例

Balances			
☐	SLA Account Description	Subledger Balance	
☐	Asset Clearing	-599,189	
☐	Computers and Software	28,199	
☐	Furniture	63,082	
☐	Vehicles	277,121	
0 record(s) selected.			
Balance Details			
☐	Asset Number	Asset Description	SLA Account
☐	107851	Computer	1570
0 record(s) selected.			

Hide/Show Attributes

Note: Aggregations displayed will be affected based on your selection

Attributes

- SLA Account Description
- SLA Cost Center Description
- Major Category
- Minor Category

Metrics

- Subledger Balance
- General Ledger Balance

Update Reset

7.3.4.1 特有構成

集計表の特有構成オプション

オプション	説明
属性の追加	+ Add Attribute をクリックします。 属性リストにはメトリック集計に使用するディメンションが含まれています。ディメンション列は表の左側に常時表示されます。
メトリックの追加	+ Add Metric をクリックします。 メトリック属性には属性タイプに応じたデフォルトの集計関数が指定されています。 デフォルトの集計を変更できます。

集計表の属性の構成

Attributes

Select Attributes ▼ + Add Attribute

▶ SLA Account Description	—	✕
▶ SLA Cost Center Description	—	✕
▶ Major Category	—	✕
▶ Minor Category	—	✕

Select Metric ▼ + Add Metric

▶ Sub Ledger Balance	—	✕
▶ General Ledger Balance	—	✕

詳細インサイト・コンポーネント

Oracle Enterprise Command Centers では結果表やグリッドを使用して詳細なインサイトを入力できます。そのためユーザーは適切な処理を実行して、プロセスのボトルネックや例外の解決、ビジネス・トランザクションの改善を図ることができます。

結果表

結果表コンポーネントにはデータのセットが表形式で表示されます。結果表コンポーネントに表示されるデータは次のいずれかです。

- 選択したデータ・セットからのレコードのフラット・リスト。各行が 1 件のレコードを表します。列にはそのレコードの属性値が表示されます。

結果表 - フラット・リスト

Asset Details								
☐ Retire Asset	Asset Number	Asset Description	Date Placed in Service	Service life in Years	Remaining In Year	Asset Cost	Accumulated Depreciation	N
☐	109351	Dell Monitor	07/31/2007	3	100%	483	13.42	
☐	109350	Software	07/31/2007	5	100%	871	14.52	
☐	109349	Toshiba Laptop	07/11/2007	3	98%	2,982	82.83	
☐	109348	Dell Desktop	07/11/2007	3	98%	3,045	84.58	
☐	108997	File Cabinets - 5 Drawer Lateral (Capitalizable)	12/31/2006	10	94%	39,000	2,600	
☐	108947	Toshiba Laptop	12/31/2006	3	80%	3,016	670.23	
☐	109006	Company Car	12/31/2006	4	85%	37,211	6,201.83	
☐	108948	Dell Monitor	12/31/2006	3	80%	477	105.99	
☐	108994	Toshiba Laptop	12/31/2006	3	80%	2,463	547.35	
☐	108968	Company Car	12/31/2006	4	85%	35,386	5,897.67	

0 record(s) selected Page 1 of 63 (1-10 of 627 items) < 1 2 3 4 5 ... 63 >

- 属性をグループ化したリスト。各グループは、属性を機能的または論理的にグループ化したものです。

結果表 - 属性グループ

Assets							
Financial							
☐ Update Asset	Asset Number	Asset Description	Date Placed in Service	Asset Cost	Accumulated Depreciation	Net Book Value	
☐	108292	MP2006 Network computer	Jan 1, 2006	720	6,720	4,800	
☐	100918	PC	Aug 1, 1997	2,500	12,500	0	
☐	102549	PULSE OXIMETER MODEL 8500	Feb 28, 2002	2,200	2,200	0	
☐	108777	Dell Monitor	Jul 14, 2006	483	201.28	281.72	
☐	108826	Toshiba Laptop	Jul 31, 2006	2,912	1,213.34	1,698.66	
☐	108074	File Cabinets - 5 Drawer Lateral (Capitalizable)	Jan 10, 2006	39,000	6,825	32,175	
☐	139762	MP2007 Network computer	Jan 1, 2007	18,850	2,600	35,100	
☐	108557	Company Car	May 31, 2006	37,211	13,178.9	24,032.1	
☐	108963	File Cabinets - 5 Drawer Lateral (Capitalizable)	Dec 31, 2006	39,000	3,250	35,750	
☐	102541	PULSE OXIMETER MODEL 8500	Feb 28, 2002	2,200	2,200	0	

0 record(s) selected. Page 1 of 63 (1-10 of 629 items) < 1 2 3 4 5 ... 63 >

結果表の構成

次に、構成オプションを示します。

共通構成

オプション

説明

Title

Component title。

Data Set

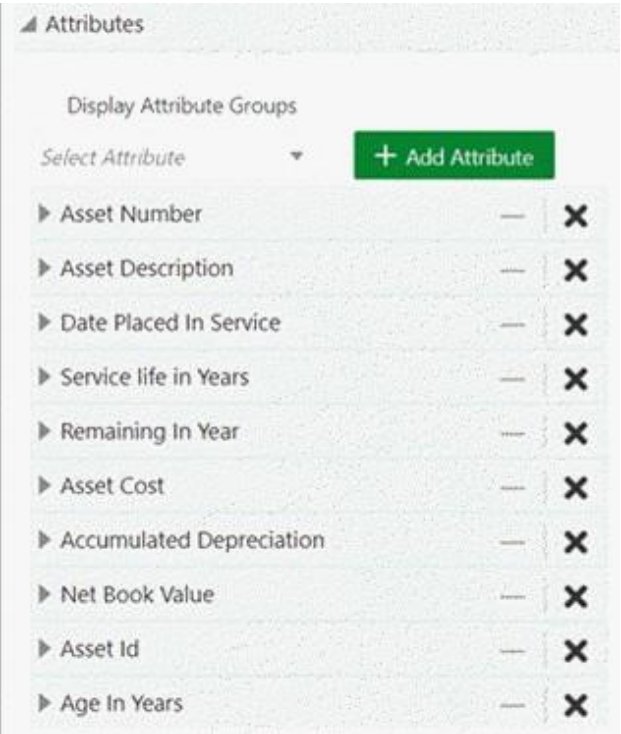
データ・セットを 1 つ選択します。

フラット・リストの構成

フラット・リスト構成オプション

オプション	説明
属性	属性を選択して、+ Add Attribute をクリックします。

結果表 - フラット・リスト構成



属性グループの構成

属性グループ構成オプション

オプション	説明
Display Attribute Group	Display Attribute Group を選択すると、結果表に属性グループを表示できます。

オプション	説明
Persistent Attributes	Persistent Attributes セクションで、+ Add Attribute をクリックします。 - 属性の表示順序を設定するには、各属性をリスト内の適切な場所にドラッグします。リストの最上部にある列が表の左端に表示されます。 - Persistent Attributes リストから列を削除するには、その列の削除アイコンをクリックします。
Interchangeable Attributes	Interchangeable Attributes セクションで、+ Add Group をクリックします。 属性グループを追加します。 - 属性グループの名前は変更できません。 - 属性グループおよびグループ内の属性の順序を変更できます。

結果表 - 属性グループの構成

The screenshot displays the 'Attributes' configuration window. At the top, there is a checkbox labeled 'Display Attribute Groups' which is checked. Below this, the interface is divided into two main sections: 'Persistent Attributes' and 'Interchangeable Attributes'.

Persistent Attributes: This section has a dropdown menu labeled 'Select Attribute' and a green button labeled '+ Add Attribute'. Below these are three rows of attributes: 'Asset Number', 'Asset Description', and 'Asset'. Each row has a right-pointing arrow icon and a red 'X' icon for deletion.

Interchangeable Attributes: This section has a dropdown menu labeled 'Select Group' and a green button labeled '+ Add Group'. Below these are three rows of attribute groups: 'Asset', 'Financial', and 'Source'. Each row has a right-pointing arrow icon and a red 'X' icon for deletion.

特有構成

処理 - 行処理

+ Add Action と、処理タイプとして Hyperlink を選択します。

- 処理名フィールドに、処理の名前を入力します。
- ハイパーリンクをアイコンとして表示するには、Display Default Icon チェック・ボックスを選択します。
- ハイパーリンクを別のブラウザ・ウィンドウに表示するには、Open link in a new window チェック・ボックスを選択します。
- URL フィールドに、リンク先の URL を入力します。
- 属性値を URL に追加するには、次の手順を実行します。
 - Add URL Parameters をクリック
 - 属性名を選択
- 条件付き処理表示を追加するには、次の手順を実行します。
 - 属性を選択
 - 演算子
 - 値
 - 処理の有効化/無効化

行処理の構成

Update Asset

Action Type
Hyperlink

Action Name
Default Action

Action Column Header
Update Asset

☒ Display default icon
☒ Open link in a new window

/ecc/ebs/formsCaller?functionCode=FF

+ Add URL Parameters

Attribute
Asset

Conditional Action Display
If Select Attri... Sel... Enter Vai Then
Enable

/ecc/ebs/formsCaller?functionCode=FAXASSET&DP_ASSET_ID={0}

処理 - 処理メニュー

+ Add Action と、処理タイプとして Pass Parameters を選択します。

- 処理名フィールドに、処理の名前を入力します。
- ハイパーリンクを別のブラウザ・ウィンドウに表示するには、Open link in a new window チェック・ボックスを選択します。
- URL フィールドに、リンク先の URL を入力します。
- 属性値を URL に追加するには、次の手順を実行します。
 - Add URL Parameters をクリック
 - 属性名を選択
- 条件付き処理表示を追加するには、次の手順を実行します。
 - 属性を選択

結果表の処理メニューの構成

Action

Pass Parameters ▼

Action

Print

☐ Open link in a new window

/OA_HTML/IARECCTableActionsPost.jsp

+ Add URL Parameters

Attribute

Account Number ▼ ×

Transaction Number ▼ ×

Installment ▼ ×

Payment Schedule Id ▼ ×

Transaction ▼ ×

Conditional Action Display

If Select Attr... ▼ Se... ▼ Enter Va Then

Enable ▼

ソート

- 表で使用するデフォルトのソート順をユーザーが構成できるようにします。
- 新しいソート・ルールを追加するには、+ Add Sort をクリックします。
- 属性と、「昇順」または「降順」のいずれかのソート・オプションを選択します。

結果表のソート構成

▲ Sorting (Optional)

Define the default sort order for the table

+ Add New Sort

1. Asset Number ▼ Ascending ▼ ×

グリッド

グリッド・コンポーネントを使用すると、レコードのリストを構成可能な任意のレイアウトに表示して、レコードの相違要素をわかりやすく表示できます。長いテキストの値や画像、インジケータ・アイコン、ボタン、スター評価などを表示する強力な機能を備えています。

特有構成

グリッドの構成オプション

オプション	説明
Layout Template	- 複数の行および列で構成され、セルは任意の数の列/行スパンをサポートします。 - 同一セル内に複数の項目タイプ(ディメンションおよびメトリック属性値、入力テキスト、カスタム・ボタン、スター評価、視覚的なインジケータ・アイコン)を表示できます。 - レイアウト・テンプレートを定義するには、グリッドの列数と行数を設定します。 Preview をクリックして、レイアウト・テンプレート内の各セルの構成を開始します。
セルへの項目の追加	+Add item ドロップダウン・リストをクリックし、次のいずれかの値を選択します。 - Attribute values - Button - Text Input - Indicator Icon

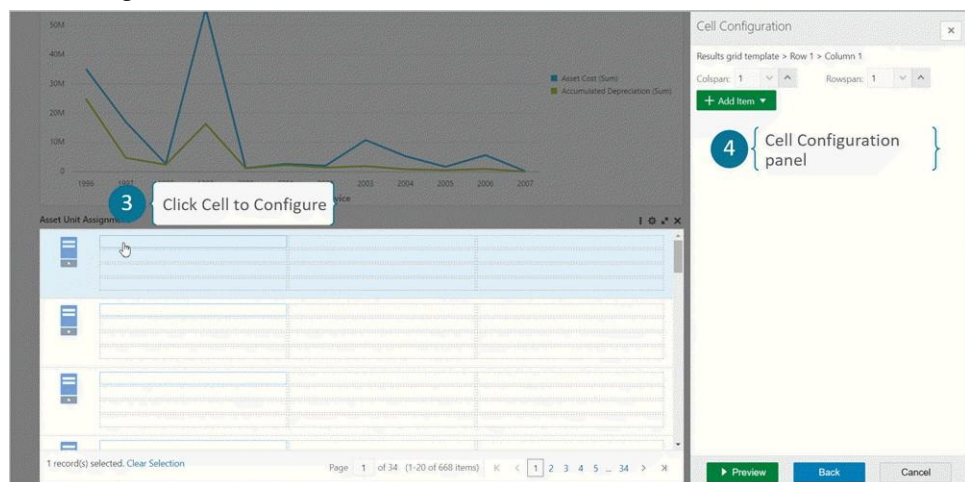
1. グリッドの列数と行数を設定します。

列数と行数の設定



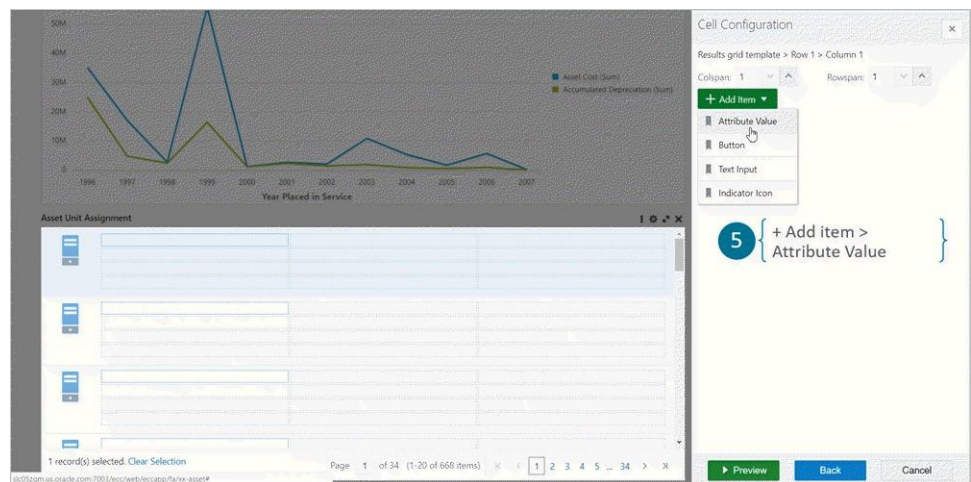
2. Preview をクリックします。
3. セルをクリックして構成します。
4. Cell Configuration パネルが表示されます。

Cell Configuration パネル



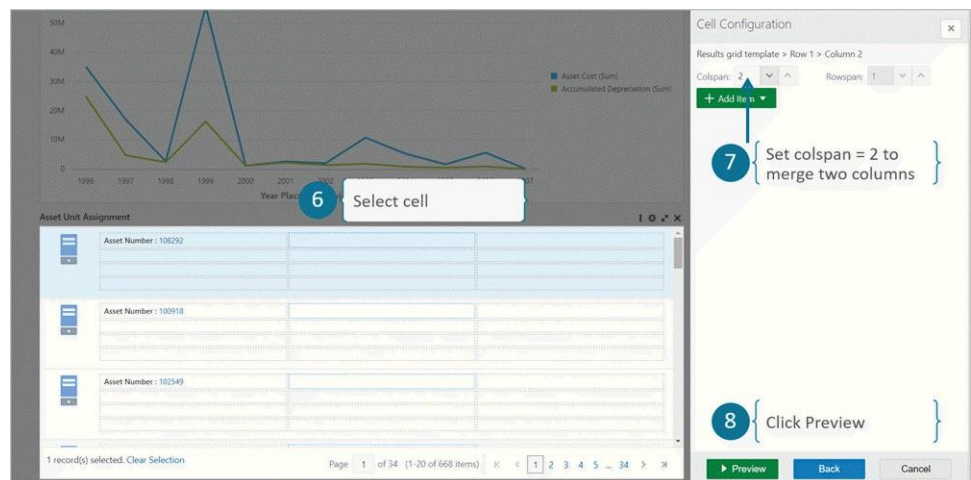
5. + Add Item をクリックし、Attribute Value を選択します。

属性値の選択



6. セルを選択します。
7. Colspan を 2 に設定し、2 列をマージします。

セルの 2 列をマージ



8. Preview をクリックします。
9. Show label の設定を、ラベルを表示しない設定に切り替えます。

ラベルの非表示

Cell Configuration

Results grid template > Row 1 > Column 2

Colspan: 2 Rowspan: 1

+ Add Item

Asset Description (Attribute Value)

Show label ☐

Asset Description

Conditional hide attribute

Enter attribute key

Action

Refinement

Conditional Action Display

If Select Attribute Value Then

Enable

11 Click Preview

Preview Back Cancel

10. 「資産摘要」を選択します。
11. Preview をクリックします。
12. セルをクリックします。
13. +項目の追加を選択して、値項目を追加します。

値項目の追加

Cell Configuration

Results grid template > Row 2 > Column 1

Colspan: 1 Rowspan: 1

+ Add Item

13 Add Value Item

12 Click Cell

1 record(s) selected. Clear Selection

Page 1 of 34 (1-20 of 668 items)

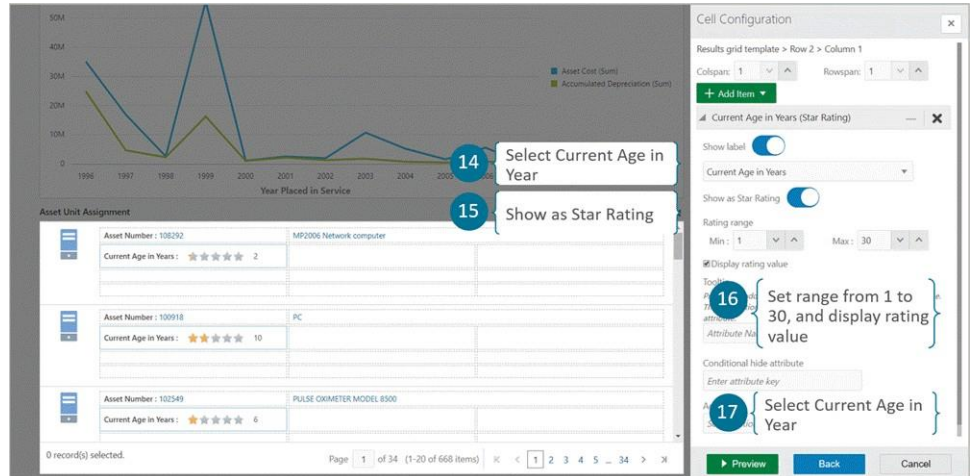
Preview Back Cancel

14. "Current Age in Years"を選択します。
15. Show as Star Rating を選択します。

16. 範囲を 1 から 30 に設定し、Display rating value を選択します。

17. "Current Age in Years"を選択します。

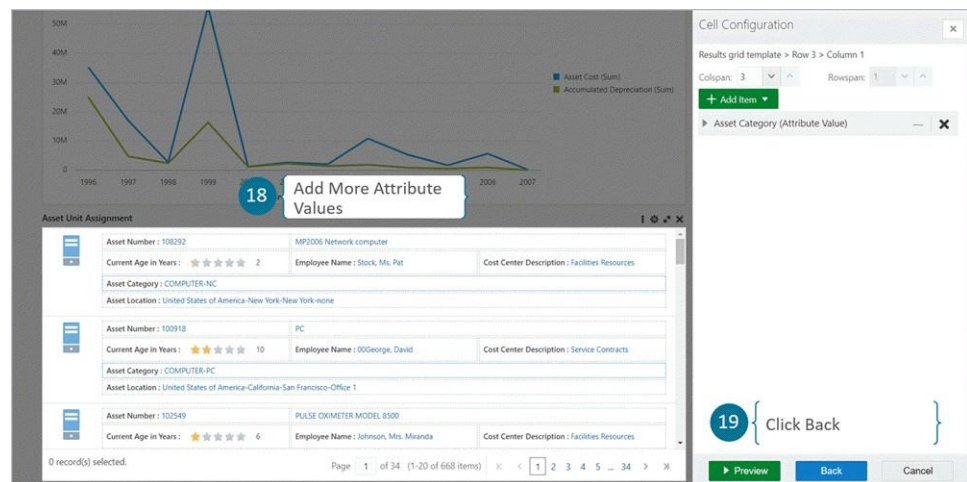
グリッドのスター評価構成



18. さらに属性値を追加できます。

オプション	説明
Visualization	- コンポーネントの高さを制御するには、高さをピクセルで設定します。 - グリッドの列数を設定します。 1 ページ当たりの項目数を設定します。 - 比較を有効にするには、Enable Compare チェック・ボックスを選択します。 Compare Header 表示名を追加します。

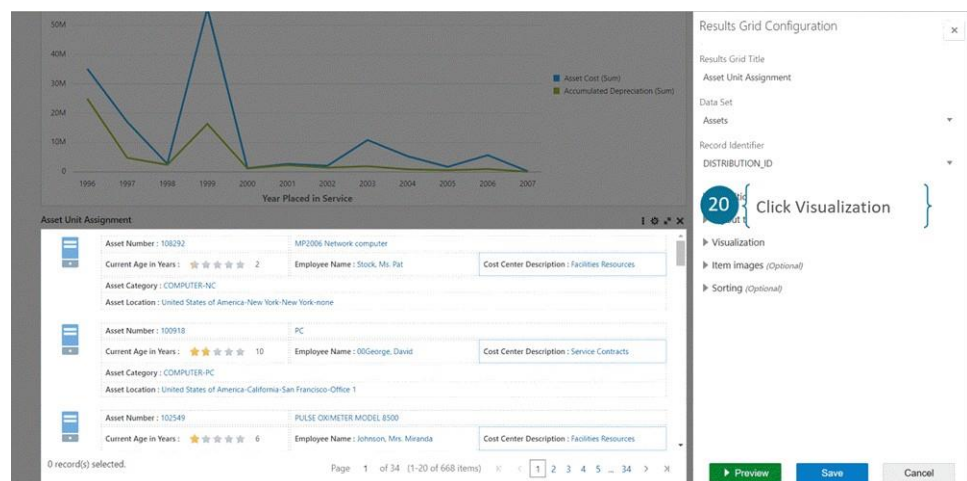
属性値の追加



19. Backをクリックします。

20. Visualization をクリックします。

Visualization の選択

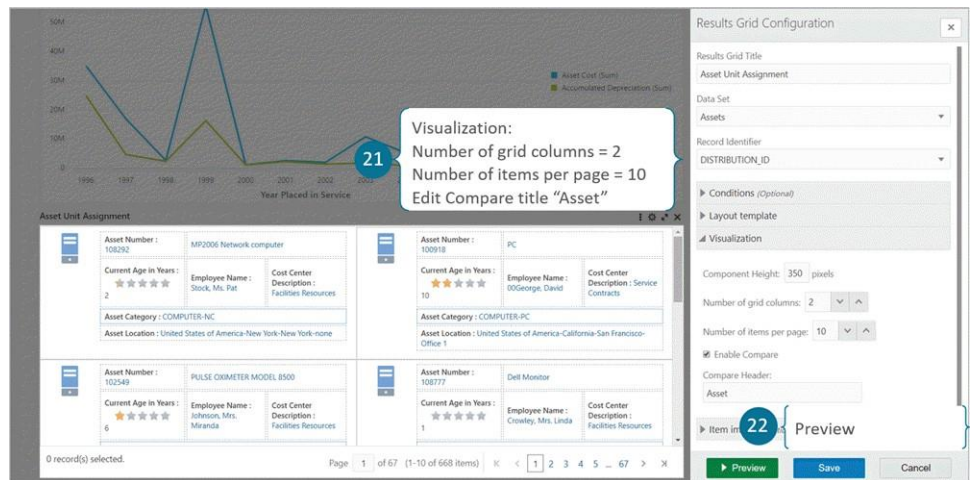


21. ビジュアル化は次のように設定されています。

- グリッド列数は 2
- ページ当たりの項目数は 10。

“Compare Header” 'Asset'を編集します。

グリッドのビジュアル化構成



22. Preview をクリックします。

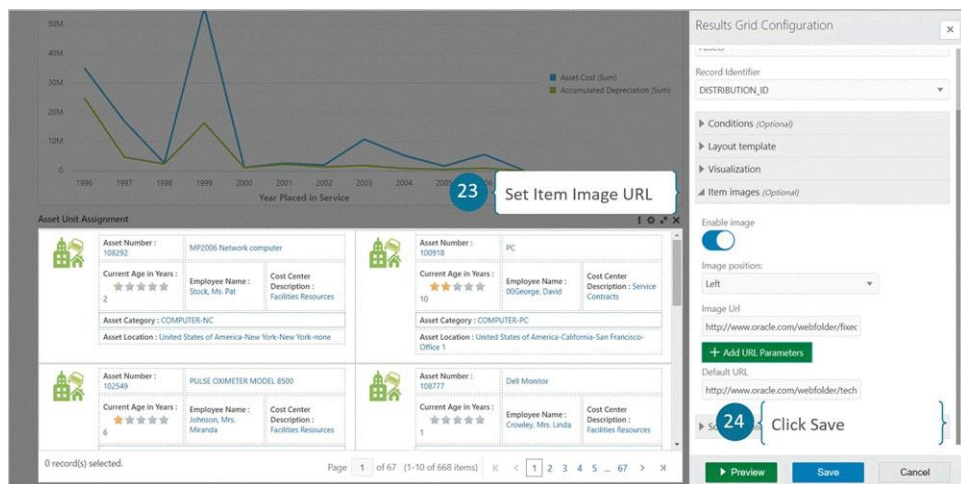
23. 項目イメージ URL を設定します。

グリッドのビジュアル化構成オプション

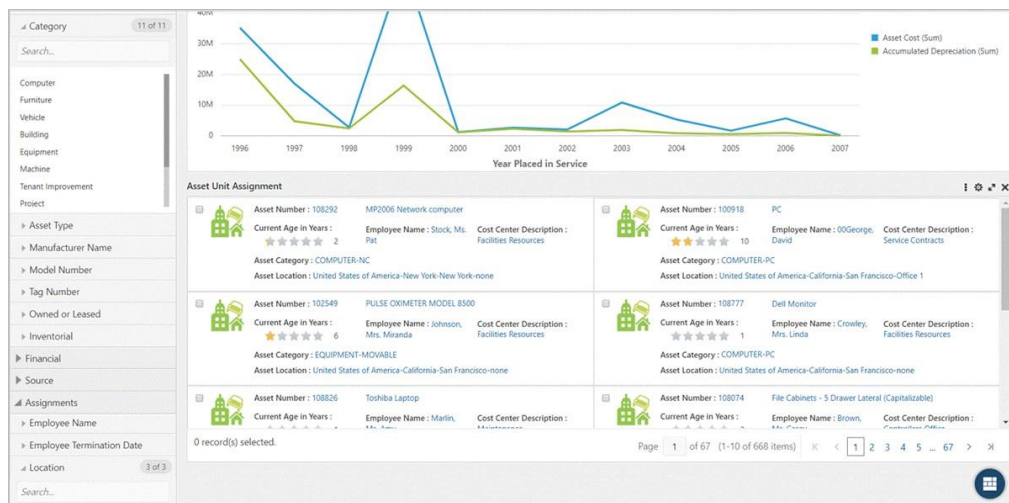
オプション	説明
Item Image	イメージを有効化して、グリッドにイメージを表示できるようにします。 - イメージの位置設定 - イメージ URL

24. Save をクリックします。

グリッド項目イメージの構成



グリッド・コンポーネント



グリッドの書式設定オプション

オプション	説明
Formatting 注意: 属性値構成の一部です。	セルを選択した後に Formatting アコーディオンを使用して、グリッド内のテキストの書式を選択します。 次のオプションを使用してテキストの書式を設定します。 • Display > Inline または Block: セル内で属性値をどのように表示するかを決定します。 1行(インライン)または別の行(ブロック)。 • Text Style > Bold, Italic, Underline • Text Alignments > Left, Right, Center • Text Size > Small (10), Medium (12), large (14) • Text Color > カラー・パレットから選択します。

グリッド内のテキストの書式設定

▲ Formatting

Display ☐ Inline ☒ Block

Text Styles: **B** *I* U

Text Alignment:   

Text Size ☐ Small ☒ Medium ☐ Large

Text Color: 

実行時のテキストの書式設定

Asset Assignments			
	Asset Number : 100051 Asset: 850 Major Category: Vehicle Owned: Luxury Units Assigned: 1 Net Book Value: 0 Black, Mr. Chris United States of America-California-San Francisco-Office Education Sales	80,000 PERSONAL ★★★★★ Assign Employee 	
		Retire Unit: 1 	
	Asset Number : 100072 BUILDING 100 Major Category: Building Office Units Assigned: 1 Net Book Value: 9,314,237.14 United States of America-New York-New York-none Controllers Office	15,523,728.47 REAL ★★★★★ Assign Employee 	
		Retire Unit: 1 	
	Asset Number : 100071 LAND Major Category: Land Occupied Units Assigned: 1 Net Book Value: 0 United States of America-California-San Francisco-none Controllers Office	500,000 REAL ★★★★★ Assign Employee 	
		Retire Unit: 1 	
	Asset Number : 100073 MANUFACTURING BUILDING Major Category: Building Manufacturing Units Assigned: 1 Net Book Value: 73,801.9 Abbott, Ms. Rachel (Rachel) United States of America-California-San Francisco-none M1, Seattle Manufacturing Plant	128,650 REAL ★★★★★ Assign Employee 	
		Retire Unit: 1 	

グリッドのソート・オプション

オプション	説明
ソート	- グリッドで使用するデフォルトのソート順をユーザーが定義できるようにします。 - 新しいソート・ルールを定義するには、+ Add New Sort をクリックします。 - 属性と、「昇順」または「降順」のいずれかのソート・オプションを選択します。 - エンド・ユーザーのソートを許可します。エンド・ユーザーが実行時にデフォルトのソートを変更できるようにします。

グリッドのソート・オプション

▲ Sorting (Optional)

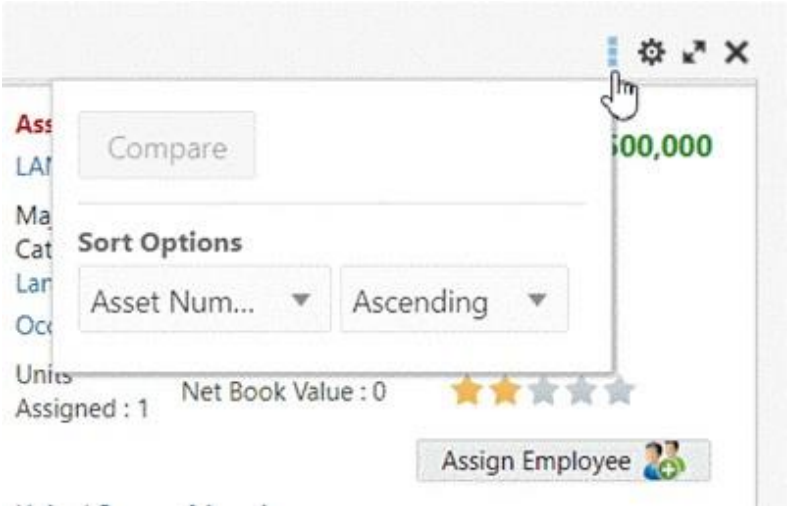
Define the default sort order for the grid

+ Add New Sort

Feature ▼ Descending ▼ 

☐ Allow end-user sorting

Allow End-User Sorting



グリッドのビジュアル化オプション

オプション	説明
Visualization	- Hide Pagination Controls > ページ送りバーなしでグリッドの表示を許可します。 - Disable Record Selection > 実行時にレコード選択チェック・ボックスを表示しないことを許可します。

ビジュアル化構成

Visualization

Component Height: 350 pixels

Number of grid columns: 5

Number of items per page: 5

Hide Paging Control ☒

Disable Record Selection ☒

☐ Enable Compare

Compare Header:
Column

Header Attribute
Select Attribute

コントロールとレコード選択がないグリッド

Top Categories

Automobile

Vehicle

Land

Office Equipment

Intangible Assets

<

レイアウト・コンポーネント

タブ付きコンポーネント・コンテナ

タブ付きコンポーネント・コンテナを使用して、ダッシュボード上のコンポーネントをグループ化できます。コンテナ内に別のコンテナを含めることはできません。また、タブ付きコンポーネントの内側または外側へのコンポーネントの移動はできません。

タブ付きコンポーネント・コンテナ

Assignments

Assigned AssetsUnassigned Assets

Asset Transfer	Asset Number	Asset Description	Units Assigned	Employee Name	Asset Location	Expense Account
	108292	MP2006 Network computer	1	Stock, Mrs. Pat	United States of America-New York-New York-none	Depreciation Con
	100918	PC	10	00George, David	United States of America-California-San Francisco-Office 1	Depreciation Con
	102549	PULSE OXIMETER MODEL 8500	1	Johnson, Mrs. Miranda	United States of America-California-San Francisco-none	Depreciation Mac
	108777	Dell Monitor	1	Crowley, Mrs. Linda	United States of America-California-San Francisco-none	Depreciation Con
	108826	Toshiba Laptop	1	Marlin, Ms. Amy	United States of America-California-San Francisco-none	Depreciation Con
	108074	File Cabinets - 5 Drawer Lateral (Capitalizable)	26	Brown, Ms. Casey	United States of America-Washington, DC-Washington, DC-none	Depreciation Furr
	108557	Company Car	1	Joseph, Brian	United States of America-California-San Francisco-none	Depreciation Veh
	108963	File Cabinets - 5 Drawer Lateral (Capitalizable)	26	Brown, Ms. Casey	United States of America-New York-New York-none	Depreciation Furr
	102541	PULSE OXIMETER MODEL 8500	1	Johnson, Mrs. Miranda	United States of America-California-San Francisco-none	Depreciation Mac
	107429	Company Car	1	Apt, Peter M.	United States of America-California-San Francisco-none	Depreciation Veh

0 record(s) selected.

Page 1 of 51 (1-10 of 509 items)12345...51

構成

- タブ付きコンテナはコンポーネントを複数のタブにグループ化します。各タブには 1 つ以上のコンポーネントを含めることができます。
- デフォルトで、コンテナには 2 つのタブが含まれています。

タブ付きコンポーネント・コンテナの構成

オプション	説明
Title	コンポーネントのタイトルを追加します。
+ Create Tab	新しいタブを追加します。
Tab Title	タブのタイトルを追加します。

実行時のタブの削除およびタブ付きコンテナ内のタブの順序変更は、管理ユーザーのみが実行できます。

削除アイコンはタブ付きコンポーネント・コンテナ内に複数のタブが含まれている場合にのみ表示されます。

タブ付きコンポーネントの構成

Tabbed Component Title

Assignments

Tab Layout ID: 1t9vxq4yh8t

+ Create Tab

Tab 1

Tab Title Assigned Assets

Tab 2

タブの削除

Assets

Total Categories x Asset Age Details x Financial Summary x

Aggregated Table

Major Category	Asset Cost (Sum)	Accumulated Depreciation (Sum)	Net Book Value (Sum)
Building	90,541,318.75	25,573,476.14	64,155,974.27
Computer	592,980.19	336,547.10	298,482.09
Construction Equipment	150,000	23,750	126,250
Equipment	1,759,660.73	1,690,869.74	68,790.99
Furniture	12,020,570	5,990,869.65	6,054,700.35
Intangible Assets	500,000	0	0
Land	8,500,000	1,188,888.84	0
Machine	3,055,450	2,386,774.96	668,675.04
Project	103,840	28,396.65	75,443.35
Tenant Improvement	709,401.64	625,389.13	84,012.51

Page 1 (1-10 of at least 11 items) < 1 2 >

共通のコンポーネント・コントロール

ページ送りコントロール

詳細データを表示するコンポーネント内にはページ送りコントロールがあり、これを使用してページ間のナビゲーションを実行できます。次のコンポーネントにはページ送りコントロールが含まれています。

- 結果表
- 集計表

ページ送りコントロール

☐	108557	Company Car				YES	OWNED	YES	V
☐	108963	File Cabinets - 5 Drawer Lateral (Capitalizable)				YES	OWNED	YES	F
☐	102541	PULSE OXIMETER MODEL 8500				YES	OWNED	YES	E

0 record(s) selected. Page 1 of 63 (1-10 of 629 items) 1 2 3 4 5 ... 63 > >>

コンポーネント内の項目の比較

一部のコンポーネントの「処理」メニューには「比較」オプションが含まれており、選択した項目を比較できます。次のコンポーネントには「比較」オプションが含まれています。

- 結果表
- グリッド

項目を比較するには、次の手順を実行します。

- 比較するレコードを選択します。
 - 比較する各項目のチェック・ボックスを選択します。
 - 「処理」メニューから「比較」を選択します。
- 「比較」ポップアップ内にベース・コンポーネントに構成されているすべての属性が表示されます。
- ユーザーは選択した項目間の相違点を強調表示できます。
 - 「処理」メニューから、「差異の強調表示」を選択します。
 - 強調表示を削除するには、「処理」メニューから「ビューの再設定」を選択します。

比較オプション

Compare Content ×

Type to filter by attribute × Comparing 2 record(s) Actions ▾

Attribute	Asset# 1	Asset# 2
Asset Number	108292	100918
Asset Description	MP2006 Network computer	PC
Date Placed in Service	2006-01-01T00:00:00Z	1997-08-01T00:00:00Z
Asset Cost	720	2500
Accumulated Depreciation	6720	12500
Net Book Value	4800	

構成

構成

オプション	説明
Enable Compare	実行時に比較を有効にするチェック・ボックス
Compare Header	統計テストを比較ヘッダーとして追加できます。
Header Attribute	比較ヘッダーをトークン化された属性として表示します。

比較の構成

☒ Enable Compare

Compare Header:

Asset

Header Attribute

Asset Number ▾

構成マトリックス

次の表に、列ヘッダーを表示する際のオプションを示します。

構成マトリックス			
ヘッダー	属性ヘッダー	表示	例
空	空	レコード n	Record 1
テキスト	空	テキスト n	Item 1
テキスト	属性名	テキスト: 属性値	Item: LCD Thin Panel Monitor
空	属性名	属性表示名: 属性値	Description: LCD Thin Panel Monitor

実行時オプション

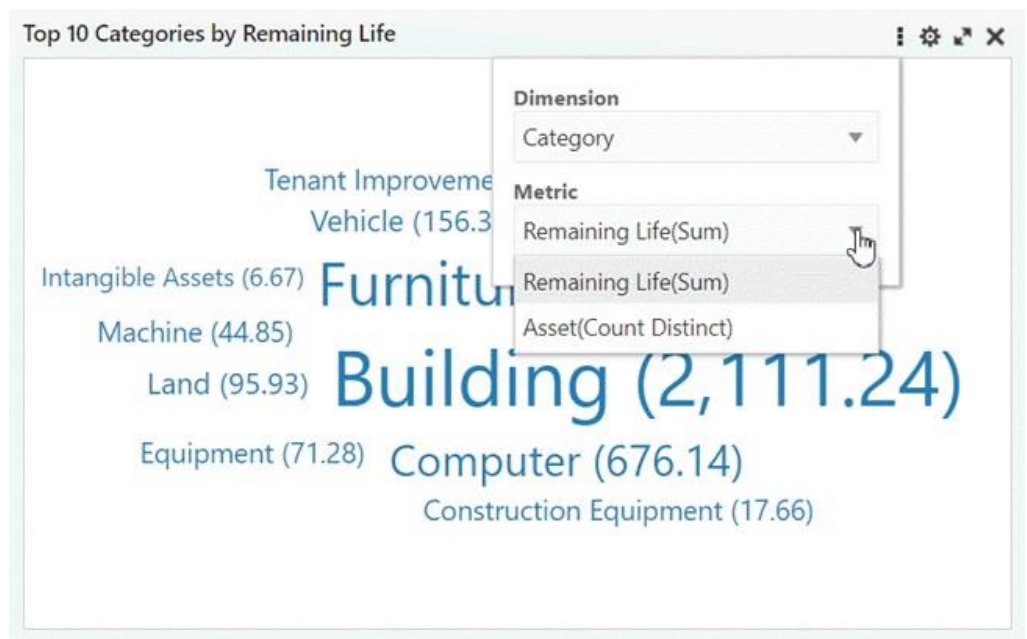
一部のコンポーネントの実行時オプションには、ビジネス・ユーザーがコンポーネント内のデータの表示方法を制御するためのオプションが含まれています。次のコンポーネントには実行時オプションが含まれています。

- タグ・クラウド
- チャート

タグ・クラウド

タグ・クラウドは、ディメンションと複数のメトリックがある場合に使用できます(構成に基づく)。ユーザーは実行時オプションを使用して、ドロップダウン・リストからメトリックまたはディメンションを選択し、表示できます。

タグ・クラウドの実行時オプション

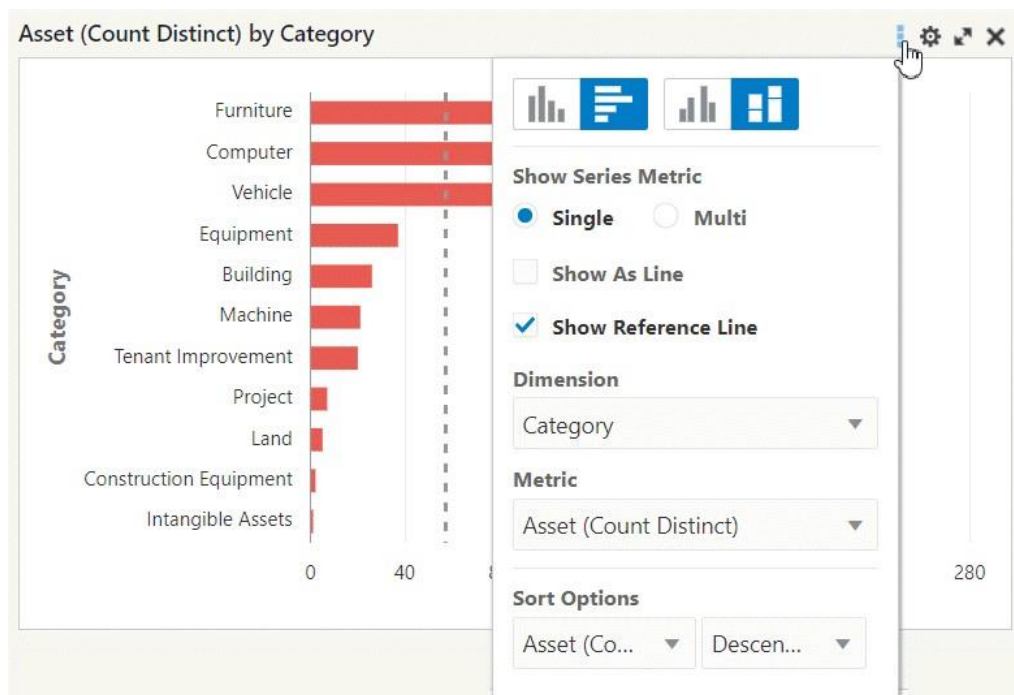


チャート

次の構成オプションがあります。

- ディメンションと複数のメトリックがある場合に使用できます(構成に基づく)。ユーザーはドロップダウン・リストからメトリックまたはディメンションを選択し、表示できます。
- チャートの表示は、水平、平行、積上げ、非積上げから選択できます。
- 1 つのメトリックのみを表示、または複数のメトリック(複数のメトリックが定義されている場合)を表示できます。
- 棒のかわりに折れ線のチャートを表示します。
- 参照線を表示/非表示にします(構成されている場合)。
- チャートのソートを実行時に変更します(有効の場合)。

チャートの実行時オプション



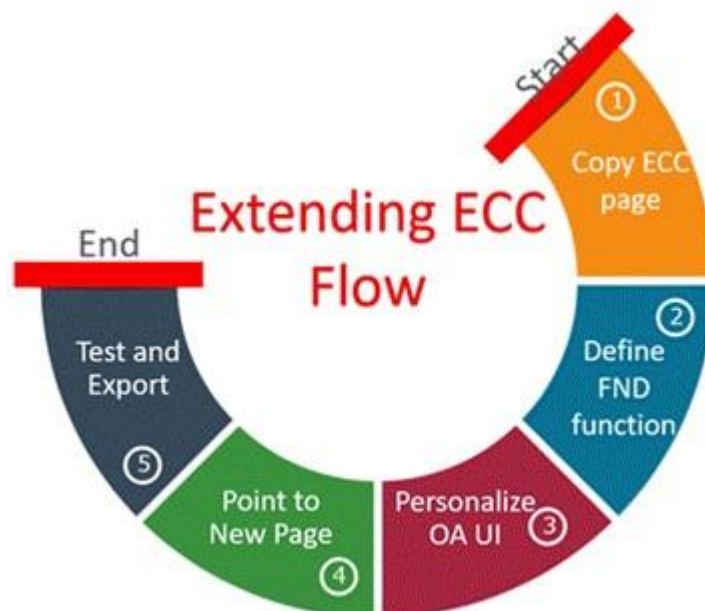
Oracle Enterprise Command Centers の拡張方法

概要

Oracle E-Business Suite は様々な業界や環境下で使用されるため、特別な要件を求めるユーザーも存在します。このようなケースには管理ユーザーが既存のダッシュボードを拡張することで対応できます。

次は、Oracle Enterprise Command Center Framework ダッシュボードの拡張フローを説明した図です。

Oracle Enterprise Command Center Framework の拡張フロー



1. Oracle Enterprise Command Center ページをコピーします。
2. Oracle Application Object Library (FND)機能を定義します。
3. Oracle Application Framework ユーザー・インタフェース・ページをパーソナライズします。
4. 新しいページが参照されるようにします。
5. 新しいページをテストおよびエクスポートします。

既存のダッシュボードを拡張するシナリオ

この項の目標は次のとおりです。

- 既存のデータ・セットで新しいダッシュボードを作成します。
- コンポーネントを構成またはビジュアル化する方法を変更します。
- データ・セットで使用可能な属性を表示/非表示にします。
- 既存のコンポーネントを削除または新しいコンポーネントを追加します。
- 処理またはリンクを削除します。

既存のダッシュボードの拡張

この項では、Oracle Enterprise Command Center Framework ダッシュボードを拡張し、同じ Oracle Application Framework リッチ・コンテナ内に表示する方法について、機能管理者の観点から説明します。

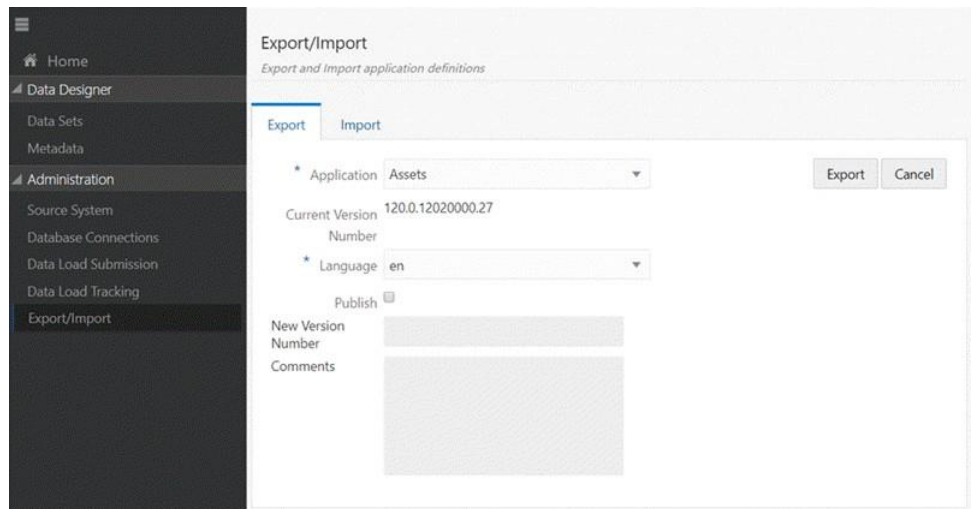
このシナリオでは、固定資産コマンド・センターの資産取得価額ダッシュボードを拡張する例を取り上げます。

既存のアプリケーションのバックアップ

アプリケーションを拡張する前に、次の手順に従って既存のアプリケーション(提供されたもの)をバックアップします。

1. ECC 開発者職責を使用して、Export/Import ページにナビゲートします。
2. Export タブをクリックします。
3. アプリケーション名を選択します。
4. 言語として en を指定します。
5. Export ボタンをクリックします。

アプリケーションのエクスポート

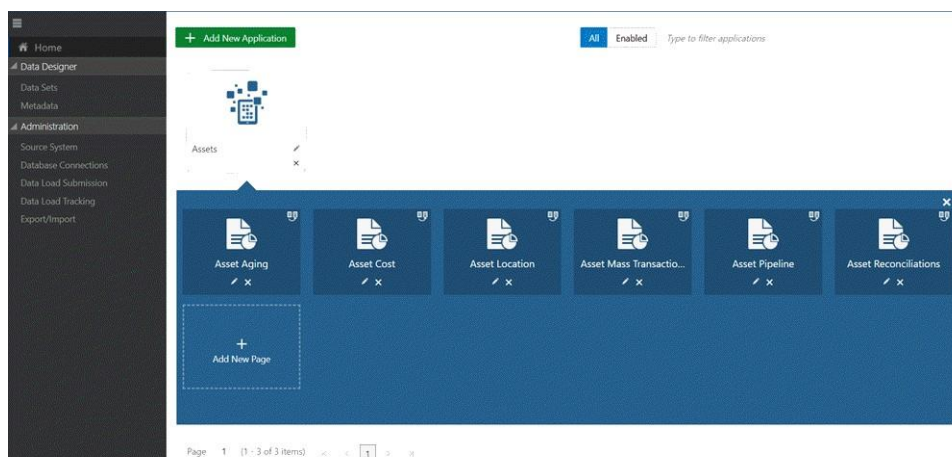


既存のページのコピー

ここでは資産取得価額ダッシュボードを拡張するため、まず既存のダッシュボードを次の手順でコピーします。

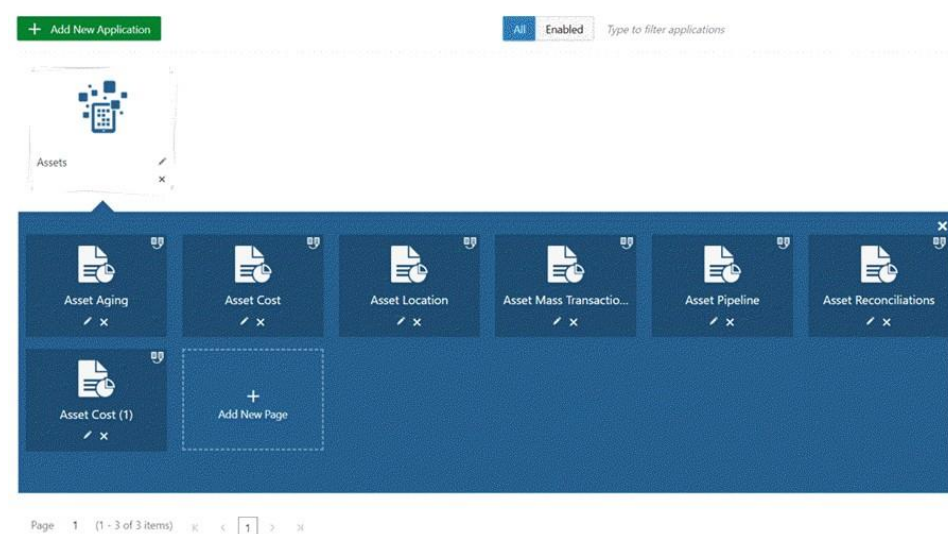
1. ECC 開発者職責でログインします。
2. ホーム・ページで、資産アプリケーションを検索します。
3. 「資産」をクリックし、アプリケーション・ページ内を探索します。

アプリケーション・ページ



4. 「資産取得価額」ページで Copy Page アイコンをクリックします。
5. コピーしたページがアプリケーションのページ・リストに、「資産取得価額(1)」という名前で追加されます。

新しくコピーされたページ



6. コピーされたページ上の鉛筆アイコンをクリックして、ページ定義を編集します。
7. ページ定義で、ページ・キーと表示名を変更します。

注意: ページ・キーと表示名には'xx'の接頭辞を付与することをお勧めします。

ページ定義の編集

Pages

Create one or more pages for your application

Application: Assets

Page Short Name: xx-asset-cost

Page Name: XX|Asset Cost

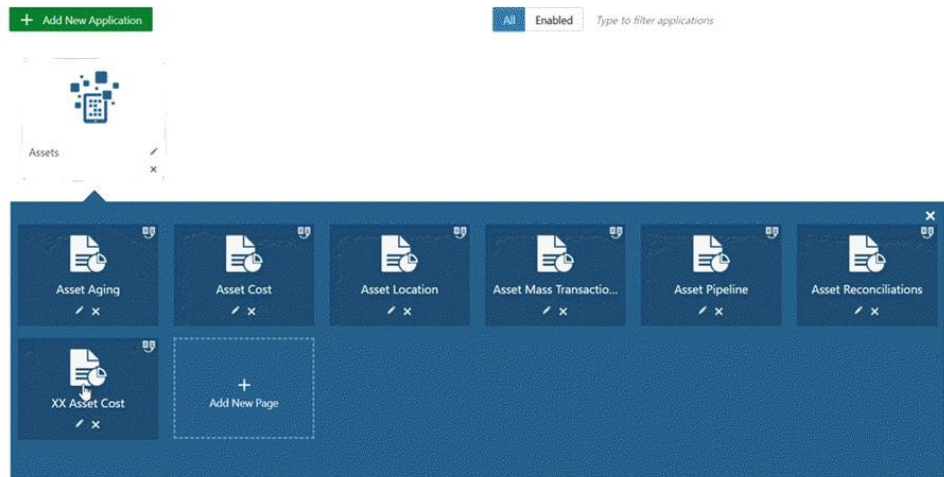
Page URL: /fa/xx-asset-cost

Enabled: ☒

Save Cancel

8. 変更を保存します。
9. ページ名(XX 資産取得価額)をクリックしてページを開き、拡張を行います。

新しくコピーされたページ



コピーしたページの拡張

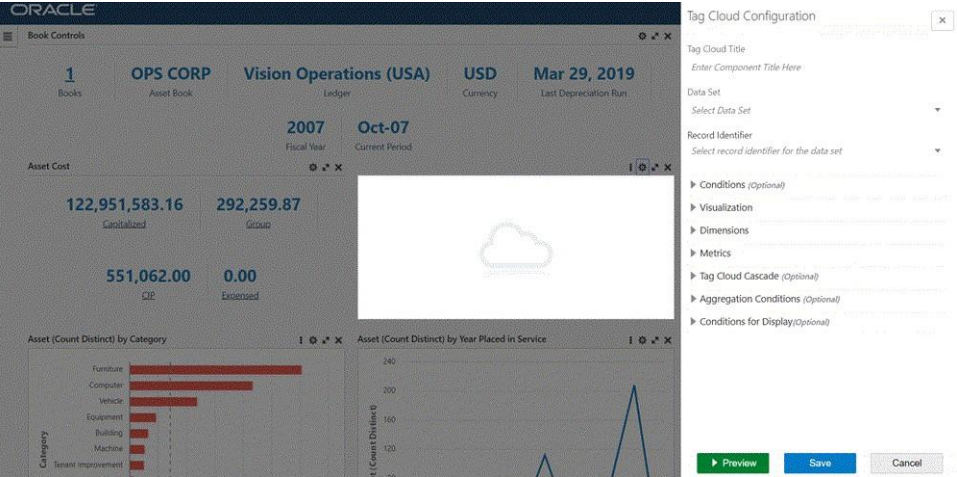
機能要件に基づいて資産取得価額ダッシュボードの機能を充実させるため、次のシナリオで拡張を行います。

既存のビジュアル化の置換

資産取得価額要約バーをタグ・クラウドに置き換えます。タグ・クラウドには資産タイプが資産取得価額に基づいて表示され、プロパティ・タイプにカスケードできます。

- 1. Add Component List を開きます。タグ・クラウド・コンポーネントを資産取得価額要約バーの横にドラッグ・アンド・ドロップします。
- 2. 「構成」アイコンをクリックします。

タグ・クラウドの構成



- 3. タグ・クラウドを構成します。

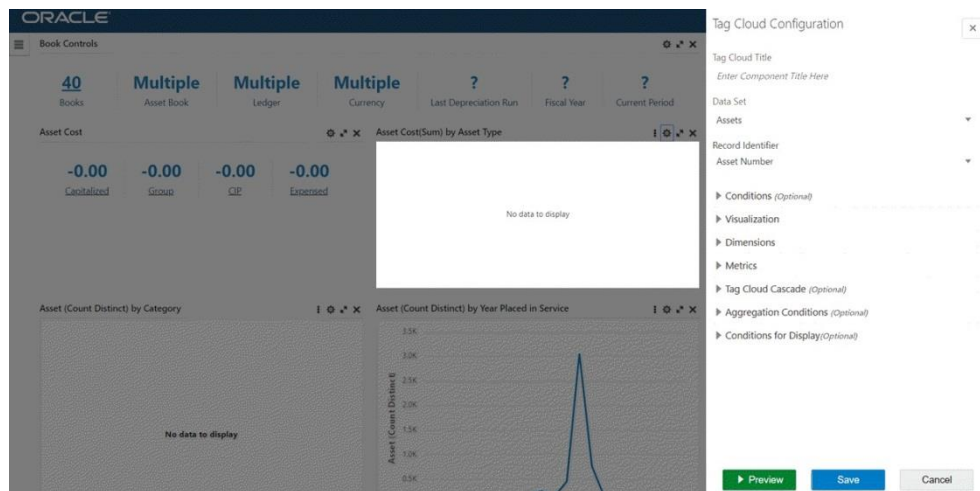
タグ・クラウドの構成オプション

オプション	説明
Title	タイトルは空のままにしておき、定義されているディメンションとメトリックに基づく動的タイトルが表示されるようにします。
Data Set	Assets
Record Identifier	Asset Number
注意: Record Identifier は、データ・セット内の粒度レベルに基づく集計重複レコードを防ぎます。	

オプション	説明
Condition	NA
Visualization	Show Metric Value を有効にします。 Number of Items = 10
ディメンション	Asset Type Property Type
メトリック	Asset Cost (Sum) Asset (Count Distinct)
Tag Cloud Cascade	Asset Type Property Type
Aggregation Condition	NA
Condition for Display	Book Type Code (Count Distinct) = 1 注意: タグ・クラウドでは取得価額に基づいて資産タイプが表示されるため、資産台帳のコンテキスト内で取得価額を表示することをお勧めします。

4. Preview をクリックして、コンポーネントの実行時ビューを表示します。

タグ・クラウドの構成プレビュー



注意: 資産台帳がまだ選択されていないため、ユーザーが 1 つの資産台帳を選択するまではタグ・クラウドに No data to Display と表示されます。

5. Save をクリックします。要約バーの「フラグ・ポップアップ」から資産台帳を選択し、選択した資産台帳のコンテキストでデータを表示します。

6. 資産取得価額要約バーを削除します。

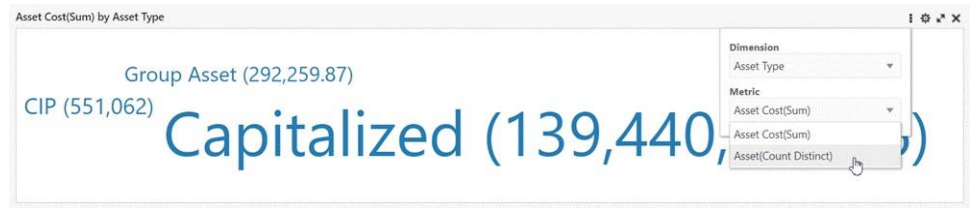
削除アイコンをクリックします。

資産タイプ別の資産取得価額(合計)のタグ・クラウド



7. ユーザーは実行時オプションを使用して、タグ・クラウドに表示するメトリックを変更できます。「オプション」をクリックし、ドロップダウン・リストを使用してメトリック値を変更します。

メトリック値を変更するドロップダウン・リスト



資産タイプ別の資産(個別の件数)



8. タグ・クラウド内の任意の用語でフィルタをかけると、Property type コードにカスケードされます。
クリックしてフィルタを適用します。

タグ・クラウドのカスケード・フィルタ



チャートの更新

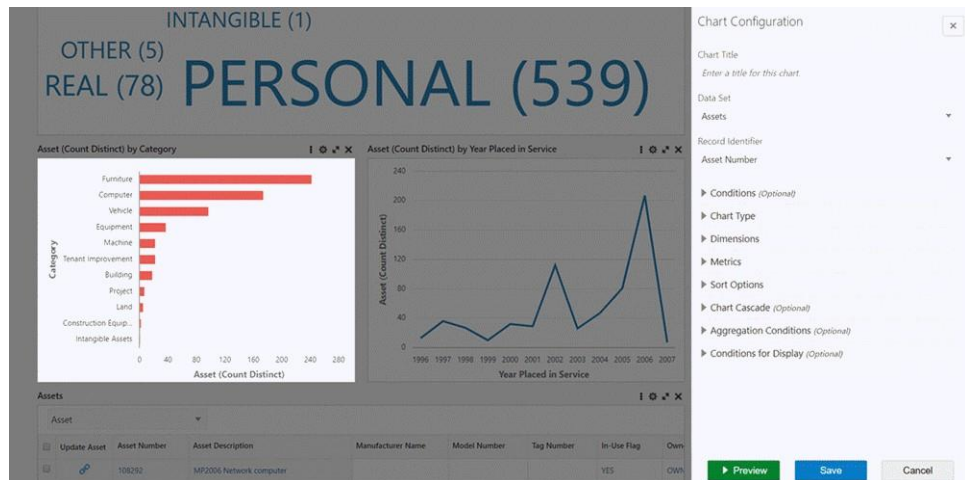
カテゴリ別平均取得価額(合計)の参照線を追加して、ユーザーがカテゴリ別資産取得価額(合計)を平均と比較できるようにします。

この拡張では、次の手順を実行します。

- 参照線を追加します。
- チャート構成を変更し、個別の件数のかわりに資産取得価額が最初に表示されるようにします。
- 資産台帳に基づいて、新しい表示条件を定義します。

1. 「構成」をクリックして、チャート構成パネルを開きます。

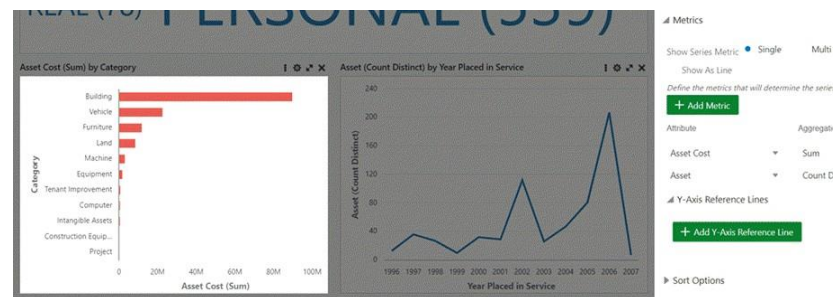
チャート構成の編集



2. 「メトリック」構成オプションを開き、次を更新します。
 1. 最初の定義済メトリック、Asset (Count Distinct)を削除します。
 2. + Add Metric をクリックします。
 3. Asset (Count Distinct)を定義します。
 4. Preview をクリックします。

実行時に Asset Cost (Sum)がチャートに表示されます。

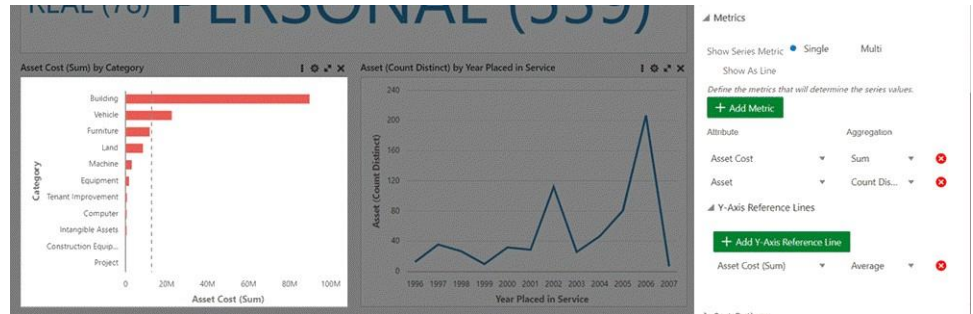
更新されたメトリック構成



3. +Add Reference Line をクリックします。
 1. 最初の定義済メトリック、"Asset Cost (Sum)"を選択します。

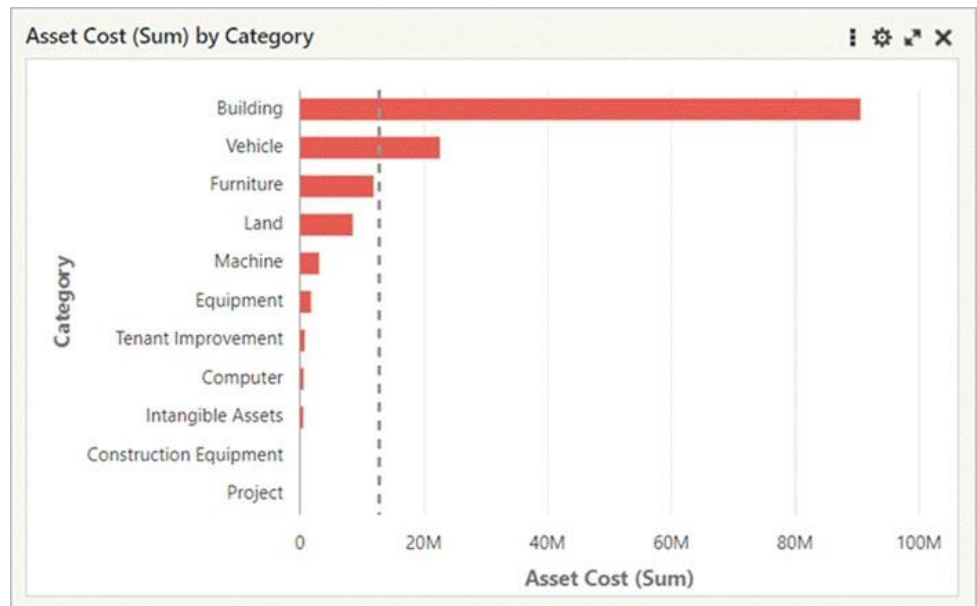
2. 集計関数「平均」を選択します。
3. Preview をクリックすると、カテゴリ別平均取得価額(合計)に基づく参照線がチャートに表示されます。

参照線の構成



4. チャート構成を保存します。

参照線



結果表の更新

資産取得価額ダッシュボードにハイパーリンク処理を定義して、資産事業所ダッシュボードにナビゲートできるようにします。

資産の結果表

Assets

Asset	Update Asset	Asset Number	Asset Description	Manufacturer Name	Model Number	Tag Number	In-Use Flag	Owned or Leased	Inventorial	C
		108292	MP2006 Network computer				YES	OWNED	YES	Cr
		100918	PC				YES	OWNED	YES	Cr
		102549	PULSE OXIMETER MODEL 8500				YES	OWNED	YES	Ed
		108777	Dell Monitor				YES	OWNED	YES	Cr
		108826	Toshiba Laptop				YES	OWNED	YES	Cr
		108074	File Cabinets - 5 Drawer Lateral (Capitalizable)				YES	OWNED	YES	Fu
		100719	MANUFACTURING PLANT				YES	OWNED	NO	Be
		139762	MP2007 Network computer				YES	OWNED	YES	Cr
		108557	Company Car				YES	OWNED	YES	W
		108963	File Cabinets - 5 Drawer Lateral (Capitalizable)				YES	OWNED	YES	Fu

0 record(s) selected. Page 1 of 64 (1-10 of 637 items) 1 2 3 4 5 ... 64

1. 構成アイコンをクリックして、構成パネルを開きます。

結果表の構成

Results Table Configuration

Results Table Title: Assets

Data Set: Assets

Record Identifier: Asset Number

Conditions (Optional):

Attributes:

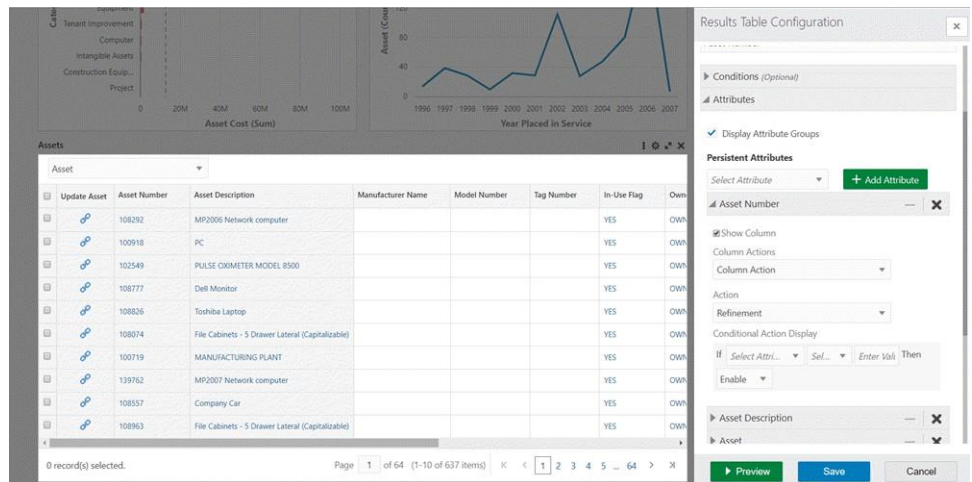
Actions (Optional):

Sorting (Optional):

Preview Save Cancel

2. 「属性」を展開し、属性レベルで処理を定義します。

列の処理の構成



3. 次の構成情報を適用します。

列の処理の構成情報

オプション	値
列処理	列処理
処理	Hyperlink
ツールチップに処理説明を表示	True
新しいウィンドウで開く	False
URL*	/ecc/web/eccapp/fa/asset-location?eccDeepLink=[{"fa-asset":[{"attributeKey":"ASSET_NUMBER","values":[{"0}],"operator":"=="}}}]
+ Add URL Parameters	属性の選択: Asset Number
	注意: 資産番号参照{0}が URL に追加されます。

* ディープ・リンク:

- 2つのページ間でナビゲーションを行い、宛先のページにフィルタを適用するときに使用します。
- 範囲フィルタ、選択フィルタおよび検索フィルタを適用できます。

4. Preview と Save をクリックします。

構成のプレビューと保存

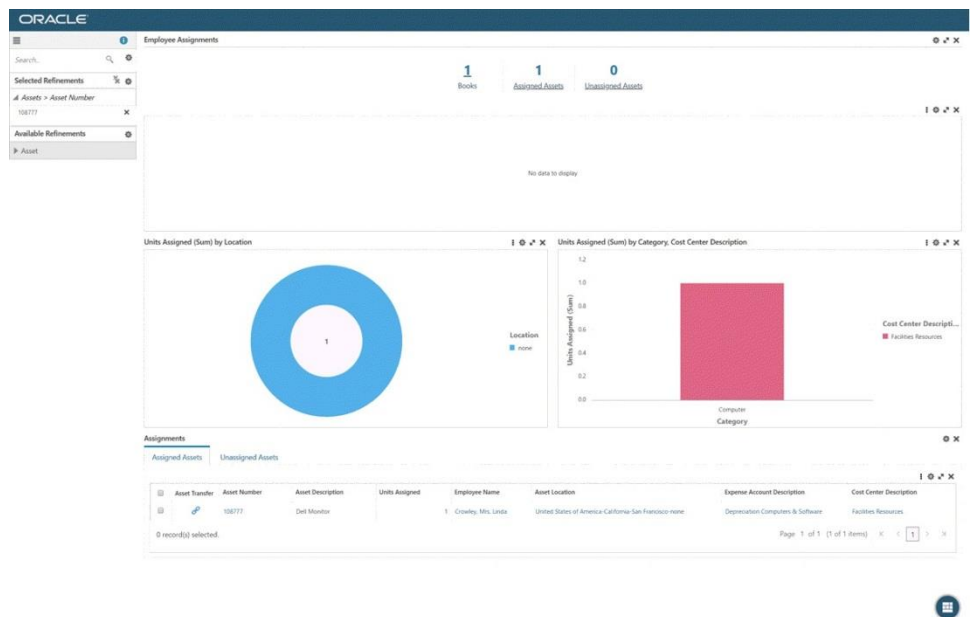
The screenshot displays the 'Assets' table with columns: Asset Number, Asset Description, Manufacturer Name, Model Number, and Tag Number. A 'Results Table Configuration' dialog box is open on the right, allowing users to configure the table's appearance and actions. The dialog includes options for 'Select Attribute', 'Add Attribute', 'Show Column', 'Column Actions', 'Column Action', 'Action', 'Hyperlink', 'Display action description in tooltip', 'Go to Asset Location', 'Open link in a new window', 'Add URL Parameters', 'Attribute', 'Conditional Action Display', and 'Enable'.

5. 実行時にユーザーは、資産番号をクリックして資産事業所ダッシュボードにナビゲートし、選択した資産番号でダッシュボードをフィルタリングできます。

資産事業所ダッシュボードの例

Asset	Asset Number	Asset Description	Manufacturer Name	Model Number	Tag Number	In-Use Flag	Owned or Leased	Inventorial	Ci
☐ Update Asset	108292	MP2006 Network computer				YES	OWNED	YES	Ci
☐ 100918	100918	PC				YES	OWNED	YES	Ci
☐ 102549	102549	PULSE OXIMETER MODEL 8500				YES	OWNED	YES	Ec
☐ 108777	108777	Dell Monitor				YES	OWNED	YES	Ci
☐ 108826	108826	Toshiba Laptop				YES	OWNED	YES	Ci
☐ 108074	108074	File Cabinets - 5 Drawer Lateral (Capitalizable)				YES	OWNED	YES	Fc
☐ 100719	100719	MANUFACTURING PLANT				YES	OWNED	NO	Bi
☐ 139762	139762	MP2007 Network computer				YES	OWNED	YES	Ci
☐ 108557	108557	Company Car				YES	OWNED	YES	Vi
☐ 108963	108963	File Cabinets - 5 Drawer Lateral (Capitalizable)				YES	OWNED	YES	Fc

資産番号でフィルタをかける例



FND 機能の作成

コピーした資産取得価額ダッシュボードに必要なすべての拡張を適用した後、新しい FND 機能を作成して、次の手順でこの機能にアクセス権を付与します。

1. Oracle E-Business Suite 環境で、機能管理者職責にナビゲートします。
2. 「コア・サービス」タブで、「機能」サブタブを選択します。
3. 「機能の作成」タブを選択します。
 1. 機能コード'FA_ECC_XX_ASSET_COST'を入力します。
 2. 機能名として'Fixed Assets Command Center Cost'と入力します。
 3. 「タイプ」では「SSWA JSP 機能」を選択します。
 4. 「続行」ボタンをクリックします。

機能の作成

ORACLE Applications Administration

Security Core Services Personalization File Manager Portletization

Lookups Messages Profile Categories Profiles Functions Menus Caching Framework Personalization

Create Function: Define Function

* Indicates required field

* Name Fixed Assets Command Center Cost

* Code FA_ECC_XX_ASSET_COST

Description New function for customized Asset Cost page

Type SSWA jsp function

Maintenance Mode Support None

Context Dependence Responsibility

Cancel Continue

Copyright (c) 1998, 2014, Oracle and/or its affiliates. All rights reserved. About this Page Privacy Statement

- 「HTML コール」テキスト・ボックスにページ URL を入力します。注意:「ページ URL」には、8-3 ページの「既存のページのコピー」項で作成したページ・キーを入力します。

機能定義のページURL の例

ORACLE Applications Administration

Security Core Services Personalization File Manager Portletization

Lookups Messages Profile Categories Profiles Functions Menus Caching Framework Personalization

Create Function: Details

* Indicates required field

Cancel Back Submit

Web

* HTML Call GWY.jsp?targetAppType=ECC&targetPage=/web/ecc/app/xx-asset-cost

☐ Secured

☐ Encrypt Parameters

Host Name

Agent Name

Icon

- 「発行」ボタンをクリックします。

機能の作成確認

Confirmation
FA_ECC_XX_ASSET_COST has been updated successfully.

Functions Save Search

Simple Search

Note that the search is case insensitive Advanced Search

Name

Code FA%ECC%

Type

Go Clear

Create Function Refresh Reset Settings

Name ▲	Code ▲	Type ▲	Last Update ▲	Duplicate	Update
Fixed Assets Command Center Accountant Home	FA_ECC_MASS_TRANS_HOME	JSP	12-Jul-2018		
Fixed Assets Command Center Aging	FA_ECC_ASSET_AGING	JSP	12-Jul-2018		
Fixed Assets Command Center Clearing	FA_ECC_ASSET_CLR	JSP	12-Jul-2018		
Fixed Assets Command Center Cost	FA_ECC_ASSET_COST	JSP	12-Jul-2018		
Fixed Assets Command Center Cost	FA_ECC_XX_ASSET_COST	JSP	03-Apr-2019		
Fixed Assets Command Center Home	FA_ECC_HOME	JSP	12-Jul-2018		

4. 「セキュリティ」タブにナビゲートし、「権限セット」を選択します。

5. 新しい FND 機能を資産権限セットに割り当てます。

1. 権限セット「固定資産コマンド・センター・アクセス権限セット」-「FA_ECC_ACCESS_PS」を検索します。
「編集」をクリックします。

権限セットの例

Permission Sets Save Search

Simple Search

Note that the search is case insensitive Advanced Search

Name

Code FA%ECC%

Go Clear

Create Permission Set Refresh Reset Settings

Name ▲	Code ▲	Last Update ▲	Duplicate	Update
Fixed Assets Command Center Access Permission Set	FA_ECC_ACCESS_PS	11-Jan-2019		
Fixed Assets Command Center Accountant Access permission Set	FA_ECC_ACC_ACCESS_PS	11-Jan-2019		

2. 新しく作成された機能を追加します。

権限セットの更新

Update Permission Set: FA_ECC_ACCESS_PS

* Indicates required field

Name: Fixed Assets Command Center Access Permission Set

Code: FA_ECC_ACCESS_PS

Description:

Permission Set Manager: Hierarchy of Children Direct Parents Grants

Select Rows: Remove

Select All | Select None

Select	Permission Set	Permission	Description
☐		Fixed Assets Command Center Cos	
☐		Fixed Assets Command Center Agr	
☐		Fixed Assets Command Center loca	
☐		Fixed Assets Command Center Mas	
☐		Fixed Assets Command Center Cles	
☐		Fixed Assets Command Center Rec	
☐		Fixed Assets Command Center Hon	
☐		Fixed Assets Command center Cos	

Name	Code	Description
Fixed Assets Command Center Cost	FA_ECC_ASSET_COST	
Fixed Assets Command Center Cost	FA_ECC_XX_ASSET_COST	New function for customized Asset Cost page

1 - 2

Apply

3. 「適用」ボタンをクリックします。

Oracle Application Framework ページのパーソナライズ

次に、Oracle Application Framework ページを次の手順でパーソナライズします。

1. Oracle E-Business Suite から固定資産コマンド・センターにナビゲートし、資産取得価額ページ(提供されたページ)にナビゲートします。
2. Oracle Application Framework のページ・レベルで「設定」、「パーソナライズ・ページ」の順に選択します。

Oracle Application Framework ページのパーソナライズ

ORACLE Fixed Assets Manager

Asset Reconciliations Asset Cost Asset Aging Asset Location Asset Mass Transactions Asset Pipeline

Book Controls

40 Books Multiple Asset Book Multiple Ledger Multiple Currency ? Last Depreciation Run ? Fiscal Year ? Current Period

Asset Cost

-0.00 Capitalized -0.00 Group -0.00 CIP -0.00 Expensed

Asset (Count Distinct) by Category

Asset (Count Distinct) by Year Placed in Service

3.5K 3.0K 2.5K

3. 「編集」をクリックして、資産取得価額ページのリッチ・コンテナをパーソナライズします。

資産取得価額ページのリッチ・コンテナのパーソナライズ

Header: (AssetRecHdrRN)	Yes								
Rich Container: (AssetRecPage)	Yes								
Header: (AssetCostHdrRN)	Yes								
Rich Container: (AssetCostPage)	Yes								
Header: (AssetAgingHdrRN)	Yes								
Rich Container: (AssetAgingPage)	Yes								
Header: (AssetLocHdrRN)	Yes								
Rich Container: (AssetLocPage)	Yes								
Header: (MassTransHdrRN)	Yes								
Rich Container: (MassTransPage)	Yes								

- 新しく作成した機能 FA_ECC_XX_ASSET_COST を「リンク先機能」に追加して、リッチ・コンテナを職責レベルでパーソナライズします。

ページはサイトまたは組織レベルでもパーソナライズできます。

職責レベルでのページのパーソナライズ

Personalization Context

Scope: Page: Assets Command Center
Document Name: /oracle/apps/fa/ecc/webui/FAECCHomePG
Site: Include
Organization: Vision Operations
Responsibility: Fixed Assets Manager

Personalization Properties

Clear Personalization [Go] Choose Levels Displayed |

	Original Definition	Site	Organization: Vision Operations	Responsibility: Fixed Assets Manager	Result / Source
Admin Personalization	true	Inherit	Inherit	Inherit	true / Original Definition
AutoResize	true	Inherit	Inherit	Inherit	true / Original Definition
Content Type	ECC	Inherit	Inherit	Inherit	ECC / Original Definition
Destination Function	FA_ECC_ASSET_COST	Inherit	Inherit	FA_ECC_XX_ASSET_COST	FA_ECC_XX_ASSET_COST / Responsibility
Height	Default	Inherit	Inherit	Inherit	Default / Original Definition
Name	Default	Inherit	Inherit	Inherit	Default / Original Definition

- 「適用」ボタンをクリックします。

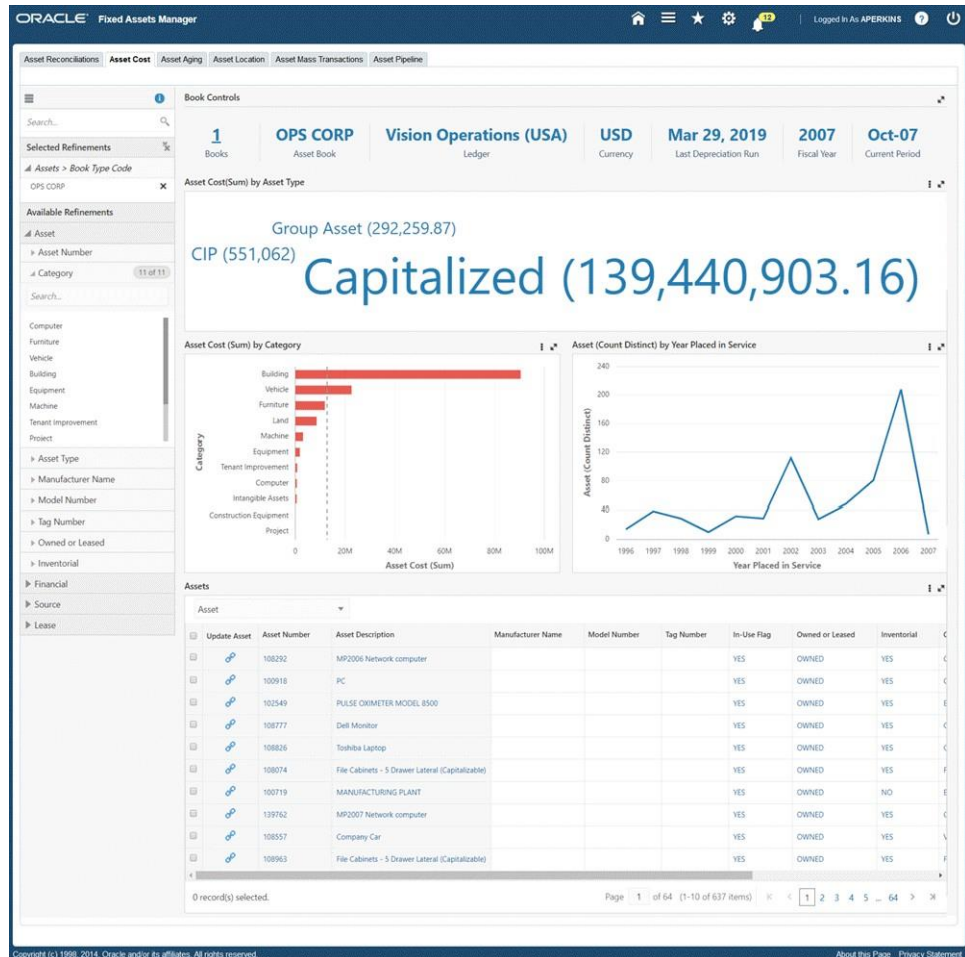
テストおよびエクスポート

Oracle Application Framework ページをパーソナライズした後に、拡張した新しいページに Oracle E-Business Suite からアクセスします。

- Fixed Assets Manager 職責にナビゲートします。

2. 固定資産コマンド・センターにナビゲートし、資産取得価額ダッシュボードのタブを選択します。
3. 拡張した新しいダッシュボードが、提供された資産取得価額ダッシュボードのかわりに表示されます。

新しい資産取得価額リスト



4. ダッシュボード機能を検証します。
5. ECC 開発者職責にナビゲートします。
6. Export/Import ページにナビゲートし、Export タブをクリックして更新したアプリケーションをエクスポートします。

更新したアプリケーションのエクスポート

The screenshot shows the 'Export/Import' interface. The left sidebar contains a menu with 'Home', 'Data Designer', 'Data Sets', 'Metadata', 'Administration', 'Source System', 'Database Connections', 'Data Load Submission', 'Data Load Tracking', and 'Export/Import'. The 'Export/Import' page has two tabs: 'Export' and 'Import'. The 'Export' tab is selected, displaying a form with the following fields: 'Application' (dropdown menu showing 'Assets'), 'Current Version Number' (text input showing '120.0.12020000.27'), 'Language' (dropdown menu showing 'en'), 'New Version Number' (text input), and 'Comments' (text area). There are 'Export' and 'Cancel' buttons at the top right, and a 'Publish' button below the form fields.

既存のデータ・セットの拡張

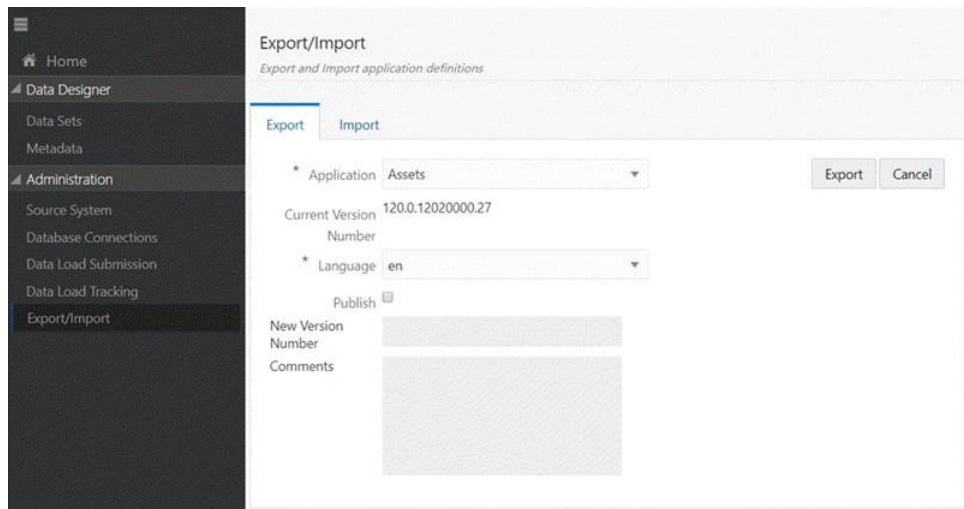
この項の目標は次のとおりです。

- データ・セットを拡張し、Oracle E-Business Suite に存在するものの即時には提供されない属性を含めます。
- それらの属性を Oracle Enterprise Command Center Framework ダッシュボードの別の部分に表示します。
- 属性シナリオを追加します。
- 既存のアプリケーションをバックアップします。

アプリケーションを拡張する前に、次の手順に従って既存のアプリケーション(提供されたもの)をバックアップします。

1. ECC 開発者職責を使用して、Export/Import ページにナビゲートします。
2. Export タブをクリックします。
3. アプリケーション名を選択します。
4. 言語として en を指定します。
5. Export ボタンをクリックします。

アプリケーションのエクスポート



メタデータ・ロード・フェーズの作成:

新しいメタデータ・ロードを作成する前に、提供されたデータ・セット DB ビューとデータ・ロード PL/SQL パッケージのコピーを作成する必要があります。A-2 ページの「サンプル PL/SQL パッケージ」を参照してください。

1. 新しいメタデータ・ロード・フェーズを実装します。

既存のデータ・セット・メタデータを拡張し、新しいメタデータ・ロード・フェーズを実装します。B-2 ページの「データ・ロードおよびメタデータ・ロードのシグネチャ」を参照してください。

注意: 新しい属性を使用してデータ・セットを拡張する場合は、データの粒度レベルを同じにする必要があります。データ・セット内のデータの粒度レベルの変更が必要になる属性は追加しないでください。データ・セット ECC_SPEC_ID 構造の詳細は、D-1 ページの「データ・セット ECC_SPEC_ID の構造」を参照してください。

2. データ・ロードの新しいロードを実装します。

メタデータ・フェーズに新しい属性を定義した後に、データ・ロードの新しいロード・ルールを実装し、定義した属性にデータが移入されるようにします。インポート・フェーズの詳細は、6-1 ページの「動的メタデータ」を参照してください。

注意: カスタム・データ・ロード・ルールの連番は 101 から開始する必要があります。

新しいロード・ルール

Load Rules Security Rule

Add Delete

Load Type	Package Name	Procedure Name	Connection name	Sequence
Metadata Load	CUSTOM_FA_ECC_FULLLOAD	CUSTOM_GET_ECC_FULLLOAD	ebsdb	101
Full Load	CUSTOM_FA_ECC_METADATA	CUSTOM_GET_ECC_METADATA_II	ebsdb	101
Full Load	FA_ECC_UTIL_PVT	GET_ECC_DATA_LOAD_INFO	ebsdb	1

3. 新しいコンカレント・プログラムを定義します。

データを取り込むため、新しいコンカレント・プログラムを定義して、データ・ロード・ルールの前に新しいメタデータ・ロードを呼び出します。C-2 ページの「サンプル・コンカレント・プログラム」を参照してください。

Oracle Enterprise Command Center Framework ページのコピー

管理者は必要なダッシュボードをコピーし、新しいコンポーネントを追加して、それらの順序を変更する必要があります。新しく追加した属性をコンポーネント構成で使用します。

ページのコピーおよび新しいコンポーネントの追加/構成の詳細は、8-2 ページの「既存のダッシュボードの拡張」を参照してください。

新しい FND 機能の作成

ページの作成後、管理者は FND_FORM_FUNCTION を作成し、新しく作成したページが Oracle E-Business Suite に表示されるようにする必要があります。8-15 ページの「FND 機能の作成」を参照してください。

Oracle E-Business Suite のパーソナライズ

新しく作成したページの URL でパーソナライズを行い、Oracle E-Business Suite を更新します。8-18 ページの「Oracle Application Framework ページのパーソナライズ」を参照してください。

Oracle E-Business Suite でユーザーが新しいページを開けること、および新しい UI が表示されることを確認します。何も問題がなければ、アプリケーションをエクスポートし、バックアップして、変更をその他の環境に反映します。

更新したアプリケーションのバックアップ

アプリケーションをテストし、新しいエクスポートを作成し、バックアップして、その他の環境に変更を反映します。8-19 ページの「テストおよびエクスポート」を参照してください。

サンプル PL/SQL パッケージ

Oracle iProcurement の例

```
CREATE OR REPLACE PACKAGE BODY icx_ecc_util_pvt AS
/* $Header: ICXECCOP.pls 120.0.1 2019/02/28 10:13:01 jdoe noship $ */
/* Utility Package for populating ECC metadata. */
FUNCTION get_descriptors_sql(
    p_category IN NUMBER,
    p_load_type IN VARCHAR2,
    p_languages IN VARCHAR2,
    p_ds_last_success_run IN TIMESTAMP)
RETURN CLOB
IS
    l_sql_text CLOB;
    v_for_lang_pivot_clause varchar2(400) := FND_ECC_UTIL_MLS_PVT.
GEN_ECC_MLS_PIVOT_FOR_LANG_CL(p_languages);
    l_desc_sql_text VARCHAR2(30000) := ' select * from ICX_CAT_ECC_' ||
p_category || '_VL';
    l_pivot_pre_sql varchar2(2000) := 'select * from ( ';
    l_pivot_post_sql1 VARCHAR2(2000) := ' )
        PIVOT ( count(1) as IS_TRANSLATION_AVAILABLE ';
    l_pivot_post_sql2 VARCHAR2(2000) := ' ';
    l_pivot_post_sql3 VARCHAR2(2000) := 'FOR LANGUAGE IN
('||v_for_lang_pivot_clause||'))' ;
    l_last_update_condition varchar2(1000) := ' WHERE
ECC_LAST_UPDATE_DATE >= to_date(to_char(to_timestamp
('||''''||p_ds_last_success_run||''''||'), 'DD-MON-YY HH24.MI.
SS'),'DD-MON-YY HH24.MI.SS')';
    l_trans_attributes dbms_sql.varchar2_table;
    cursor c_trans_attributes is
        SELECT Decode( UPPER(regexp_replace(KEY,'[-*# ]','_') ), 'SIZE',
'SIZE_', UPPER(regexp_replace(KEY,'[-*# ]','_') ) ) ATTRIBUTE_NAME
        FROM ICX_CAT_ATTRIBUTES_TL
        WHERE stored_in_table = 'PO ATTRIBUTE_VALUES_TLP'
        AND STORED_IN_COLUMN IS NOT NULL
        AND ATTRIBUTE_ID>28
        AND LANGUAGE='US'
        AND RT_CATEGORY_ID=p_category;
BEGIN
    OPEN c_trans_attributes;
    FETCH c_trans_attributes BULK COLLECT INTO l_trans_attributes;
    CLOSE c_trans_attributes;
/*
    IF(l_trans_attributes.COUNT = 0) THEN
        IF ( p_load_type = 'FULL_LOAD' ) THEN
            l_sql_text := l_desc_sql_text;
        ELSE
            l_sql_text := l_desc_sql_text || l_last_update_condition;
        END IF;
        RETURN l_sql_text;
    END IF;
*/
    FOR i IN 1..l_trans_attributes.count
    LOOP
        l_pivot_post_sql2 := l_pivot_post_sql2 || ', MAX( ' ||
l_trans_attributes(i) || ' ) as ' || l_trans_attributes(i) ;
    END LOOP;

    IF ( p_load_type = 'FULL_LOAD' ) THEN
        l_sql_text := l_pivot_pre_sql || l_desc_sql_text ||
l_pivot_post_sql1 || l_pivot_post_sql2 || l_pivot_post_sql3;
    ELSE
        l_sql_text := l_pivot_pre_sql || l_desc_sql_text ||
l_last_update_condition || l_pivot_post_sql1 || l_pivot_post_sql2 ||
l_pivot_post_sql3;
```

```

END IF;

        RETURN l_sql_text;
    END get_descriptors_sql;
/*
** GET_ECC_DATA_LOAD_INFO - Retrieves the data load details for the
given data set key
**
** IN parameters
** p_dataset_id - data set key
** p_load_type - Indicating Full or incremental load (F/I/Custom).For
custom there are no data load rules defined
** p_ds_last_success_run - Returns the last successful etl run for the
data set key
** OUT parameters
** x_ecc_ds_meta_rec - ecc_ds_meta_rec
** x_return_status - return status
*/
-----
PROCEDURE get_ecc_data_load_info(
    p_dataset_key          IN VARCHAR2,
    p_load_type            IN VARCHAR2,
    p_ds_last_success_run  IN TIMESTAMP,
    p_languages            IN VARCHAR2,
    p_addl_params          IN ecc_sec_field_values,
    x_ecc_ds_meta_rec OUT NOCOPY ecc_ds_meta_rec,
    x_return_status OUT NOCOPY VARCHAR2 )
IS
    query_det_arr ecc_query_det_arr_type := ecc_query_det_arr_type
(NULL);
    l_apps_schema_name varchar2(25) := 'APPS';
    CURSOR c_category_view
    IS
        SELECT DISTINCT RT_CATEGORY_ID FROM ICX_CAT_CATEGORIES_TL
        WHERE LANGUAGE='US'
        AND EXISTS
            (SELECT 1 FROM ALL_VIEWS
            WHERE VIEW_NAME = 'ICX_CAT_ECC_' || RT_CATEGORY_ID || '_VL'
            AND owner=l_apps_schema_name);

    l_categories dbms_sql.number_table;
    l_category NUMBER;
    l_sql_index number := 1;
BEGIN
    select oracle_username
    into l_apps_schema_name
    from fnd_oracle_userid
    where read_only_flag = 'U';

    IF ( p_load_type = 'INCREMENTAL_LOAD' ) THEN
        query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_del_op, 'ITEMS',
p_ds_last_success_run,p_languages),g_icx_ecc_del_op);
        query_det_arr.extend ();
        l_sql_index := l_sql_index +1;
    END IF;

    query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_ins_op, 'ITEMS',
p_ds_last_success_run,p_languages),g_icx_ecc_ins_op);

    query_det_arr.extend ();
    l_sql_index := l_sql_index +1;
    query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_upd_op, 'HIERARCHY',
p_ds_last_success_run,p_languages),g_icx_ecc_upd_op);

```

```

        query_det_arr.extend ();
        l_sql_index := l_sql_index +1;
        query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_upd_op, 'ZONESB',
p_ds_last_success_run,p_languages),g_icx_ecc_upd_op);

        query_det_arr.extend ();
        l_sql_index := l_sql_index +1;
        query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_upd_op, 'ZONESP',
p_ds_last_success_run,p_languages),g_icx_ecc_upd_op);

        query_det_arr.extend ();
        l_sql_index := l_sql_index +1;
        query_det_arr(l_sql_index) := ecc_query_det_rec(get_sql_text
(p_dataset_key,p_load_type,g_icx_ecc_upd_op, 'ZONESI',
p_ds_last_success_run,p_languages),g_icx_ecc_upd_op);

OPEN c_category_view;
FETCH c_category_view BULK COLLECT INTO l_categories;
CLOSE c_category_view;
FOR i IN 1..l_categories.count
LOOP
    query_det_arr.extend ();
    l_sql_index := l_sql_index +1;
    query_det_arr(l_sql_index) := ecc_query_det_rec
(get_descriptors_sql(l_categories(i), p_load_type, p_languages,
p_ds_last_success_run ),g_icx_ecc_upd_op);
END LOOP;
x_ecc_ds_meta_rec := ecc_ds_meta_rec(p_dataset_key,query_det_arr);
x_return_status := 'S';
END get_ecc_data_load_info;
FUNCTION get_sql_text(
    p_dataset_key          IN VARCHAR2,
    p_load_type            IN VARCHAR2,
    p_operation            IN VARCHAR2,
    p_data_type            IN VARCHAR2,
    p_ds_last_success_run  IN TIMESTAMP,
    p_languages            IN VARCHAR2)
RETURN CLOB
IS
    l_sql_text CLOB;
    v_for_lang_pivot_clause varchar2(400) := FND_ECC_UTIL_MLS_PVT.
GEN_ECC_MLS_PIVOT_FOR_LANG_CL(p_languages);
    l_last_update_condition varchar2(1000) := ' WHERE
ECC_LAST_UPDATE_DATE >= to_date(to_char(to_timestamp
('||''''||p_ds_last_success_run||''''||'),'DD-MON-YY HH24.MI.
SS'),'DD-MON-YY HH24.MI.SS') ' ;
    l_pivot_pre_sql VARCHAR2(100) := 'SELECT * FROM ( ';
    l_pivot_post_sql VARCHAR2(2000) := ' )
        PIVOT (
            MAX(SHOPPING_CATEGORY) AS SHOPPING_CATEGORY,
            MAX(DESCRIPTION) AS DESCRIPTION,
            MAX(UNIT_OF_MEASURE) AS UNIT_OF_MEASURE,
            MAX(COMMENTS) AS COMMENTS,
            MAX(LONG_DESCRIPTION) AS LONG_DESCRIPTION,
            MAX(ITEM_SOURCE_TEXT) AS ITEM_SOURCE_TEXT,
            MAX(CONTENT_TYPE_FILTER) AS CONTENT_TYPE_FILTER,
            MAX(DISPLAY_PRICE) AS DISPLAY_PRICE,
            MAX(PRICE_BREAK) AS PRICE_BREAK,
            MAX(ITEM_RATING_FILTER) AS ITEM_RATING_FILTER,
            MAX(SUPPLIER_RATING_FILTER) AS SUPPLIER_RATING_FILTER,
            MAX(PUNCHOUT_MORE_DETAILS) AS PUNCHOUT MORE DETAILS,
            MAX(CONTENT_TYPE_RESULTS) AS CONTENT_TYPE_RESULTS,
            MAX(SHOPPING_CATEGORY_1) AS SHOPPING_CATEGORY_1,

```

```

MAX(SHOPPING_CATEGORY_2) AS SHOPPING_CATEGORY_2,
    MAX(SHOPPING_CATEGORY_3) AS SHOPPING_CATEGORY_3,
    MAX(KEYWORDS) AS KEYWORDS,
    MAX(SOURCE) AS SOURCE,
    COUNT(1) AS IS_TRANSLATION_AVAILABLE
    FOR LANGUAGE IN ('||v_for_lang_pivot_clause||'))' ;
l_items_sql_text VARCHAR2(30000) :=
'SELECT  ECC_SPEC_ID,
        THUMBNAI_IMAGE           ,
        SHOPPING_CATEGORY       ,
        SUPPLIER                 ,
        SUPPLIER_SITE            ,
        SUPPLIER_PART_NUM        ,
        SUPPLIER_PART_AUXID      ,
        INTERNAL_ITEM_NUM        ,
        SOURCE                    ,
        MANUFACTURER             ,
        MANUFACTURER_PART_NUM    ,
        PURCHASING_CATEGORY      ,
        DESCRIPTION               ,
        ITEM_REVISION            ,
        UNIT_OF_MEASURE          ,
        PRICE                     ,
        CURRENCY                  ,
        FUNCTIONAL_PRICE         ,
        FUNCTIONAL_CURRENCY      ,
        AVAILABILITY             ,
        LEAD_TIME                 ,
        UNSPSC                    ,
        ALIAS                     ,
        COMMENTS                  ,
        LONG_DESCRIPTION          ,
        ATTACHMENT_URL           ,
        SUPPLIER_URL              ,
        MANUFACTURER_URL         ,
        ORG_ID                    ,
        LANGUAGE                  ,
        ITEM_SOURCE_TEXT          ,
        CONTENT_TYPE_FILTER      ,
        dbms_lob.substr(ATTACHMENT, 4000, 1) ATTACHMENT ,
        ZONE_ID                   ,
        DISPLAY_PRICE             ,
        CONTENT_ID                ,
        CONTENT_URL               ,
        KEYWORDS                  ,
        CONTENT_TYPE              ,
        HIDE_SMARTFORM_ATTRS      ,
        HIDE_INFOCONTENT_ATTRS   ,
        PRICE_BREAK               ,
        HIDE_PRICE_BREAK          ,
        ITEM_RATING_FILTER        ,
        SUPPLIER_RATING_FILTER    ,
        HIDE_ADDTOCART            ,
        PUNCHOUT_MORE_DETAILS    ,
        OPEN_DESCRIPTION_IN_NEW_TAB ,
        ITEM_RATING               ,
        SUPPLIER_RATING           ,
        HIDE_CONTENT_TYPE         ,
        CONTENT_TYPE_RESULTS      ,
        SHOPPING_CATEGORY_1,
        SHOPPING_CATEGORY_2,
        SHOPPING_CATEGORY_3,
        ZONESB,
        ZONESP,
        ZONESI,
        PREFERRED_SUPPLIER_ICON,

```

```

ECO_FRIENDLY_SUPPLIER_ICON,
    GREEN_SUPPLIER_ICON,
    OFF_CONTRACT_ITEM_ICON,
    OVER_PRICED_ICON
FROM ICX_CAT_ECC_ITEMS_V ' ' ;
l_hierarchy_sql_text_pre VARCHAR2(30000) := 'SELECT * FROM(
    SELECT
        ecc_spec_id,
        shopping_category,
        CASE
            WHEN shopping_category_3 IS NOT NULL THEN
shopping_category_3
            WHEN shopping_category_2 IS NOT NULL THEN
shopping_category_2
            ELSE 'Others' END AS shopping_category_1 ,
        CASE
            WHEN shopping_category_3 IS NOT NULL AND
shopping_category_2 IS NOT NULL THEN shopping_category_2
            WHEN shopping_category_2 IS NOT NULL AND
shopping_category_1 IS NOT NULL THEN shopping_category_1
            ELSE shopping_category_1
            END AS shopping_category_2,
        CASE
            WHEN shopping_category_3 IS NOT NULL AND
shopping_category_2 IS NOT NULL AND shopping_category_1 IS NOT NULL
THEN shopping_category_1
            ELSE null
            END AS shopping_category_3,
        LANGUAGE
    FROM icx_ecc_category_hierarchy ' ;
l_hierarchy_sql_text_post VARCHAR2(30000) := ' ) PIVOT(
        MAX(SHOPPING_CATEGORY) AS SHOPPING_CATEGORY,
        MAX(SHOPPING_CATEGORY_1) AS SHOPPING_CATEGORY_1,
        MAX(SHOPPING_CATEGORY_2) AS SHOPPING_CATEGORY_2,
        MAX(SHOPPING_CATEGORY_3) AS SHOPPING_CATEGORY_3
    FOR LANGUAGE IN ('||v_for_lang_pivot_clause||'))' ;
l_zonesb_sql_text VARCHAR2(30000) := 'SELECT
    ecc_spec_id,
    zonesb
    from icx_cat_ecc_zones_b';
l_zonesp_sql_text VARCHAR2(30000) := 'SELECT
    ecc_spec_id,
    zonesp
    from icx_cat_ecc_zones_p';
l_zonesi_sql_text VARCHAR2(30000) := 'SELECT
    ecc_spec_id,
    zonesi
    from icx_cat_ecc_zones_i';

BEGIN

    IF ( p_operation = g_icx_ecc_del_op ) THEN
        l_sql_text := 'SELECT recordkey ECC_SPEC_ID
            FROM icx_cat_endeca_item_attributes
            WHERE attributekey LIKE '###DELETERECORD###'
;
        return l_sql_text;
    END IF;

    IF ( p_data_type = 'ITEMS' ) THEN
        IF ( p_load_type = 'FULL_LOAD' ) THEN
            l_sql_text := l_pivot_pre_sql || l_items_sql_text ||
l_pivot_post_sql;
        ELSE
            l_sql_text := l_pivot_pre_sql || l_items_sql_text ||
l_last_update_condition || l_pivot_post_sql;

```

```

END IF;

ELSIF( p_data_type = 'HIERARCHY' ) THEN
  IF ( p_load_type = 'FULL_LOAD' ) THEN
    l_sql_text := l_hierarchy_sql_text_pre ||
l_hierarchy_sql_text_post ;
  ELSE
    l_sql_text := l_hierarchy_sql_text_pre ||
l_last_update_condition || l_hierarchy_sql_text_post ;
  END IF;
ELSIF( p_data_type = 'ZONESB' ) THEN
  IF ( p_load_type = 'FULL_LOAD' ) THEN
    l_sql_text := l_zonesb_sql_text;
  ELSE
    l_sql_text := l_zonesb_sql_text || l_last_update_condition ;
  END IF;
ELSIF( p_data_type = 'ZONESP' ) THEN
  IF ( p_load_type = 'FULL_LOAD' ) THEN
    l_sql_text := l_zonesp_sql_text;
  ELSE
    l_sql_text := l_zonesp_sql_text || l_last_update_condition ;
  END IF;
ELSIF( p_data_type = 'ZONESI' ) THEN
  IF ( p_load_type = 'FULL_LOAD' ) THEN
    l_sql_text := l_zonesi_sql_text;
  ELSE
    l_sql_text := l_zonesi_sql_text || l_last_update_condition ;
  END IF;
END IF;

RETURN l_sql_text;
END get_sql_text;
PROCEDURE create_ecc_desc_view(
  p_category_id IN NUMBER )
IS
  l_view_sql      VARCHAR2(20000);
  l_category_id   NUMBER      := p_category_id;
  l_attr_list     VARCHAR2(2000) := '';
  l_attr_val_list VARCHAR2(2000) := '';
  l_attribute_name VARCHAR2(100) := '';
  CURSOR c_base_attrs
  IS
    SELECT UPPER(regexp_replace(KEY,'[-*# ]','_') ),
      DECODE(icon_indicator,1,'decode(PAV.'
        || stored_in_column
        || ' , ''Y'', ''true'', ''false'') ','PAV.'
        || stored_in_column) value
    FROM icx_cat_attributes_tl
    WHERE rt_category_id = l_category_id
    AND stored_in_table = 'PO_ATTRIBUTE_VALUES'
    AND stored_in_column IS NOT NULL
    AND attribute_id > 28
    AND language = 'US';
  CURSOR c_base_attrs_tl
  IS
    SELECT UPPER(regexp_replace(KEY,'[-*# ]','_') ),
      'PAV_TLP'
        || '.'
        || stored_in_column value
    FROM icx_cat_attributes_tl
    WHERE rt_category_id = l_category_id
    AND stored_in_table = 'PO_ATTRIBUTE_VALUES_TLP'
    AND stored_in_column IS NOT NULL
    AND attribute_id > 28
    AND language = 'US';
  l_attrs_tbl dbms_sql.varchar2_table;

```

```

l_attr_vals_tbl dbms_sql.varchar2_table;
BEGIN
    l_view_sql := 'create or replace view ICX_CAT_ECC_' || l_category_id
|| ' _VL (';
    l_view_sql := l_view_sql || 'ECC_SPEC_ID,';
    l_view_sql := l_view_sql || 'LANGUAGE,';
    l_view_sql := l_view_sql || 'ECC_LAST_UPDATE_DATE ';
    OPEN c_base_attrs;
    FETCH c_base_attrs BULK COLLECT INTO l_attrs_tbl,l_attr_vals_tbl;
    FOR i IN 1..l_attrs_tbl.count
    LOOP
        l_attribute_name := l_attrs_tbl(i);
        IF ( l_attribute_name = 'SIZE' ) THEN
            l_attribute_name := l_attribute_name || '_';
        END IF;
        l_attr_list := l_attr_list || ' , ' || l_attribute_name;
        l_attr_val_list := l_attr_val_list || ' , ' || l_attr_vals_tbl(i);
    END LOOP;
    CLOSE c_base_attrs;
    OPEN c_base_attrs_tl;
    FETCH c_base_attrs_tl BULK COLLECT INTO l_attrs_tbl,l_attr_vals_tbl;
    FOR i IN 1..l_attrs_tbl.count
    LOOP
        l_attribute_name := l_attrs_tbl(i);
        IF ( l_attribute_name = 'SIZE' ) THEN
            l_attribute_name := l_attribute_name || '_';
        END IF;
        l_attr_list := l_attr_list || ' , ' || l_attribute_name;
        l_attr_val_list := l_attr_val_list || ' , ' || l_attr_vals_tbl(i);
    END LOOP;
    CLOSE c_base_attrs_tl;
    l_view_sql := l_view_sql || l_attr_list || ') AS ';
    l_view_sql := l_view_sql || 'select items.inventory_item_id
|| ''#'' || items.po_line_id || ''#''
|| ''#'' || items.org_id ECC_SPEC_ID,';
    l_view_sql := l_view_sql || ' items.LANGUAGE,';
    l_view_sql := l_view_sql || ' GREATEST(items.
last_update_date, NVL(PAV.LAST_UPDATE_DATE, items.last_update_date), NVL
(items.last_update_date, PAV_TLP.LAST_UPDATE_DATE)) ECC_LAST_UPDATE_DATE
';
    l_view_sql := l_view_sql || l_attr_val_list;
    l_view_sql := l_view_sql || ' FROM
ICX_CAT_ITEMS_CTX_HDRS_TLP ITEMS,';
    l_view_sql := l_view_sql || ' PO_ATTRIBUTE_VALUES PAV, ' ||
' PO_ATTRIBUTE_VALUES_TLP PAV_TLP ';
    l_view_sql := l_view_sql || ' WHERE items.inventory_item_id
= PAV.inventory_item_id(+);
    l_view_sql := l_view_sql || ' AND items.po_line_id = PAV.
po_line_id(+);
    l_view_sql := l_view_sql || ' AND items.req_template_name =
PAV.req_template_name(+);
    l_view_sql := l_view_sql || ' AND items.
req_template_line_num = PAV.req_template_line_num(+);
    l_view_sql := l_view_sql || ' AND items.org_id = PAV.org_id
(+);
    l_view_sql := l_view_sql || ' AND items.inventory_item_id =
PAV_TLP.inventory_item_id(+);
    l_view_sql := l_view_sql || ' AND items.po_line_id =
PAV_TLP.po_line_id(+);
    l_view_sql := l_view_sql || ' AND items.req_template_name =
PAV_TLP.req_template_name(+);
    l_view_sql := l_view_sql || ' AND items.
req_template_line_num = PAV_TLP.req_template_line_num(+);
    l_view_sql := l_view_sql || ' AND items.org_id = PAV_TLP.
org_id(+);

```

```

l_view_sql := l_view_sql || ' AND items.language = PAV_TLP.
language(+)' ;
IF ( l_category_id <> 0 ) THEN
    l_view_sql := l_view_sql || ' AND items.ip_category_id = ' ;
    l_view_sql := l_view_sql || l_category_id ;
END IF ;
--      dbms_output.put_line(substr(l_view_sql, 0, 200));
--      dbms_output.put_line(substr(l_view_sql, 201, 200));
--      dbms_output.put_line(substr(l_view_sql, 401, 200));
--      dbms_output.put_line(substr(l_view_sql, 601, 200));
--      dbms_output.put_line(substr(l_view_sql, 801, 200));
EXECUTE IMMEDIATE l_view_sql ;
END create_ecc_desc_view ;
PROCEDURE create_ecc_desc_views
IS
    l_cat_id_list dbms_sql.varchar2_table ;
    CURSOR c_categories
    IS
        SELECT DISTINCT rt_category_id
        FROM icx_cat_attributes_tl
        WHERE EXISTS
            (SELECT 1 FROM icx_cat_items_ctx_hdrs_tlp WHERE
ip_category_id=rt_category_id
            UNION
            SELECT 1 FROM icx_cat_punchout_items WHERE
ip_category_id=rt_category_id
            )
        OR rt_category_id=0 ;
BEGIN
    icx_ecc_util_pvt.drop_ecc_desc_views ;
    OPEN c_categories ;
    FETCH c_categories BULK COLLECT INTO l_cat_id_list ;
    CLOSE c_categories ;
    FOR i IN 1..l_cat_id_list.count
    LOOP
        icx_ecc_util_pvt.create_ecc_desc_view(l_cat_id_list(i) ) ;
    END LOOP ;
END create_ecc_desc_views ;
PROCEDURE drop_ecc_desc_views
IS
    l_apps_schema_name varchar2(25) := 'APPS' ;
    CURSOR c_ecc_views
    IS
        SELECT DISTINCT view_name
        FROM all_views
        WHERE view_name LIKE 'ICX_CAT_ECC%VL'
        AND owner=l_apps_schema_name ;
    l_view_list dbms_sql.varchar2_table ;
BEGIN
    select oracle_username
    into l_apps_schema_name
    from fnd_oracle_userid
    where read_only_flag = 'U' ;

    OPEN c_ecc_views ;
    FETCH c_ecc_views BULK COLLECT INTO l_view_list ;
    CLOSE c_ecc_views ;
    FOR i IN 1..l_view_list.count
    LOOP
        EXECUTE IMMEDIATE 'drop view ' || l_view_list(i) ;
    END LOOP ;
END drop_ecc_desc_views ;
PROCEDURE get_desc_metadata_load_info(
    p_dataset_key IN VARCHAR2,
    p_dataset_attrs IN ecc_sec_field_values DEFAULT NULL,
    p_languages IN VARCHAR2 ,

```

```

x_ecc_ds_meta_rec OUT NOCOPY ecc_ds_meta_rec,
  x_return_status OUT NOCOPY VARCHAR2 )
IS
  l_apps_schema_name varchar2(25);
  l_desc_sql_text      VARCHAR2(4000) ;
  l_prec_sql_text      VARCHAR2(4000) ;
  l_prec_cat1_sql_text VARCHAR2(4000) := ' SELECT ' ||
  '''SHOPPING_CATEGORY_1''' TRIGGER_INSTANCE_ATTRIBUTE, '
  || '''SHOPPING_CATEGORY_2''' TARGET_INSTANCE_ATTRIBUTE, ' || '''*'''
  TRIGGER_ATTR_VALUE , ' || '''Y''' ENABLED_FLAG '
  || ' FROM dual ';
  l_prec_cat2_sql_text VARCHAR2(4000) := ' SELECT ' ||
  '''SHOPPING_CATEGORY_2''' TRIGGER_INSTANCE_ATTRIBUTE, '
  || '''SHOPPING_CATEGORY_3''' TARGET_INSTANCE_ATTRIBUTE, ' || '''*'''
  TRIGGER_ATTR_VALUE , ' || '''Y''' ENABLED_FLAG '
  || ' FROM dual ';
  l_prec_cat3_sql_text VARCHAR2(4000) := ' SELECT ' ||
  '''SHOPPING_CATEGORY_3''' TRIGGER_INSTANCE_ATTRIBUTE, '
  || '''SHOPPING_CATEGORY''' TARGET_INSTANCE_ATTRIBUTE, ' || '''*'''
  TRIGGER_ATTR_VALUE , ' || '''Y''' ENABLED_FLAG '
  || ' FROM dual ';
  query_det_arr ecc_query_det_arr_type := ecc_query_det_arr_type
(NULL);
BEGIN
  begin
    select oracle_username
    into l_apps_schema_name
    from fnd_oracle_userid
    where read_only_flag = 'U';
  exception
    when others then
      l_apps_schema_name := 'APPS';
  end;

  l_desc_sql_text := 'SELECT distinct Decode(UPPER(regexp_replace
(key, ''[-*# ]'', ''_'')) , ' 'SIZE'', ' 'SIZE_', UPPER(regexp_replace
(key, ''[-*# ]'', ''_'')) ) NAME,
                                ATTRIBUTE_NAME DISPLAY_NAME,
                                Decode(SubStr(stored_in_column,1,3) ,
' 'NUM'', ' 'NUMBER'', ' 'VARCHAR2'') TYPE,
                                NULL SCALE,
                                NULL PRECISION,
                                decode(rt_category_id, 0, ' 'Y'', ' 'N'')
SHOW_IN_IN_GUDIED_DISCOVERY,
                                ' 'REFINEMENTS'' GROUP_KEY,
                                Decode(stored_in_TABLE ,
' 'PO_ATTRIBUTE_VALUES_TLP'', ' 'Y'', ' 'N'') TRANSLATABLE,
                                LANGUAGE
                                FROM icx_cat_attributes_tl
                                WHERE stored_in_table IN
(' 'PO_ATTRIBUTE_VALUES'', ' 'PO_ATTRIBUTE_VALUES_TLP'')
                                AND STORED_IN_COLUMN IS NOT NULL
                                AND KEY NOT IN
(' 'PREFERRED_SUPPLIER_ICON'', ' 'ECO_FRIENDLY_SUPPLIER_ICON'',
' 'GREEN_SUPPLIER_ICON'', ' 'OFF_CONTRACT_ITEM_ICON'',
' 'OVER_PRICED_ICON'')
                                AND ATTRIBUTE_ID>28
                                AND LANGUAGE IN ( select COLUMN_VALUE
from TABLE(FND_ECC_UTIL_MLS_PVT.GET_LANG_FOR_METADATA('' ||
p_languages || '')))';

  l_prec_sql_text := ' SELECT ' || '''SHOPPING_CATEGORY_1'''
  TRIGGER_INSTANCE_ATTRIBUTE, '
  || ' UPPER(SubStrB(icx_ecc_util_pvt.makeNCName(ita.KEY), 1,30) )
  TARGET_INSTANCE_ATTRIBUTE, '
  || '''*''' TRIGGER_ATTR_VALUE , ' || '''Y''' ENABLED_FLAG '

```

```

|| ' FROM icx_cat_attributes_tl ita, ' || 'fnd_languages lang '
|| ' WHERE lang.language_code= ita.LANGUAGE ' || ' AND lang.
installed_flag =''B'''
|| ' AND STORED_IN_TABLE IN
(''PO_ATTRIBUTE_VALUES'', ''PO_ATTRIBUTE_VALUES_TLP'') '
|| ' AND STORED_IN_COLUMN like ''%ATTRIBUTE%'''
|| ' AND ita.rt_category_id <> 0 '
|| ' AND upper(icx_ecc_util_pvt.makeNCName(ita.KEY)) in '
|| ' (select column_name FROM all_tab_columns where table_name like
''ICX_CAT_ECC%VL'' and owner = '' ' || l_apps_schema_name || '' ' || '
)'
|| ' GROUP BY ita.KEY';

icx_ecc_util_pvt.create_ecc_desc_views;
icx_endeca_util_pkg.seed_icon_descriptors;
query_det_arr(1) := ecc_query_det_rec(l_desc_sql_text,
icx_ecc_util_pvt.g_icx_exc_att_op);
query_det_arr.extend ();
query_det_arr(2) := ecc_query_det_rec(l_prec_sql_text,
icx_ecc_util_pvt.g_icx_exc_pre_op);
query_det_arr.extend ();
query_det_arr(3) := ecc_query_det_rec(l_prec_cat1_sql_text,
icx_ecc_util_pvt.g_icx_exc_pre_op);
query_det_arr.extend ();
query_det_arr(4) := ecc_query_det_rec(l_prec_cat2_sql_text,
icx_ecc_util_pvt.g_icx_exc_pre_op);
query_det_arr.extend ();
query_det_arr(5) := ecc_query_det_rec(l_prec_cat3_sql_text,
icx_ecc_util_pvt.g_icx_exc_pre_op);
x_ecc_ds_meta_rec := ecc_ds_meta_rec(p_dataset_key, query_det_arr);
x_return_status := 'S';
END get_desc_metadata_load_info;
FUNCTION makencname(
    p_attribute_name IN VARCHAR2 )
RETURN VARCHAR2
IS
    l_attribute_name VARCHAR2(450);
BEGIN
    l_attribute_name
        := p_attribute_name;
    l_attribute_name
        := REPLACE(l_attribute_name, ' ', '_');
    l_attribute_name
        := REPLACE(l_attribute_name, '/', '_');
    l_attribute_name
        := REPLACE(l_attribute_name, '#', '_');
    l_attribute_name
        := REPLACE(l_attribute_name, ',', '_');
    IF ( upper(l_attribute_name) = 'SIZE' ) THEN
        l_attribute_name
            := l_attribute_name || '_';
    END IF;
    RETURN l_attribute_name;
END makencname;

PROCEDURE SUBMIT_ECC_DATA_LOAD(
    ERRBUF          OUT NOCOPY VARCHAR2 ,
    RETCODE         OUT NOCOPY VARCHAR2 ,
    p_system_name   IN VARCHAR2,
    p_load_type     IN VARCHAR2,
    p_languages     IN VARCHAR2,
    p_log_level     IN VARCHAR2,
    p_trace_enabled IN VARCHAR2
) IS
    l_app_short_name varchar2(10) := 'icx';
    l_request_id number;
    l_req_data varchar2(10);
    l_req_data1 varchar2(10);
    l_load_type varchar2(200);
BEGIN

    l_req_data := fnd_conc_global.request_data;

```

```

SELECT decode(p_load_type, 'FULL_LOAD', 'METADATA_LOAD', p_load_type)
INTO l_load_type FROM dual;

IF l_req_data = 'END' THEN
    fnd_file.put_line(FND_FILE.OUTPUT, 'Executed the sub request: ICX
ECC Data Load for load type : ' || p_load_type);
    fnd_file.put_line(FND_FILE.LOG, 'Executed the sub request: ICX ECC
Data Load for load type : ' || p_load_type);
    RETURN;
END IF;

IF (l_req_data = 'END-META' and p_load_type = 'FULL_LOAD') THEN
    fnd_file.put_line(FND_FILE.OUTPUT, 'Executed the sub request:
ICX ECC Data Load for load type : ' || l_load_type);
    fnd_file.put_line(FND_FILE.LOG, 'Executed the sub request: ICX
ECC Data Load for load type : ' || l_load_type);
    l_request_id := fnd_request.submit_request(
        application => 'FND',
        program      => 'ECCRUNDL',
        description  => 'ECC - Run Data Load',
        start_time   => sysdate,
        argument1    => p_system_name,
        argument2    => l_app_short_name,
        argument3    => null,
        argument4    => p_load_type,
        argument5    => p_languages,
        argument6    => p_trace_enabled,
        argument7    => p_log_level,
        sub_request  => true);

    fnd_file.put_line(FND_FILE.OUTPUT, 'ECC Load for load type : '
|| p_load_type || ' Job Request ID: ' || l_request_id);
    fnd_file.put_line(FND_FILE.LOG, 'ECC Load for load type : ' ||
p_load_type || ' Job Request ID: ' || l_request_id);

    IF l_request_id = 0 THEN
        --
        -- If request submission failed, exit with error.
        --
        ERRBUF := fnd_message.get;
        RETCODE := 2;
    ELSE
        --
        -- Here we set the globals to put the program into the
        -- PAUSED status on exit, and to save the state in
        -- request_data.
        --
        fnd_conc_global.set_req_globals(conc_status => 'PAUSED',
request_data => 'END');

        ERRBUF := 'Sub-Request submitted!';
        RETCODE := 0 ;

    END IF;
    RETURN;
END IF;
-- Submit l_load_type which can be either METADATA_LOAD or
INCREMENTAL_LOAD.
l_request_id := fnd_request.submit_request(
    application => 'FND',
    program      => 'ECCRUNDL',
    description  => 'ECC - Run Data Load',
    start_time   => sysdate,
    argument1    => p_system_name,
    argument2    => l_app_short_name,
    argument3    => null,

```

```

argument4 => l_load_type,
                                argument5    => p_languages,
                                argument6    => p_trace_enabled,
                                argument7    => p_log_level,
                                sub_request  => true);

    fnd_file.put_line(FND_FILE.OUTPUT,'ECC Load for load type : ' ||
l_load_type || ' Job Request ID:' || l_request_id);
    fnd_file.put_line(FND_FILE.LOG,'ECC Load for load type : ' ||
l_load_type || ' Job Request ID:' || l_request_id);

    IF l_request_id = 0 THEN
        --
        -- If request submission failed, exit with error.
        --
        ERRBUF := fnd_message.get;
        RETCODE := 2;
    ELSE
        --
        -- Here we set the globals to put the program into the
        -- PAUSED status on exit, and to save the state in
        -- request_data.
        --
        IF(l_load_type = 'METADATA_LOAD' AND p_load_type='FULL_LOAD')
THEN
            l_req_data1 := 'END-META';
        ELSE
            l_req_data1 := 'END';
        END IF;

        fnd_conc_global.set_req_globals(conc_status => 'PAUSED',request_data
=> l_req_data1);

        ERRBUF := 'Sub-Request submitted!';
        RETCODE := 0 ;

    END IF;
    RETURN;
EXCEPTION
WHEN OTHERS THEN
    RETCODE := 1;
    fnd_file.put_line(FND_FILE.OUTPUT,'OTHERS exception while submitting
: ICX ECC Data Load' || sqlerrm);
    fnd_file.put_line(FND_FILE.LOG,'OTHERS exception while submitting :
ICX ECC Data Load' || sqlerrm);

    END SUBMIT_ECC_DATA_LOAD;

END icx_ecc_util_pvt;
/
--show errors;
COMMIT;
EXIT;

```

Order Management の例

```

CREATE OR REPLACE PACKAGE BODY APPS.OE_ECC_UTIL_PVT
AS
    /* $Header: OEXUEXMP.pls 120.0.1 2019/07/22 06:14:03 jdoe noship $
    */
    /* Procedure GET_ECC_DATA_LOAD_INFO is the main procedure which
    is called during both full load and
    ** incremental load, for both ont-lines as well as ont-headers
    data sets.
    **
    ** GET_ECC_DATA_LOAD_INFO - Retrieves the data load details for
    the given data set key
    **
    ** IN parameters
    **   p_dataset_id - dataset key
    **   p_load_type - Indicating Full or incremental load
    (F/I/Custom).For custom there are no data load rules defined
    **   p_ds_last_success_run - Returns the last successful etl run
    for the dataset key
    ** OUT parameters
    **   x_ecc_ds_meta_rec - ecc_ds_meta_rec
    **   x_return_status - return status
    */
    -----
    PROCEDURE GET_ECC_DATA_LOAD_INFO (
        p_dataset_key          IN          VARCHAR2,
        p_load_type            IN          VARCHAR2,
        p_ds_last_success_run  IN          DATE,
        p_languages            IN          VARCHAR2,
        P_ADDDL_PARAMS         IN          ECC_SEC_FIELD_VALUES
    DEFAULT NULL,
        x_ecc_ds_meta_rec      OUT NOCOPY ecc_ds_meta_rec,
        x_return_status        OUT NOCOPY VARCHAR2)
    IS
        l_masstrans_d_sql_text CLOB;
        l_masstrans_i_sql_text CLOB;
        l_ont_header_full_text VARCHAR2 (30000);
        l_ont_header_hold_text VARCHAR2 (2500);
        l_base_language        VARCHAR2 (20);
        l_line_del_sql_text    VARCHAR2 (256)
            := 'select line_id ecc_spec_id from oe_ecc_deletion_map
where line_id is not null';
        l_hdr_del_sql_text    VARCHAR2 (256)
            := 'select distinct header_id ecc_spec_id from
oe_ecc_deletion_map where line_id is null';
        v_for_lang_pivot_clause VARCHAR2 (400)
            := FND_ECC_UTIL_MLS_PVT.GEN_ECC_MLS_PIVOT_FOR_LANG_CL (
                p_languages);
        l_ont_line_hold_text VARCHAR2 (4000)
            := 'SELECT * FROM
            (SELECT DISTINCT oegt.line_id ecc_spec_id,
                oegt.total_hold_name,
                oegt.header_hold_name,
                oegt.line_hold_name,
                oegt.GRAPH_ALERT_NAME,
                oegt.alert_text,
                oegt.language_code
            FROM oe_ecc_global_temp oegt
            ) pivot (max (GRAPH_ALERT_NAME) as GRAPH_ALERT_NAME,
                max(alert_text) as alert_text
            for language_code in ('

```

```

|| v_for_lang_pivot_clause
|| '))';
l_ont_full_sql_text VARCHAR2 (8000)
:= 'SELECT * FROM (SELECT
                                ECC_SPEC_ID,
ECC_LAST_UPDATE_DATE,ORDER_NUMBER,LINE_NUMBER,LINE_ID,HEADER_ID,PRODUC
T, LINE_QUANTITY,UNIT_SELLING_PRICE,
                                ORDER_UOM,
PRODUCT_DESCRIPTION,LINE_TYPE,PRICE_LIST,PRICE_LIST_ID,ORDER_DATE,
SHIP_FROM_ORG_ID,WAREHOUSE,SOURCE_TYPE,
                                SHIPMENT_METHOD,
CARRIER,SHIPMENT_PRIORITY,SHIPPING_INSTRUCTIONS,PACKING_INSTRUCTIONS,
FREIGHT_TERMS,SALESPERSON,CUSTOMER,
                                CUSTOMER_NUMBER,
SHIP_TO_CUSTOMER_NAME,SHIP_TO_CUSTOMER_NUMBER,SHIP_TO_CONTACT,
SHIP_TO_CONTACT_ID,SHIP_TO_ADDRESS,SHIP_TO_ORG_ID,
                                BILL_TO_CUSTOMER_NAME,BILL_TO_CUSTOMER_NUMBER,BILL_TO_CONTACT,
BILL_TO_CONTACT_ID,BILL_TO_ADDRESS,BILL_TO_ORG_ID,EXPECTED_DELAY,
                                REQUEST_DATE,
REQUEST_DATE_TYPE,SCHEDULE_SHIP_DATE,ACTUAL_SHIPMENT_DATE,
SCHEDULE_ARRIVAL_DATE,ACTUAL_ARRIVAL_DATE,
                                LATEST_ACCEPTABLE_DATE,FULFILLMENT_DATE,FLOW_STATUS_CODE,LINE_STATUS,
ITEM_TYPE_CODE,ORG_ID,OPERATING_UNIT,LINE_CATEGORY_CODE,
                                FULFILLED_FLAG,
LINE_TRAN_AMOUNT,OPEN_FLAG,TXN_CURRENCY,TXN_CURRENCY_CODE,
FUNC_CURRENCY_CODE,FUNC_CURRENCY,LINE_AMOUNT,
                                SHIP_TO_SITE,
SHIP_TO_LOCATION,SHIP_TO_STATE,SHIP_TO_COUNTRY,BILL_TO_SITE,
BILL_TO_CITY,BILL_TO_STATE,BILL_TO_COUNTRY,
                                LINE_LAST_UPDATE_DATE,PRIMARY_UOM,INVENTORY_ITEM_ID,SALES_CHANNEL,
ORDER_TYPE,STATUS,SHIPPED_QUANTITY,FULFILLED_QUANTITY,
                                INVOICED_QUANTITY,
SCHEDULE_STATUS,PROMISE_DATE,CUSTOMER_PO,PAYMENT_TERM,AGREEMENT_NAME,
SALES_AGREEMENT_NUMBER,SALES_AGREEMENT_LINE_NUMBER,
                                SUBINVENTORY,
PICK_STATUS,TRANSACTIONAL_ORDER_TOTAL,ORDER_TOTAL,RESERVED_QUANTITY,
PARTY_ID,PREDICTED_RETURN_REASON_CODE,
                                ANOMALIES,
LINE_HOLD_NAME,HEADER_HOLD_NAME,TOTAL_HOLD_NAME,ALERT_FLAG,ALERT_COUNT,
ALERT_TEXT,GRAPH_ALERT_NAME,DELAY_FLAG,DELAY_COUNT,
                                CONVERSION_RATE,
CONVERSION_TYPE_CODE,RESERVED_QTY,BOOKED_FLAG,ORDER_ALERT_COUNT,
ALERT_TYPE,HDR_ALERT_COUNT,
                                LANGUAGE_CODE,
ALERT_TYPE_CODE,TABLE_ALERT_FLAG,HDR_LAST_UPDATE_DATE,ON_ALERT,IS_OPEN,
IS_BOOKED,
                                LINE_QUANTITY_1,
SHIPPED_QUANTITY_1,CURRENCY,CSR_USER_NAME
                                FROM
                                OE_ECC_GLOBAL_TEMP )
                                PIVOT (
                                max(freight_terms)
as freight_terms,
                                max
(product_description) as product_description,
                                max(price_list) as
price_list,
                                max(line_type) as
line_type,
                                max(payment_term) as
payment_term,
                                max(ship_to_contact)
as ship_to_contact,

```

```

max(bill_to_contact) as bill_to_contact,
line_status,
as operating_unit,
as sales_channel,
as SHIPMENT_METHOD,
on_alert,
is_open,
is_booked,
order_type,
as agreement_name,
SOURCE_TYPE,
TXN_CURRENCY,
as func_currency,
as BILL_TO_COUNTRY,
as ship_to_country,
STATUS,
(GRAPH_ALERT_NAME) as GRAPH_ALERT_NAME,
max(alert_text) as alert_text ,
warehouse,
alert_type
( '
        || v_for_lang_pivot_clause
        || '))';

query_det_arr          ecc_query_det_arr_type
                        := ecc_query_det_arr_type
(NULL);

CURSOR lines IS SELECT * FROM OE_ECC_GLOBAL_TEMP;
l_current_header_id    NUMBER := 0;
l_current_total        NUMBER := 0;
l_func_total           NUMBER := 0;
l_hold_cnt             NUMBER := 0;
l_anam_cnt             NUMBER := 0;
l_reserved_qty2        NUMBER := 0;
l_return_status        VARCHAR2 (200);
l_currency_multiple    VARCHAR2 (200);
l_exists              VARCHAR2 (1);
l_hdr_total_sql_text   VARCHAR2 (2500);
l_ont_line_dff_text    VARCHAR2 (2500);
l_ont_header_dff_text  VARCHAR2 (2500);
l_alert_anomaly        VARCHAR2 (200);
l_alert_delay          VARCHAR2 (200);
l_alert_expected_delay VARCHAR2 (200);
l_alert_hold           VARCHAR2 (200);
l_alert_predictive_alert VARCHAR2 (200);
l_alert_pred_return_reason VARCHAR2 (200);
max(line_status) as
max(operating_unit)
max(sales_channel)
max(SHIPMENT_METHOD)
max(on_alert) as
max(is_open) as
max(is_booked) as
max(order_type) as
max(agreement_name)
max(SOURCE_TYPE) as
max(TXN_CURRENCY) as
max(func_currency)
max(BILL_TO_COUNTRY)
max(ship_to_country)
max(STATUS) as
max
max(warehouse) as
max(alert_type) as
for language_code in

```

```

l_alert_return_prediction    VARCHAR2 (200);
l_operating_unit            VARCHAR2 (100);
l_yes_label_meaning         VARCHAR2 (20);
l_no_label_meaning          VARCHAR2 (20);
l_hdr_hold_cnt              NUMBER;
lines_to_process             SYS_REFCURSOR;
l_ECC_BULK_LOAD_SIZE        NUMBER
    := NVL (fnd_profile.Value_wnps ('OM_ECC_BULK_LOAD_SIZE'),
10000);
l_alert_type_tbl            alert_type_tbl;
l_yes_no_tbl                alert_type_tbl;
l_ecc_tbl                   ecc_temp_tbl_type;
l_full_ecc_tbl              ecc_temp_tbl_type;

CURSOR alert_types IS
    SELECT meaning, language, lookup_code
    FROM fnd_lookup_values
    WHERE lookup_type = 'ONT_ALERT_TYPE'
    ORDER BY lookup_code;

CURSOR yes_no IS
    SELECT meaning, language, lookup_code
    FROM fnd_lookup_values
    WHERE      lookup_type = 'YES_NO'
              AND view_application_id = 660
              AND enabled_flag = 'Y'
    ORDER BY lookup_code;

CURSOR request_date_type_cur IS
    SELECT meaning, lookup_code
    FROM oe_lookups
    WHERE lookup_type = 'REQUEST_DATE_TYPE' AND enabled_flag =
'Y';

TYPE request_date_type_tbl IS TABLE OF request_date_type_cur%
ROWTYPE
    INDEX BY BINARY_INTEGER;

l_request_date_type_tbl request_date_type_tbl;
BEGIN
    -- insert_debug('Inside GET_ECC_DATA_LOAD_INFO procedure');
    -- set_db_trace('enable');
    -- insert_debug('l_ECC_BULK_LOAD_SIZE:'||l_ECC_BULK_LOAD_SIZE);
    -- insert_debug('p_dataset_key:'||p_dataset_key);
    -- insert_debug('p_load_type:'||p_load_type);
    -- insert_debug('p_ds_last_success_run:'||TO_CHAR
(p_ds_last_success_run,'DD-MON-YYYY HH24:MI:SS'));
    -- insert_debug('After truncating oe_ecc_global_temp');
    -- first delete all the records from oe_ecc_deletion_map that
are already processed.
    -- delete from oe_ecc_deletion_map WHERE processed = 'Y';
commented. trying truncation instead.
    -- get the meanings of alert types by opening the cursor

    OPEN alert_types;

    FETCH alert_types BULK COLLECT INTO l_alert_type_tbl;

    CLOSE alert_types;

    OPEN yes_no;

    FETCH yes_no BULK COLLECT INTO l_yes_no_tbl;

    CLOSE yes_no;

```

```

OPEN request_date_type_cur;

    FETCH request_date_type_cur BULK COLLECT INTO
l_request_date_type_tbl;

    CLOSE request_date_type_cur;

    IF p_dataset_key = 'ont-lines'
    THEN
        -- if data set is ont-lines, first truncate all the
necessary temp tables.
        trunc_temp_tables (p_load_type);

        IF (p_load_type = 'FULL_LOAD')
        THEN
            -- full load of
ont-lines

            OPEN lines_to_process FOR
            SELECT *
            FROM (-- Subquery Refactoring
            WITH
            requested_languages
            AS
                ( SELECT REGEXP_SUBSTR (
                    p_languages,
                    '^[^,]+',
                    1,
                    LEVEL) AS
LANGUAGE_CODE
                    FROM DUAL
                    CONNECT BY REGEXP_SUBSTR
(p_languages,
                    '^[^,]
+',
                    1,
                    LEVEL)
                    IS NOT NULL)
            -- Actual select query
            SELECT a.*
            FROM oe_ecc_order_lines_v a,
            requested_languages lang
            WHERE a.open_flag = 'Y'
            AND a.language_code IN
                (lang.LANGUAGE_CODE)
            AND org_id IS NOT NULL -- added

to avoid count mismatch

            UNION ALL
            SELECT /*+ index(OE_ECC_ORDER_LINES_V.1
OE_ORDER_LINES1_ENDECA_N1) */
                a.*
            FROM oe_ecc_order_lines_v a,
            requested_languages lang
            WHERE a.line_LAST_UPDATE_DATE >=
                SYSDATE
                - NVL (
FND_PROFILE.VALUE_WNPS (
                'OM_ECC_FULL_LOAD_DAYS'),
                90)
            AND a.open_flag = 'N'
            AND a.language_code IN
                (lang.LANGUAGE_CODE)
            AND a.org_id IS NOT NULL -- added

to avoid count mismatch
        )

        ORDER BY header_id;

```

```

ELSIF (p_load_type = 'INCREMENTAL_LOAD')
    THEN
        --incremental load of
ont-lines
        OPEN lines_to_process FOR
        --SELECT /*+ index(oeol.l
OE_ORDER_LINES1_ENDECA_N1) leading(oeol.l oeol.h) cardinality(oeol.l 10)
*/
        --FROM oe_ecc_order_lines_v oeol
        --WHERE ( ecc_last_update_date >=
p_ds_last_success_run ) AND language_code in (
        -- select regexp_substr(p_languages,'[^,]+', 1,
level) AS LANGUAGE_CODE from dual
        -- connect by regexp_substr(p_languages, '[^,]
+', 1, level) is not null)

        -- Subquery Refactoring
WITH
    requested_languages
AS
    ( SELECT REGEXP_SUBSTR (
        p_languages,
        '[^,]+',
        1,
        LEVEL) AS LANGUAGE_CODE
      FROM DUAL
      CONNECT BY REGEXP_SUBSTR (p_languages,
        '[^,]+',
        1,
        LEVEL)
        IS NOT NULL)
        -- Actual select query
        SELECT /*+ index(oeol.h OE_ORDER_LINES6_ENDECA_N1)
leading(oeol.h oeol.l) cardinality(oeol.l 10) */
        oeol.*
        FROM oe_ecc_order_lines_v oeol,
        requested_languages lang
        WHERE (hdr_last_update_date >=
p_ds_last_success_run)
        AND oeol.language_code IN (lang.
LANGUAGE_CODE)
        UNION
        SELECT /*+ index(oeol.l OE_ORDER_LINES1_ENDECA_N1)
leading(oeol.l oeol.h) cardinality(oeol.l 10) */
        oeol.*
        FROM oe_ecc_order_lines_v oeol,
        requested_languages lang
        WHERE (line_last_update_date >=
p_ds_last_success_run)
        AND oeol.language_code IN (lang.
LANGUAGE_CODE)
        UNION
        SELECT /*+ leading(oeol.tta oeol.l oeol.h)
cardinality(oeol.l 10) */
        oeol.*
        FROM oe_ecc_order_lines_v oeol,
        requested_languages lang
        WHERE (line_type_last_update_date >=
p_ds_last_success_run)
        AND oeol.language_code IN (lang.
LANGUAGE_CODE)
        UNION
        SELECT /*+ index(h OE_ORDER_LINES4_ENDECA_N1) */
        a.*
        FROM oe_ecc_order_lines_v a,
        oe_order_holds_all h,
        requested_languages lang

```

```

WHERE      a.line_id = h.line_id
              AND h.last_update_date >=
p_ds_last_success_run
              AND a.language_code IN (lang.LANGUAGE_CODE)
UNION
SELECT /*+ index(h OE_ORDER_LINES4_ENDECA_N1) */
      a.*
      FROM oe_ecc_order_lines_v a,
            oe_order_holds_all h,
            requested_languages lang
      WHERE      a.header_id = h.header_id
              AND h.last_update_date >=
p_ds_last_success_run
              AND a.language_code IN (lang.LANGUAGE_CODE)
UNION
SELECT /*+ index(p OE_PREDICTIONS_N2) */
      a.*
      FROM oe_ecc_order_lines_v a,
            oe_predictions p,
            requested_languages lang
      WHERE      a.line_id = p.line_id
              AND p.last_update_date >=
p_ds_last_success_run
              AND a.language_code IN (lang.LANGUAGE_CODE);
      END IF;
      -- load type is
      incremental load

      LOOP
      -- fetch
      lines_to_process
      FETCH lines_to_process
      BULK COLLECT INTO l_ecc_tbl
      LIMIT l_ECC_BULK_LOAD_SIZE;

      EXIT WHEN l_ecc_tbl.COUNT = 0;

      --insert_debug('l_ecc_tbl.COUNT is:'||l_ecc_tbl.COUNT);

      FOR i IN l_ecc_tbl.FIRST .. l_ecc_tbl.LAST
      LOOP
        l_alert_anomaly := NULL;
        l_alert_delay := NULL;
        l_alert_expected_delay := NULL;
        l_alert_hold := NULL;
        l_alert_predictive_alert := NULL;
        l_alert_pred_return_reason := NULL;
        l_alert_return_prediction := NULL;
        l_operating_unit := NULL;

        l_yes_label_meaning := NULL;
        l_no_label_meaning := NULL;

        FOR j IN l_alert_type_tbl.FIRST .. l_alert_type_tbl.
LAST
      LOOP
        IF l_alert_type_tbl (j).language =
          l_ecc_tbl (i).language_code
        THEN
          IF l_alert_type_tbl (j).lookup_code =
'ANOMALY'
          THEN
            l_alert_anomaly :=
              l_alert_type_tbl (j).meaning;
          ELSIF l_alert_type_tbl (j).lookup_code =
'DELAY'
          THEN
            l_alert_anomaly :=
              l_alert_type_tbl (j).meaning;
          ELSE
            l_alert_anomaly := NULL;
          END IF;
        END IF;
      END LOOP;
    END LOOP;
  END LOOP;

```

```

l_alert_delay := l_alert_type_tbl (j).meaning;
ELSIF l_alert_type_tbl (j).lookup_code =
    'EXP_DELAY'
THEN
    l_alert_expected_delay :=
        l_alert_type_tbl (j).meaning;
ELSIF l_alert_type_tbl (j).lookup_code =
'HOLD'
THEN
    l_alert_hold := l_alert_type_tbl (j).
meaning;
ELSIF l_alert_type_tbl (j).lookup_code =
    'PRED_ALERT'
THEN
    l_alert_predictive_alert :=
        l_alert_type_tbl (j).meaning;
ELSIF l_alert_type_tbl (j).lookup_code =
    'PRE_RET_REASON'
THEN
    l_alert_pred_return_reason :=
        l_alert_type_tbl (j).meaning;
ELSIF l_alert_type_tbl (j).lookup_code =
    'RETURN_PRED'
THEN
    l_alert_return_prediction :=
        l_alert_type_tbl (j).meaning;
END IF;
END IF;
END LOOP;

FOR j IN l_yes_no_tbl.FIRST .. l_yes_no_tbl.LAST
LOOP
    IF l_yes_no_tbl (j).language =
        l_ecc_tbl (i).language_code
    THEN
        IF l_yes_no_tbl (j).lookup_code = 'Y'
        THEN
            l_yes_label_meaning :=
                l_yes_no_tbl (j).meaning;
        ELSIF l_yes_no_tbl (j).lookup_code = 'N'
        THEN
            l_no_label_meaning :=
                l_yes_no_tbl (j).meaning;
        END IF;
    END IF;
END LOOP;

-- convert line total into functional currency
IF l_ecc_tbl (i).FUNC_CURRENCY_CODE IS NULL
THEN
    l_ecc_tbl (i).FUNC_CURRENCY_CODE :=
        oe_upgrade_misc.get_sob_currency (
            l_ecc_tbl (i).org_id);

    IF l_ecc_tbl (i).FUNC_CURRENCY_CODE IS NOT NULL
    THEN
        SELECT cur.NAME
        INTO l_ecc_tbl (i).FUNC_CURRENCY
        FROM fnd_currencies_tl cur
        WHERE cur.currency_code =
            l_ecc_tbl (i).FUNC_CURRENCY_CODE
        AND cur.LANGUAGE =
            l_ecc_tbl (i).LANGUAGE_CODE;
    END IF;
-- 'US';

```

```

END IF;

IF l_ecc_tbl (i).flow_status_code IS NOT NULL
THEN
    l_ecc_tbl (i).line_status :=
        oe_ecc_util_pvt.Get_Line_Status (
            l_ecc_tbl (i).line_id,
            l_ecc_tbl (i).flow_status_code,
            l_ecc_tbl (i).language_code);
END IF;

IF      l_ecc_tbl (i).FUNC_CURRENCY IS NULL
AND l_currency_multiple IS NULL
THEN
    BEGIN
        SELECT MESSAGE_TEXT
            INTO l_currency_multiple
            FROM fnd_new_messages
            WHERE message_name =
                'ONT_ENDECA_MULTIPLE_CURRENCIES';
    EXCEPTION
        WHEN OTHERS
        THEN
            l_currency_multiple := NULL;
    END;

    l_ecc_tbl (i).FUNC_CURRENCY :=
l_currency_multiple;
ELSIF      l_ecc_tbl (i).FUNC_CURRENCY IS NULL
AND l_currency_multiple IS NOT NULL
THEN
    l_ecc_tbl (i).FUNC_CURRENCY :=
l_currency_multiple;
END IF;

IF NVL (l_ecc_tbl (i).TXN_CURRENCY_CODE, 'abc') <>
NVL (l_ecc_tbl (i).FUNC_CURRENCY_CODE, 'abc')
THEN
    OE_UPGRADE_MISC.CONVERT_CURRENCY (
        l_ecc_tbl (i).line_tran_amount,
        l_ecc_tbl (i).TXN_CURRENCY_CODE,
        l_ecc_tbl (i).FUNC_CURRENCY_CODE,
        l_ecc_tbl (i).order_date,
        NULL,
        'Corporate',
        l_return_status,
        l_ecc_tbl (i).line_amount);
ELSE
    l_ecc_tbl (i).line_amount :=
        l_ecc_tbl (i).line_tran_amount;
END IF;

FOR j IN l_request_date_type_tbl.FIRST ..
        l_request_date_type_tbl.LAST
LOOP
    IF l_ecc_tbl (i).request_date_type =
        l_request_date_type_tbl (j).lookup_code
    THEN
        l_ecc_tbl (i).request_date_type :=
            l_request_date_type_tbl (j).meaning;
    END IF;
END LOOP;

-- insert_debug('l_ecc_tbl(i).
line_amount:'||l_ecc_tbl(i).line_amount);
-- insert_debug

```

```

('l_current_header_id'||l_current_header_id);
-- insert_debug('l_ecc_tbl(i).header_id:'||l_ecc_tbl
(i).header_id);
-- insert_debug('l_ecc_tbl(i).
order_total:'||l_func_total);
-- insert_debug('l_ecc_tbl(i).
transactional_order_total:'||l_ecc_tbl(i).transactional_order_total);
-- derive total holds
BEGIN
    SELECT LISTAGG (holds_tbl.HOLD_NAME, '|')
        WITHIN GROUP (ORDER BY holds_tbl.
line_id) total_hold_name
        INTO l_ecc_tbl (i).total_hold_name
        FROM ( SELECT /*+ QB_NAME
(HOLD_APPLY_INFO_SELECT2) */
                LISTAGG (hdf.name, '|')
                WITHIN GROUP (ORDER BY l.
line_id)
                HOLD_NAME,
                l.line_id
                line_id
        FROM oe_order_lines_all l,
             oe_order_holds_all h,
             oe_hold_sources_all hsr,
             oe_hold_definitions hdf,
             oe_lookups oel
        WHERE      h.header_id = l.header_id
                AND l.line_category_code =
'ORDER'
                AND h.released_flag = 'N'
                AND h.hold_source_id =
                hsr.hold_source_id
                AND hdf.hold_id = hsr.hold_id
                AND oel.lookup_type =
'HOLD_TYPE'
                AND oel.lookup_code = hdf.
type_code
                AND l.header_id =
                l_ecc_tbl (i).header_id
                AND ( h.line_id IS NULL
                OR h.line_id =
                l_ecc_tbl (i).
line_id)
                GROUP BY l.line_id) holds_tbl
        WHERE line_id = l_ecc_tbl (i).line_id
        GROUP BY line_id;
--insert_debug('After total holds
query:'||l_ecc_tbl (i).total_hold_name);

EXCEPTION
    WHEN OTHERS
    THEN
        --insert_debug('in total holds exception');
        l_ecc_tbl (i).total_hold_name := NULL;
END;

-- derive header holds
BEGIN
    SELECT LISTAGG (holds_tbl.HOLD_NAME, '|')
        WITHIN GROUP (ORDER BY holds_tbl.
line_id) header_hold_name
        INTO l_ecc_tbl (i).header_hold_name
        FROM ( SELECT LISTAGG (hdf.name, '|')
                WITHIN GROUP (ORDER BY lin.
line_id)

```

```

HOLD_NAME,

                                lin.line_id
                                line_id
FROM oe_order_headers_all hdr,
     oe_order_lines_all lin,
     oe_order_holds_all oha,
     oe_hold_sources_all hsr,
     oe_hold_definitions hdf,
     oe_lookups oel
WHERE   hdr.open_flag = 'Y'
AND oha.hold_source_id =
      hsr.hold_source_id
AND hsr.hold_id = hdf.hold_id
AND oha.header_id = hdr.

header_id

                                AND oha.released_flag = 'N'
                                AND hdr.header_id = lin.

header_id

                                AND lin.line_category_code =

'ORDER'

                                AND oha.line_id IS NULL
                                AND oel.lookup_type =

'HOLD_TYPE'

                                AND oel.lookup_code = hdf.

type_code

                                AND hdr.header_id =
                                l_ecc_tbl (i).header_id
                                GROUP BY lin.line_id) holds_tbl
                                WHERE line_id = l_ecc_tbl (i).line_id
                                GROUP BY line_id;
--insert_debug('in header holds :'||l_ecc_tbl(i).
header_hold_name);

EXCEPTION
  WHEN OTHERS
  THEN
    --insert_debug('in header holds exception');
    l_ecc_tbl (i).header_hold_name := NULL;
END;

-- derive line holds

BEGIN
  SELECT /*+ QB_NAME(HOLD_APPLY_INFO_SELECT2) */
        LISTAGG (hdf.name, '|')
        WITHIN GROUP (ORDER BY l.line_id)

HOLD_NAME

                                INTO l_ecc_tbl (i).line_hold_name
FROM oe_order_lines_all l,
     oe_order_holds_all h,
     oe_hold_sources_all hsr,
     oe_hold_definitions hdf,
     oe_lookups oel
WHERE   h.header_id = l.header_id
AND l.line_id = h.line_id
AND l.line_category_code = 'ORDER'
AND h.released_flag = 'N'
AND h.hold_source_id = hsr.

hold_source_id

                                AND hdf.hold_id = hsr.hold_id
                                AND oel.lookup_type = 'HOLD_TYPE'
                                AND oel.lookup_code = hdf.type_code
                                AND l.line_id = l_ecc_tbl (i).line_id
                                GROUP BY l.line_id;
--insert_debug('in line holds:'||l_ecc_tbl(i).
line_hold_name);

```

```

EXCEPTION
    WHEN OTHERS
    THEN
        --insert_debug('in line holds exception');
        l_ecc_tbl (i).line_hold_name := NULL;
END;

-- derive line level anomalies
BEGIN
    SELECT LISTAGG (ol.meaning, '|')
           WITHIN GROUP (ORDER BY op.line_id)
anomalies
    INTO l_ecc_tbl (i).anomalies
    FROM fnd_lookup_values ol, oe_predictions op
    WHERE      op.anomaly_code = ol.lookup_code
              AND ol.lookup_type = 'ONT_ANOMALY_CODES'
              AND ol.language = l_ecc_tbl (i).
LANGUAGE_CODE --'US'//saadepu
              AND op.line_id = l_ecc_tbl (i).line_id;
EXCEPTION
    WHEN OTHERS
    THEN
        --insert_debug('in Anomalies exception');
        l_ecc_tbl (i).anomalies := NULL;
END;

-- derive return predictions
BEGIN
    SELECT ol.meaning --
PREDICTED_RETURN_REASON_CODE
    INTO l_ecc_tbl (i).
PREDICTED_RETURN_REASON_CODE
    FROM fnd_lookup_values ol, oe_predictions op
    WHERE      op.predicted_return_reason_code =
              ol.lookup_code
              AND ol.lookup_type =
'ONT_PRED_RETURN_REASON'
              AND ol.language = l_ecc_tbl (i).
LANGUAGE_CODE --'US'//saadepu
              AND op.line_id = l_ecc_tbl (i).line_id;
EXCEPTION
    WHEN OTHERS
    THEN
        --insert_debug('in
PREDICTED_RETURN_REASON_CODE exception');
        l_ecc_tbl
        (i).PREDICTED_RETURN_REASON_CODE :
=
        NULL;
END;

--insert_debug('l_ecc_tbl(i).LINE_ID:'||l_ecc_tbl
(i).LINE_ID);
--insert_debug('l_ecc_tbl(i).
SHIP_FROM_ORG_ID:'||l_ecc_tbl(i).SHIP_FROM_ORG_ID);
OE_LINE_UTIL.Get_Reserved_Quantities (
    p_header_id          => l_ecc_tbl (i).
header_id,
    p_line_id            => l_ecc_tbl (i).LINE_ID,
    p_org_id              => l_ecc_tbl (i).
SHIP_FROM_ORG_ID,
    x_reserved_quantity  => l_ecc_tbl (i).
reserved_qty,
    x_reserved_quantity2 => l_reserved_qty2);
--insert_debug('l_ecc_tbl(i).
reserved_qty:'||l_ecc_tbl(i).reserved_qty);

```

```

--insert_debug('l_ecc_tbl(i).ANOMALIES:'||l_ecc_tbl(i).ANOMALIES);
--insert_debug('l_ecc_tbl(i).
PREDICTED_RETURN_REASON_CODE:'||l_ecc_tbl(i).
PREDICTED_RETURN_REASON_CODE);
--insert_debug('l_ecc_tbl(i).
total_hold_name:'||l_ecc_tbl(i).total_hold_name);
--insert_debug('l_ecc_tbl(i).
header_hold_name:'||l_ecc_tbl(i).header_hold_name);
--insert_debug('l_ecc_tbl(i).
line_hold_name:'||l_ecc_tbl(i).line_hold_name);

IF l_ecc_tbl (i).expected_delay <= 0
THEN -- negative expected delay essentially means
no delay

    l_ecc_tbl (i).expected_delay := NULL;
END IF;

IF      ( l_ecc_tbl (i).ANOMALIES IS NOT NULL
OR l_ecc_tbl (i).
PREDICTED_RETURN_REASON_CODE
IS NOT NULL
OR l_ecc_tbl (i).total_hold_name IS NOT NULL
OR ( l_ecc_tbl (i).expected_delay IS NOT
NULL
AND l_ecc_tbl (i).expected_delay > 0
AND NVL (l_ecc_tbl (i).shipped_quantity,
0) =
0))

    AND l_ecc_tbl (i).open_flag = 'Y'
THEN
    l_ecc_tbl (i).alert_flag := 'Y';
    l_ecc_tbl (i).table_alert_flag := 'ecc_warning';
ELSE
    l_ecc_tbl (i).alert_flag := 'N';
    l_ecc_tbl (i).table_alert_flag := 'ecc_blank';
END IF;

--insert_debug('l_ecc_tbl(i).alert_flag:'||l_ecc_tbl
(i).alert_flag);

l_ecc_tbl (i).alert_count := 0;

IF l_ecc_tbl (i).anomalies IS NOT NULL
THEN
    l_anam_cnt := 0;

    SELECT      REGEXP_COUNT (l_ecc_tbl (i).anomalies,
',,')
                + 1
    INTO l_anam_cnt
    FROM DUAL;

    l_ecc_tbl (i).alert_count :=
        l_ecc_tbl (i).alert_count + l_anam_cnt;
    l_ecc_tbl (i).alert_type :=
        l_ecc_tbl (i).alert_type
        || l_alert_predictive_alert; --
l_alert_type_tbl(5).meaning; --'Predictive Alerts';
    l_ecc_tbl (i).alert_type_code := 'PA';
--'Predictive Alerts';
    l_ecc_tbl (i).alert_text := l_ecc_tbl (i).
anomalies; --(Anomaly:Quantity Anomaly)
    l_ecc_tbl (i).graph_alert_name :=
l_alert_anomaly; --l_alert_type_tbl(1).meaning;
--l_ecc_tbl(i).anomalies;
END IF;

```

```

NULL
        IF l_ecc_tbl (i).PREDICTED_RETURN_REASON_CODE IS NOT
        THEN
            l_ecc_tbl (i).alert_count :=
                l_ecc_tbl (i).alert_count + 1;
            IF l_ecc_tbl (i).alert_type IS NOT NULL
            THEN
                l_ecc_tbl (i).alert_type :=
                    l_ecc_tbl (i).alert_type
                    || '|'
                    || l_alert_predictive_alert; --
l_alert_type_tbl(5).meaning; --Predictive Alerts
                l_ecc_tbl (i).alert_type_code := 'PA'; --
Predictive Alerts
            ELSE
                l_ecc_tbl (i).alert_type :=
                    l_alert_predictive_alert; --
l_alert_type_tbl(5).meaning;
                l_ecc_tbl (i).alert_type_code := 'PA';
            END IF;
            --l_ecc_tbl(i).alert_type IS
NOT NULL

            IF l_ecc_tbl (i).alert_text IS NOT NULL
            THEN
                l_ecc_tbl (i).alert_text :=
                    l_ecc_tbl (i).alert_text
                    || '|'
                    || l_alert_pred_return_reason --
l_alert_type_tbl(6).meaning
                    || ': '
                    || l_ecc_tbl (i).
PREDICTED_RETURN_REASON_CODE;
            ELSE
                l_ecc_tbl (i).alert_text :=
                    l_alert_pred_return_reason --
l_alert_type_tbl(6).meaning
                    || ': '
                    || l_ecc_tbl (i).
PREDICTED_RETURN_REASON_CODE;
            END IF;
            --l_ecc_tbl(i).alert_text IS
NOT NULL

            IF l_ecc_tbl (i).graph_alert_name IS NOT NULL
            THEN
                l_ecc_tbl (i).graph_alert_name :=
                    l_ecc_tbl (i).graph_alert_name
                    || '|'
                    || l_alert_return_prediction; --
l_alert_type_tbl(7).meaning; --Return Prediction
            ELSE
                l_ecc_tbl (i).graph_alert_name :=
                    l_alert_return_prediction; --
l_alert_type_tbl(7).meaning; --return prediction--l_ecc_tbl(i).
PREDICTED_RETURN_REASON_CODE;
            END IF;
            END IF; --l_ecc_tbl(i).PREDICTED_RETURN_REASON_CODE
IS NOT NULL

/*hdr_alert_count stores the number of holds at
header level*/

IF l_ecc_tbl (i).header_hold_name IS NOT NULL
THEN
    l_hdr_hold_cnt := 0;

```

```

SELECT      REGEXP_COUNT (
                                l_ecc_tbl (i).header_hold_name,
                                '\|')
                                + 1
                                INTO l_hdr_hold_cnt
                                FROM DUAL;

                                l_ecc_tbl (i).hdr_alert_count :=
                                NVL (l_ecc_tbl (i).hdr_alert_count, 0)
                                + l_hdr_hold_cnt;
                                --insert_debug('After hdr_alert_count check
l_hdr_hold_cnt is:'||l_hdr_hold_cnt);
                                --insert_debug('After hdr_alert_count check count
is:'||l_ecc_tbl(i).hdr_alert_count);

                                END IF;          -- l_ecc_tbl(i).header_hold_name IS
NOT NULL

                                IF l_ecc_tbl (i).line_hold_name IS NOT NULL
                                THEN
                                    l_hold_cnt := 0;

                                    SELECT REGEXP_COUNT (l_ecc_tbl (i).
line_hold_name,
                                                                '\|')
                                                                + 1
                                                                INTO l_hold_cnt
                                                                FROM DUAL;

                                    --insert_debug('After line_alert_count check
l_hold_cnt is:'||l_hold_cnt);

                                    l_ecc_tbl (i).alert_count :=
                                    l_ecc_tbl (i).alert_count + l_hold_cnt;

                                    --insert_debug('After line_alert_count check
count is:'||l_ecc_tbl(i).alert_count);

                                    IF l_ecc_tbl (i).alert_text IS NOT NULL
                                    THEN
                                        l_ecc_tbl (i).alert_text :=
                                        l_ecc_tbl (i).alert_text
                                        || '|'
                                        || l_alert_hold --l_alert_type_tbl(4).
meaning
                                        || ': '
                                        || l_ecc_tbl (i).line_hold_name;
                                    ELSE
                                        l_ecc_tbl (i).alert_text :=
                                        l_alert_hold --l_alert_type_tbl(4).
meaning
                                        || ': '
                                        || l_ecc_tbl (i).line_hold_name;
                                    END IF;          --l_ecc_tbl(i).alert_text IS
NOT NULL

                                END IF;          -- l_ecc_tbl(i).line_hold_name IS
NOT NULL

                                IF l_ecc_tbl (i).total_hold_name IS NOT NULL
                                THEN
                                    IF l_ecc_tbl (i).alert_type IS NOT NULL
                                    THEN
                                        l_ecc_tbl (i).alert_type :=
                                        l_ecc_tbl (i).alert_type
                                        || '|'
                                        || l_alert_hold; --l_alert_type_tbl(4).

```

```

meaning;

ELSE
    l_ecc_tbl (i).alert_type := l_alert_hold; --
l_alert_type_tbl(4).meaning;
END IF;      -- l_ecc_tbl(i).alert_type IS
NOT NULL

END IF;

IF l_ecc_tbl (i).total_hold_name IS NOT NULL
THEN
    IF l_ecc_tbl (i).graph_alert_name IS NOT NULL
    THEN
        l_ecc_tbl (i).graph_alert_name :=
            l_ecc_tbl (i).graph_alert_name
            || ' '
            || l_ecc_tbl (i).total_hold_name;
    ELSE
        l_ecc_tbl (i).graph_alert_name :=
            l_ecc_tbl (i).total_hold_name;
    END IF;
END IF;

IF      l_ecc_tbl (i).expected_delay IS NOT NULL
AND l_ecc_tbl (i).expected_delay > 0
AND NVL (l_ecc_tbl (i).shipped_quantity, 0) = 0
THEN
    l_ecc_tbl (i).alert_count :=
        l_ecc_tbl (i).alert_count + 1;

    IF l_ecc_tbl (i).alert_type IS NOT NULL
    THEN
        l_ecc_tbl (i).alert_type :=
            l_ecc_tbl (i).alert_type
            || ' '
            || l_alert_expected_delay; --
l_alert_type_tbl(3).meaning;
    ELSE
        l_ecc_tbl (i).alert_type :=
            l_alert_expected_delay; --
l_alert_type_tbl(3).meaning;
    END IF;      -- l_ecc_tbl(i).alert_type IS
NOT NULL

    IF l_ecc_tbl (i).alert_text IS NOT NULL
    THEN
        l_ecc_tbl (i).alert_text :=
            l_ecc_tbl (i).alert_text
            || ' '
            || l_alert_expected_delay; --
l_alert_type_tbl(3).meaning;
    ELSE
        l_ecc_tbl (i).alert_text :=
            l_alert_expected_delay; --
l_alert_type_tbl(3).meaning;
    END IF;      -- l_ecc_tbl(i).alert_text IS NOT
NULL.

    IF l_ecc_tbl (i).graph_alert_name IS NOT NULL
    THEN
        l_ecc_tbl (i).graph_alert_name :=
            l_ecc_tbl (i).graph_alert_name
            || ' '
            || l_alert_expected_delay; --
l_alert_type_tbl(3).meaning;
    ELSE
        l_ecc_tbl (i).graph_alert_name :=

```

```

l_alert_expected_delay; --l_alert_type_tbl(3).meaning;
    END IF;
    END IF; -- end if for expected delay is not null
condition

    --insert_debug('l_ecc_tbl(i).
alert_count:'||l_ecc_tbl(i).alert_count);
    --insert_debug('l_ecc_tbl(i).
expected_delay:'||l_ecc_tbl(i).expected_delay);
    IF ( l_ecc_tbl (i).expected_delay IS NOT NULL
        AND l_ecc_tbl (i).expected_delay > 0
        AND NVL (l_ecc_tbl (i).shipped_quantity, 0) = 0)
    THEN
        l_ecc_tbl (i).delay_flag := 'Y';
        l_ecc_tbl (i).delay_count := 0;
        IF ( l_ecc_tbl (i).expected_delay IS NOT NULL
            AND l_ecc_tbl (i).expected_delay > 0)
        THEN
            l_ecc_tbl (i).delay_count :=
                l_ecc_tbl (i).delay_count + 1;
        END IF;
    END IF;

    IF l_ecc_tbl (i).csr_user_id IS NOT NULL
    THEN
        BEGIN
            SELECT p.full_name
            INTO l_ecc_tbl (i).CSR_USER_NAME
            FROM per_all_people_f p, fnd_user u
            WHERE      p.person_id = u.employee_id
                    AND TRUNC (SYSDATE) BETWEEN p.
effective_start_date
                                AND p.
effective_end_date
                                AND u.user_id = l_ecc_tbl (i).
csr_user_id;
        EXCEPTION
            WHEN OTHERS
            THEN
                --insert_debug('exception while fetching
CSR_USER_NAME ');
                l_ecc_tbl (i).CSR_USER_NAME := NULL;
        END;
    END IF;

    --insert_debug('l_ecc_tbl(i).delay_flag:'||l_ecc_tbl
(i).delay_flag );
    --insert_debug('l_ecc_tbl(i).
delay_count:'||l_ecc_tbl(i).delay_count);

    IF (p_load_type = 'INCREMENTAL_LOAD')
    THEN
        -- incremental load of
ont-lines
        -- first check if record already exists. if
record already exists, just update the alert count and total
        -- for that record.

        /*BEGIN

            SELECT 'Y'
            INTO l_exists
            FROM oe_ecc_header_totals
            WHERE line_id=l_ecc_tbl(i).line_id;*/

```

```

UPDATE oe_ecc_header_totals
    SET header_id = l_ecc_tbl (i).header_id,
        line_id = l_ecc_tbl (i).line_id,
        alert_count = l_ecc_tbl (i).alert_count,
        hdr_alert_count =
            l_ecc_tbl (i).hdr_alert_count,
        tran_amount = l_ecc_tbl (i).
line_tran_amount,
        func_amount = l_ecc_tbl (i).line_amount,
        language_code = l_ecc_tbl (i).
language_code
    WHERE      line_id = l_ecc_tbl (i).line_id
              AND language_code =
                  l_ecc_tbl (i).language_code;
IF SQL%ROWCOUNT = 0
THEN
    INSERT INTO oe_ecc_header_totals (
        HEADER_ID,
        LINE_ID,
        ALERT_COUNT,
        HDR_ALERT_COUNT,
        tran_amount,
        func_amount,
        language_code
    )
    VALUES (l_ecc_tbl (i).header_id,
            l_ecc_tbl (i).line_id,
            l_ecc_tbl (i).alert_count,
            l_ecc_tbl (i).hdr_alert_count,
            l_ecc_tbl
            (i).line_tran_amount,
            l_ecc_tbl (i).line_amount,
            l_ecc_tbl (i).language_code
            );

END IF;

DELETE FROM oe_ecc_header_totals
    WHERE line_id IN (SELECT line_id
                      FROM
oe_ecc_deletion_map
                      WHERE line_id IS NOT
NULL);

/*EXCEPTION
WHEN NO_DATA FOUND THEN
    -- line does not exist already.Looks like line
got created.So insert into oe_ecc_header_totals

END ;*/

-- also mark the header into oe_ecc_headers
table, for headers data set to pick the records and process.

IF l_current_header_id <> l_ecc_tbl (i).
header_id
THEN
    -- added the IF condition to minimize the
number of inserts into temp table.
    INSERT INTO oe_ecc_headers (HEADER_ID)
        VALUES (l_ecc_tbl (i).header_id);

    l_current_header_id := l_ecc_tbl (i).
header_id;

END IF;
END IF;

```

```

-- set the lookup based values for open, booked
and alert flags

        IF NVL (l_ecc_tbl (i).alert_flag, NULL) = 'Y'
        THEN
            l_ecc_tbl (i).on_alert := l_yes_label_meaning;
--l_yes_no_tbl(2).meaning;
        ELSE
            l_ecc_tbl (i).on_alert := l_no_label_meaning; --
l_yes_no_tbl(1).meaning;
        END IF;

        IF NVL (l_ecc_tbl (i).open_flag, NULL) = 'Y'
        THEN
            l_ecc_tbl (i).is_open := l_yes_label_meaning; --
l_yes_no_tbl(2).meaning;
        ELSE
            l_ecc_tbl (i).is_open := l_no_label_meaning; --
l_yes_no_tbl(1).meaning;
        END IF;

        IF NVL (l_ecc_tbl (i).booked_flag, NULL) = 'Y'
        THEN
            l_ecc_tbl (i).is_booked := l_yes_label_meaning;
--l_yes_no_tbl(2).meaning;
        ELSE
            l_ecc_tbl (i).is_booked := l_no_label_meaning;
--l_yes_no_tbl(1).meaning;
        END IF;
    END LOOP; -- inner loop for processing records bulk
collected.

    FORALL idx IN INDICES OF l_ecc_tbl
        INSERT INTO oe_ecc_global_temp
            VALUES l_ecc_tbl (idx);

    COMMIT;

    /*if it is full load, insert all the record totals
into oe_ecc_header_totals table */

    IF p_load_type = 'FULL_LOAD'
    THEN
-- full load of
ont-lines
        l_full_ecc_tbl := l_ecc_tbl;

        FOR j IN l_full_ecc_tbl.FIRST .. l_full_ecc_tbl.LAST
        LOOP
            IF l_full_ecc_tbl (j).open_flag = 'N'
            THEN
                l_full_ecc_tbl (j).line_amount := 0;
            END IF;
        END LOOP;

        FORALL idx IN INDICES OF l_ecc_tbl
            INSERT INTO oe_ecc_header_totals
                VALUES (l_full_ecc_tbl (idx).header_id,
                    l_full_ecc_tbl (idx).line_id,
                    l_full_ecc_tbl (idx).alert_count,
                    l_full_ecc_tbl (idx).
hdr_alert_count,
                    l_full_ecc_tbl (idx).
line_tran_amount,
                    l_full_ecc_tbl (idx).line_amount,
                    l_full_ecc_tbl (idx).language_code

```

```

);

        DELETE FROM oe_ecc_header_totals
            WHERE line_id IN (SELECT line_id
                            FROM oe_ecc_deletion_map
                            WHERE line_id IS NOT NULL);

    END IF;

    COMMIT;

    --insert_debug('After inserting into temp table');

    l_ecc_tbl.DELETE ();
    --insert_debug('After deleting data from plsqli table. Count
is:'||l_ecc_tbl.count);

    END LOOP;                                -- outer loop for bulk
collect.

    SELECT LANGUAGE_CODE
        INTO l_base_language
        FROM FND_LANGUAGES
        WHERE INSTALLED_FLAG = 'B';

    --update order totals at one go
    UPDATE oe_ecc_global_temp oegt
        SET (oegt.transactional_order_total, oegt.order_total) =
            (SELECT SUM (oehl.tran_amount), SUM (oehl.
func_amount)
            FROM oe_ecc_header_totals oehl
            WHERE      oegt.header_id = oehl.header_id
                    AND oehl.language_code = l_base_language

    );

    query_det_arr.EXTEND (1);
    query_det_arr (1) :=
        ecc_query_det_rec (l_ont_full_sql_text,
G_ONT_ECC_UPS_OP);

    IF (p_load_type = 'INCREMENTAL_LOAD')
    THEN
        query_det_arr.EXTEND (2);
        query_det_arr (2) :=
            ecc_query_det_rec (l_line_del_sql_text,
G_ONT_ECC_DEL_OP);
        -- query_det_arr.extend(3);
        -- query_det_arr(3) := ecc_query_det_rec
(l_ont_line_hold_text,G_ONT_ECC_RPL_OP);

    END IF;

    x_ecc_ds_meta_rec :=
        ecc_ds_meta_rec (p_dataset_key, query_det_arr);
    x_return_status := 'S';
    ELSIF p_dataset_key = 'ont-headers'
    THEN
        IF p_load_type = 'FULL_LOAD'
        THEN                                -- full load of
ontheaders
            --insert_debug('Full load for headers data set');

            l_ont_header_full_text :=

```

```

' SELECT * FROM (
                                SELECT order_number,
customer,customer_number,ship_to_location, order_date, status,
order_alert_count,
                                order_alert_flag,
ecc_spec_id,header_id,header_hold_name,customer_po,request_date,
order_type,
                                org_id,
order_currency,language_code,
                                (SELECT p.full_name
FROM per_all_people_f p, fnd_user u WHERE p.person_id = u.employee_id
AND
                                trunc(sysdate)
between p.effective_start_date and p.effective_end_date
                                AND u.user_id =
csr_user_id) csr_user_name
                                FROM
                                (
                                SELECT oegt.
                                order_number,
                                oegt.customer,
                                oegt.
                                customer_number,
                                ship_su.LOCATION
                                ship_to_location,
                                oegt.order_date,
                                oegt.status,
                                SUM (oegt.
                                alert_count) + MAX(nvl(oegt.hdr_alert_count,0)) order_alert_count,
                                DECODE (SUM (oegt.
                                alert_count) + MAX(nvl(oegt.hdr_alert_count,0)),
                                0,'ecc_blank','ecc_warning') order_alert_flag,
                                oegt.header_id
                                ecc_spec_id,
                                oegt.header_id,
                                oegt.
                                header_hold_name,
                                oh.cust_po_number
                                customer_po,
                                oh.request_date ,
                                oegt.order_type,
                                oegt.org_id,
                                oegt.func_currency
                                order_currency,
                                oh.csr_user_id,
                                oegt.language_code
                                language_code
                                FROM
                                oe_ecc_global_temp
                                oe_order_headers_all
                                oegt,
                                oh,
                                hz_cust_site_uses_all ship_su
                                WHERE
                                oh.header_id
                                =oegt.header_id AND
                                oh.ship_to_org_id =
                                ship_su.site_use_id(+) AND
                                oegt.org_id IS NOT
                                NULL
                                GROUP BY
                                oegt.header_id,
                                oegt.order_number,
                                oegt.customer,
                                oegt.

```

```

customer_number,

                                oegt.order_date,
                                oegt.status,
                                ship_su.LOCATION,
                                oegt.

header_hold_name,

                                oh.cust_po_number,
                                oh.request_date ,
                                oegt.

hdr_alert_count,

                                oegt.order_type,
                                oegt.org_id,
                                oegt.func_currency,
                                oh.csr_user_id,
                                oegt.language_code)
                                )
                                PIVOT (
                                    max(STATUS) as STATUS,
                                    max(ORDER_CURRENCY) as
ORDER_CURRENCY,
                                    max(ORDER_TYPE) as
ORDER_TYPE
                                    for language_code in (
                                        || v_for_lang_pivot_clause
                                        || '))';

                                l_hdr_total_sql_text :=
                                'select

                                oh.header_id ecc_spec_id,
                                oe_ecc_util_pvt.get_order_total
                                (oh.header_id,

                                oh.transactional_curr_code,
                                oh.org_id,
                                fnd_profile.Value
                                (''OM_ECC_DISPLAY_CURRENCY''),

                                oh.ordered_date,
                                null,
                                ''Corporate''
                                ) order_total
                                from
                                oe_order_headers_all oh
                                where
                                oh.header_id in
                                (select distinct header_id
                                from oe_ecc_global_temp)';

                                query_det_arr.EXTEND (1);
                                query_det_arr (1) :=
                                ecc_query_det_rec (l_ont_header_full_text,
                                G_ONT_ECC_UPS_OP);

                                query_det_arr.EXTEND (2);
                                query_det_arr (2) :=
                                ecc_query_det_rec (l_hdr_total_sql_text,
                                G_ONT_ECC_UPS_OP);

                                x_ecc_ds_meta_rec :=
                                ecc_ds_meta_rec (p_dataset_key, query_det_arr);
                                x_return_status := 'S';
                                ELSIF (p_load_type = 'INCREMENTAL_LOAD')
                                THEN
                                --incremental load of
                                ontheaders
                                l_ont_header_full_text :=
                                'SELECT * FROM (
                                SELECT order_number,
                                customer,customer_number,ship_to_location, order_date, status,

```

```

order_alert_count,
                                                    order_alert_flag,
ecc_spec_id,header_id,header_hold_name,customer_po,request_date,
order_type,
                                                    org_id,
order_currency,language_code,
                                                    (SELECT p.full_name
FROM per_all_people_f p, fnd_user u WHERE p.person_id = u.employee_id
AND
                                                    trunc(sysdate)
between p.effective_start_date and p.effective_end_date
AND u.user_id =
csr_user_id) csr_user_name
FROM
(
SELECT oegt.
order_number ,
oegt.customer,
customer_number,
oegt.
ship_to_location,
ship_su.LOCATION
oegt.order_date,
oegt.status,
SUM (oeht.
alert_count) + MAX(NVL(oeht.hdr_alert_count,0)) order_alert_count,
DECODE (SUM (oegt.
alert_count) + MAX(NVL(oeht.hdr_alert_count,0)),
0,'ecc_blank','ecc_warning') order_alert_flag,
oegt.header_id
ecc_spec_id,
oegt.header_id,
header_hold_name,
oegt.
customer_po,
oh.cust_po_number
oh.request_date,
oegt.order_type,
oegt.org_id,
oegt.func_currency
order_currency,
oh.csr_user_id,
oegt.language_code
FROM
oe_ecc_global_temp
oe_ecc_header_totals
oe_order_headers_all
hz_cust_site_uses_all ship_su
WHERE
oegt.line_id =oeht.
line_id AND
oegt.header_id =oh.
header_id AND
oeht.header_id =oh.
header_id AND
oh.ship_to_org_id =
ship_su.site_use_id(+) AND
oegt.header_id IN
(SELECT
header_id FROM oe_ecc_headers
UNION
SELECT DISTINCT

```

```

header_id FROM oe_ecc_deletion_map )

customer_number,

header_hold_name,

hdr_alert_count,

status,

order_type,

as order_currency

( '
        || v_for_lang_pivot_clause
        || '))';

l_hdr_total_sql_text :=
    'select oh.header_id ecc_spec_id,
            oe_ecc_util_pvt.
get_order_total(oh.header_id,
oh.transactional_curr_code,
oh.org_id,
fnd_profile.Value(''OM_ECC_DISPLAY_CURRENCY''),
oh.ordered_date,
null,
''Corporate''
) order_total
from
    oe_order_headers_all oh
where
    oh.header_id in (select
header_id from oe_ecc_headers
UNION
SELECT DISTINCT
header_id FROM oe_ecc_deletion_map)';

l_ont_header_hold_text :=
    'SELECT DISTINCT oegt.header_id ecc_spec_id, oegt.
header_hold_name FROM oe_ecc_global_temp oegt';

query_det_arr.EXTEND (1);
query_det_arr (1) :=
    ecc_query_det_rec (l_ont_header_full_text,
G_ONT_ECC_UPS_OP);
query_det_arr.EXTEND (2);
query_det_arr (2) :=

```

```

ecc_query_det_rec (l_hdr_total_sql_text,
                  G_ONT_ECC_UPS_OP);

      query_det_arr.EXTEND (3);
      query_det_arr (3) :=
        ecc_query_det_rec (l_hdr_del_sql_text,
G_ONT_ECC_DEL_OP);

      query_det_arr.EXTEND (4);
      query_det_arr (4) :=
        ecc_query_det_rec (l_ont_header_hold_text,
G_ONT_ECC_RPL_OP);
      x_ecc_ds_meta_rec :=
        ecc_ds_meta_rec (p_dataset_key, query_det_arr);

      UPDATE oe_ecc_deletion_map
        SET processed = 'Y';

      --trunc_temp_tables('END_LOAD');
      x_return_status := 'S';
    END IF;
-- load type =
full load
    END IF;
-- data set is ont-lines or
onheaders
  -- set_db_trace('disable');
  END GET_ECC_DATA_LOAD_INFO;

PROCEDURE OM_ECC_DATALOAD (ERRBUF          OUT NOCOPY VARCHAR2,
                          RETCODE          OUT NOCOPY VARCHAR2,
                          SYSTEMNAME       IN          VARCHAR2,
                          LOADTYPE         IN          VARCHAR2,
                          LANGUAGES        IN          VARCHAR2,
                          TRACEENABLED     IN          VARCHAR2,
                          LOGLEVEL         IN          VARCHAR2)
IS
  l_req_data          VARCHAR2 (1000);
  l_request_id         NUMBER;
  lc_phase             VARCHAR2 (50);
  lc_status            VARCHAR2 (50);
  lc_dev_phase         VARCHAR2 (50);
  lc_dev_status        VARCHAR2 (50);
  lc_message           VARCHAR2 (50);
  l_req_return_status  BOOLEAN;
BEGIN
  l_req_data := fnd_conc_global.request_data;

  IF l_req_data = 'END'
  THEN
    fnd_file.put_line (FND_FILE.OUTPUT,
                      'Executed the sub request: OM ECC Data
Load');
    fnd_file.put_line (FND_FILE.LOG,
                      'Executed the sub request: OM ECC Data
Load');
    RETURN;
  END IF;

  fnd_file.put_line (
    FND_FILE.LOG,
    'Submitting request for ont for ont-lines and loadType: '
    || loadType);

  l_request_id :=
    FND_REQUEST.SUBMIT_REQUEST (
      application => 'FND',

```

```

program          => 'ECCRUNDL',
description      => 'ECC - Run Data Load',
start_time       => SYSDATE,
ARGUMENT1        => systemName,
ARGUMENT2        => 'ont',
ARGUMENT3        => 'ont-lines',
ARGUMENT4        => loadType,
ARGUMENT5        => LANGUAGES,
ARGUMENT6        => TRACEENABLED,
ARGUMENT7        => LOGLEVEL,
sub_request      => FALSE);

COMMIT;

IF l_request_id > 0
THEN
    fnd_file.put_line (FND_FILE.LOG,
                        'Submitted request id: ' ||
l_request_id);
    fnd_file.put_line (FND_FILE.LOG,
                        'Waiting for submitted request
complete');

    l_req_return_status :=
        fnd_concurrent.wait_for_request (
            request_id => l_request_id,
            interval    => 3,
            max_wait    => 0,
            phase       => lc_phase,
            STATUS      => lc_STATUS,
            dev_phase   => lc_dev_phase,
            dev_status  => lc_dev_status,
            MESSAGE     => lc_message);

    fnd_file.put_line (FND_FILE.LOG,
                        'Waiting for submitted request
complete');
    fnd_file.put_line (FND_FILE.LOG,
                        ' lc_dev_phase:' || lc_dev_phase);
    fnd_file.put_line (FND_FILE.LOG,
                        ' lc_dev_status:' || lc_dev_status);
END IF;

IF UPPER (lc_dev_phase) = 'COMPLETE'
THEN
    IF    UPPER (lc_dev_status) = 'NORMAL'
    OR    UPPER (lc_dev_status) = 'WARNING'
    THEN
        fnd_file.put_line (
            FND_FILE.LOG,
            'Submitting request -> ECCRUNDL: ECC - Run Data Load
for ont-headers');
        BEGIN
            l_request_id :=
                FND_REQUEST.SUBMIT_REQUEST (
                    application => 'FND',
                    program     => 'ECCRUNDL',
                    description  => 'ECC - Run Data Load',
                    start_time   => SYSDATE,
                    ARGUMENT1    => systemName,
                    ARGUMENT2    => 'ont',
                    ARGUMENT3    => 'ont-headers',
                    ARGUMENT4    => loadType,
                    ARGUMENT5    => LANGUAGES,

```

```

ARGUMENT6      => TRACEENABLED,
                ARGUMENT7      => LOGLEVEL,
                sub_request     => FALSE);
COMMIT;
fnd_file.put_line (
    FND_FILE.OUTPUT,
    'ECC Load Job Request ID:' || l_request_id);
fnd_file.put_line (
    FND_FILE.LOG,
    'ECC Load Job Request ID:' || l_request_id);
EXCEPTION
    WHEN OTHERS
    THEN
        fnd_file.put_line (
            FND_FILE.LOG,
            'OTHERS exception while submitting 2: '
            || SQLERRM);
    END;
END IF;

IF l_request_id > 0
THEN
    fnd_file.put_line (FND_FILE.LOG,
        'Submitted request id: ' ||
l_request_id);
    fnd_file.put_line (
        FND_FILE.LOG,
        'Waitting for submitted request complete+');

    l_req_return_status :=
        fnd_concurrent.wait_for_request (
            request_id => l_request_id,
            interval   => 3,
            max_wait    => 0,
            phase       => lc_phase,
            STATUS      => lc_STATUS,
            dev_phase    => lc_dev_phase,
            dev_status   => lc_dev_status,
            MESSAGE      => lc_message);

    fnd_file.put_line (
        FND_FILE.LOG,
        'Waitting for submitted request complete-');
    fnd_file.put_line (FND_FILE.LOG,
        ' lc_dev_phase:' || lc_dev_phase);
    fnd_file.put_line (FND_FILE.LOG,
        ' lc_dev_status:' || lc_dev_status);
END IF;
END IF;

--fnd_conc_global.set_req_globals(conc_status => 'PAUSED',
request_data => 'END');
IF lc_dev_phase = 'COMPLETE'
THEN
    IF lc_dev_status = 'WARNING'
    THEN
        RETCODE := 1;
    ELSIF lc_dev_status = 'ERROR'
    THEN
        RETCODE := 2;
    ELSE
        RETCODE := 0;
    END IF;
END IF;

```

```

ERRBUF := 'Sub-Request submitted!';
EXCEPTION
  WHEN OTHERS
  THEN
    RETCODE := 1;
    fnd_file.put_line (
      FND_FILE.OUTPUT,
      'OTHERS exception while submitting : WSH ECC Data
Load'
      || SQLERRM);
    fnd_file.put_line (
      FND_FILE.LOG,
      'OTHERS exception while submitting : WSH ECC Data
Load'
      || SQLERRM);
END OM_ECC_DATALOAD;

PROCEDURE get_desc_metadata_load_info (
  p_dataset_key      IN      VARCHAR2,
  p_dataset_attrs    IN      VARCHAR2,
  ecc_sec_field_values DEFAULT
NULL,
  p_languages        IN      VARCHAR2,
  x_ecc_ds_meta_rec   OUT NOCOPY ecc_ds_meta_rec,
  x_return_status     OUT NOCOPY VARCHAR2)
IS
  --l_ecc_dff_segments varchar2(40) := fnd_profile.value('OM:
ECC DFF SEGMENTS');
  query_det_arr      ecc_query_det_arr_type
                    := ecc_query_det_arr_type (NULL);

  l_view_stmt        VARCHAR2 (4000);
  l_column_prefix    VARCHAR2 (2500);
  --l_dff_query_tbl    FND_ECC_DFF_UTIL.dff_query_tbl;
  --vrow              FND_ECC_DFF_UTIL.dff_query_rec;

BEGIN
  --insert_debug('Inside get_desc_metadata_load_info' ||
p_dataset_key || ' - ' || p_languages);
  /*
    vrow.flexfield_name      := 'OE_HEADER_ATTRIBUTES';
    vrow.app_id              := 660;
    vrow.column_prefix       := 'ORDER_';
    vrow.attribute_metadata  := FND_ECC_DFF_UTIL.
FND_ECC_DFF_ATTRIBUTES;
    vrow.precedence_metadata := FND_ECC_DFF_UTIL.
FND_ECC_DFF_PRECEDENCE;
    vrow.context_metadata   := FND_ECC_DFF_UTIL.
FND_ECC_DFF_CONTEXT;

    l_dff_query_tbl(l_dff_query_tbl.count) :=vrow;

    vrow.flexfield_name      := 'OE_LINE_ATTRIBUTES';
    vrow.app_id              := 660;
    vrow.column_prefix       := '';
    vrow.attribute_metadata  := FND_ECC_DFF_UTIL.
FND_ECC_DFF_ATTRIBUTES;
    vrow.precedence_metadata := FND_ECC_DFF_UTIL.
FND_ECC_DFF_PRECEDENCE;
    vrow.context_metadata   := FND_ECC_DFF_UTIL.
FND_ECC_DFF_CONTEXT;

    l_dff_query_tbl(l_dff_query_tbl.count) :=vrow;

    FND_ECC_DFF_UTIL.GET_ECC_DFF_INFO(p_dataset_key,p_languages,
l_dff_query_tbl, x_ecc_ds_meta_rec ,x_return_status);*/

```

```

        x_return_status := 'S';
    END get_desc_metadata_load_info;

    /* This function is copied from OE_LINE_STATUS_PUB.
    Get_Line_Status
    * Modified to add language as input parameter
    */
    FUNCTION Get_Line_Status (p_line_id           IN NUMBER,
                             p_flow_status_code  IN VARCHAR2,
                             p_language_code     IN VARCHAR2)
        RETURN VARCHAR2
    IS
        l_flow_status_code      VARCHAR2 (80);
        l_flow_meaning          VARCHAR2 (80);
        released_count          NUMBER;
        total_count             NUMBER;
        l_debug_level CONSTANT NUMBER := oe_debug_pub.g_debug_level;
        l_language_code         VARCHAR2 (10);
    BEGIN
        l_flow_status_code := p_flow_status_code;
        l_language_code := p_language_code;

        IF p_flow_status_code IS NULL
        THEN
            SELECT flow_status_code
            INTO l_flow_status_code
            FROM oe_order_lines
            WHERE line_id = p_line_id;
        END IF;

        IF      l_flow_status_code <> 'AWAITING_SHIPPING'
           AND l_flow_status_code <> 'PRODUCTION_COMPLETE'
           AND l_flow_status_code <> 'PICKED'
           AND l_flow_status_code <> 'PICKED_PARTIAL'
           AND l_flow_status_code <> 'PO_RECEIVED'
           AND l_flow_status_code <> 'SUPPLY_PARTIAL'
        THEN
            SELECT meaning
            INTO l_flow_meaning
            FROM fnd_lookup_values lv
            WHERE      lookup_type = 'LINE_FLOW_STATUS'
                   AND lookup_code = l_flow_status_code
                   AND LANGUAGE = l_language_code
                   AND VIEW_APPLICATION_ID = 660
                   AND SECURITY_GROUP_ID =
                        fnd_global.Lookup_Security_Group (
                            lv.lookup_type,
                            lv.view_application_id);
        ELSE
            SELECT SUM (DECODE (released_status, 'Y', 1, 'C', 1, 0)),
                   SUM (1)
            INTO released_count, total_count
            FROM wsh_delivery_details
            WHERE      source_line_id = p_line_id
                   AND source_code = 'OE'
                   AND released_status <> 'D';

            IF released_count = total_count
            THEN
                SELECT meaning
                INTO l_flow_meaning
                FROM fnd_lookup_values lv
                WHERE      lookup_type = 'LINE_FLOW_STATUS'
                       AND lookup_code = 'PICKED'
                       AND LANGUAGE = l_language_code
            END IF;
        END IF;
    END Get_Line_Status;

```

```

AND VIEW_APPLICATION_ID = 660
      AND SECURITY_GROUP_ID =
        fnd_global.Lookup_Security_Group (
          lv.lookup_type,
          lv.view_application_id);
ELSIF released_count < total_count AND released_count <> 0
THEN
  SELECT meaning
    INTO l_flow_meaning
  FROM fnd_lookup_values lv
 WHERE   lookup_type = 'LINE_FLOW_STATUS'
        AND lookup_code = 'PICKED_PARTIAL'
        AND LANGUAGE = l_language_code
        AND VIEW_APPLICATION_ID = 660
        AND SECURITY_GROUP_ID =
          fnd_global.Lookup_Security_Group (
            lv.lookup_type,
            lv.view_application_id);
ELSE
  SELECT meaning
    INTO l_flow_meaning
  FROM fnd_lookup_values lv
 WHERE   lookup_type = 'LINE_FLOW_STATUS'
        AND lookup_code = l_flow_status_code
        AND LANGUAGE = l_language_code
        AND VIEW_APPLICATION_ID = 660
        AND SECURITY_GROUP_ID =
          fnd_global.Lookup_Security_Group (
            lv.lookup_type,
            lv.view_application_id);

  END IF;
END IF;

IF l_debug_level > 0
THEN
  fnd_file.put_line (FND_FILE.LOG,
                    'Exiting Get_Line_Status:' ||
l_flow_meaning);
END IF;

RETURN l_flow_meaning;
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN
    NULL;
  WHEN TOO_MANY_ROWS
  THEN
    NULL;
  WHEN OTHERS
  THEN
    NULL;
END Get_Line_Status;

PROCEDURE set_db_trace (p_action VARCHAR2)
IS
BEGIN
  --insert_debug('Inside enable_db_trace procedure');

  IF p_action = 'enable'
  THEN
    --insert_debug('before enabling trace');
    EXECUTE IMMEDIATE 'ALTER SYSTEM SET max_dump_file_size =
unlimited';
    EXECUTE IMMEDIATE 'ALTER SYSTEM SET timed_statistics =
true';

```

```

EXECUTE IMMEDIATE 'ALTER session set tracefile_identifier=
''suneela_ecc''';

EXECUTE IMMEDIATE 'ALTER session set events ''10046 trace
name context forever, level 8''';

EXECUTE IMMEDIATE 'ALTER session set statistics_level =
''ALL''';
--insert_debug('after enabling trace');

ELSIF p_action = 'disable'
THEN
--insert_debug('before disabling trace');
DBMS_SESSION.reset_package;

EXECUTE IMMEDIATE 'ALTER SESSION SET EVENTS ''10046 trace
name context off''';
--insert_debug('after disabling trace');

END IF;
EXCEPTION
WHEN OTHERS
THEN
NULL;
--insert_debug(sqlerrm);
END set_db_trace;

PROCEDURE get_accessible_orgs (
p_user_id          IN          NUMBER,
p_resp_id          IN          NUMBER,
x_accessible_outab OUT NOCOPY icx_tbl_varchar240)
IS
l_ou_tab          mo_global.OrgIdTab;
l_tab             icx_tbl_varchar240;
l_count           NUMBER := 0;
l_application_id  NUMBER := NULL;
BEGIN
BEGIN
SELECT application_id
INTO l_application_id
FROM fnd_responsibility
WHERE responsibility_id = p_resp_id;
EXCEPTION
WHEN OTHERS
THEN
l_application_id := 660;
END;
END;

Fnd_Global.Apps_Initialize (p_user_id, p_resp_id,
l_application_id);

mo_global.init ('ONT');

l_tab := icx_tbl_varchar240 ();

SELECT TO_CHAR (organization_id)
BULK COLLECT INTO l_tab
FROM mo_glob_org_access_tmp;

x_accessible_outab := l_tab;
END get_accessible_orgs;

/*Function get_order_total is used to calculate the order total
during full load and incremental load. The order
total value is used in ont-headers data set. */

```

```

FUNCTION get_order_total (p_header_id          NUMBER,
                        p_header_currency_code VARCHAR2,
                        p_header_org_id        NUMBER,
                        p_func_currency_code   VARCHAR2,
                        p_order_date           DATE,
                        p_conversion_rate       NUMBER,
                        p_conversion_type_code  VARCHAR2)

RETURN NUMBER

IS
    l_total_transactional_amt  NUMBER;
    l_order_total              NUMBER;
    l_return_status            VARCHAR2 (100);
    l_func_currency_code       VARCHAR2 (100);
BEGIN
    -- first select the amount in ordered currency.
    --insert_debug('Inside get_order_total');
    BEGIN
        SELECT SUM (
            ROUND (
                NVL (l.ordered_quantity * l.
unit_selling_price, 0),
                2))
            INTO l_total_transactional_amt
            FROM oe_order_lines_all l
            WHERE l.header_id = p_header_id;
        --insert_debug('l_total_transactional_amt is
: '||l_total_transactional_amt);
    EXCEPTION
        WHEN OTHERS
        THEN
            l_total_transactional_amt := 0;
    END;

    -- convert the ordered amount into currency based on the value
of profile option "ECC display currency"
    --insert_debug('p_func_currency_code:'||p_func_currency_code);
    BEGIN
        IF p_func_currency_code IS NOT NULL
        THEN
            l_func_currency_code := p_func_currency_code;
        ELSE
            l_func_currency_code :=
                oe_upgrade_misc.get_sob_currency (p_header_org_id);
        --insert_debug
('l_func_currency_code:'||l_func_currency_code);
    END IF;

    --insert_debug
('l_func_currency_code:'||l_func_currency_code);
    IF NVL (l_func_currency_code, 'abc') <>
p_header_currency_code
    THEN
        OE_UPGRADE_MISC.CONVERT_CURRENCY
(l_total_transactional_amt,
p_header_currency_code,
                                l_func_currency_code,
                                p_order_date,
                                NULL,
                                'Corporate',
                                l_return_status,
                                l_order_total);
        --insert_debug('l_order_total is:'||l_order_total);
    ELSE
        -- currency is same. No need to
convert.
        l_order_total := l_total_transactional_amt;
    END IF;
END;

```

```

--insert_debug('in else block l_order_total is: '||l_order_total);
    END IF;
    EXCEPTION
    WHEN OTHERS
    THEN
        RETURN NULL;
    END;

    --insert_debug('before returning l_order_total
is: '||l_order_total);
    RETURN l_order_total;
    END get_order_total;

    /* Procedure trunc_temp_tables is used to truncate the tables
before full load and incremental load.
    OE_ECC_HEADER_TOTALS table needs to be truncated only before full
load. However, the tables OE_ECC_GLOBAL_TEMP
    and OE_ECC_HEADERS need to be truncated before every run, both
full load and incremental load of ont-lines
    data set only. temp tables are not truncated during ont-headers run.
Becuase ont-headers data set works on
    data populated into temp tables via loads on ont-lines data sets.*/

    PROCEDURE trunc_temp_tables (p_load_type IN VARCHAR2)
    IS
        l_api_success    BOOLEAN := FALSE;
        l_status          VARCHAR2 (128);
        l_industry        VARCHAR2 (128);
        l_schema          VARCHAR2 (128);
        l_stmt            VARCHAR2 (256);
    BEGIN
        l_api_success :=
            fnd_installation.get_app_info (
                application_short_name => 'ONT',
                status                  => l_status,
                industry                => l_industry,
                oracle_schema           => l_schema);

        IF l_api_success
        THEN
            NULL;
            --insert_debug('api is success');
        ELSE
            NULL;
            --insert_debug('api is failure');
        END IF;

        --l_stmt := 'truncate table test_debug';
        --EXECUTE IMMEDIATE l_stmt;
        --insert_debug('inside trunc_temp_tables');

        IF l_api_success
        THEN
            api_success
            l_stmt :=
                'TRUNCATE TABLE ' || l_schema || '.' ||
                'OE_ECC_GLOBAL_TEMP';

            --insert_debug('before truncating OE_ECC_GLOBAL_TEMP');
            EXECUTE IMMEDIATE l_stmt;

            l_stmt :=
                'TRUNCATE TABLE ' || l_schema || '.' ||
                'OE_ECC_HEADERS';

            --insert_debug('before truncating OE_ECC_HEADERS');

```

```

EXECUTE IMMEDIATE l_stmt;

IF p_load_type = 'FULL_LOAD'
THEN
    l_stmt :=
        'TRUNCATE TABLE '
        || l_schema
        || '.'
        || 'OE_ECC_HEADER_TOTALS';

    --insert_debug('before truncating
OE_ECC_HEADER_TOTALS');
    EXECUTE IMMEDIATE l_stmt;

    l_stmt :=
        'TRUNCATE TABLE '
        || l_schema
        || '.'
        || 'OE_ECC_DELETION_MAP';

    EXECUTE IMMEDIATE l_stmt;
END IF;
-- load type is full
load.
END IF;
--if
api_success

--insert_debug('before truncating OE_ECC_DELETION_MAP');
--insert_debug('l_stmt:'||l_stmt);
DELETE FROM oe_ecc_deletion_map
WHERE processed = 'Y';
--insert_debug('After truncating');

END trunc_temp_tables;

/* ECC V3 ER start */

FUNCTION get_csr_details (p_sold_to_org_id NUMBER,
                          p_inventory_item_id NUMBER)
RETURN NUMBER
IS
    l_csr_user      VARCHAR2 (200);
    l_csr_user_id   NUMBER := NULL;
BEGIN
    IF p_sold_to_org_id IS NOT NULL AND p_inventory_item_id IS
NOT NULL
    THEN
        --insert_debug('both ct and product are present.try to
derive csr');
        fnd_file.put_line (
            FND_FILE.LOG,
            'both ct and product are present.try to derive csr');

        BEGIN
            SELECT csr_user_id
            INTO l_csr_user_id
            FROM ( SELECT *
                    FROM oe_csr_assignment_rules
                    WHERE      sold_to_org_id = p_sold_to_org_id
                          AND ( item_category IN
                                (SELECT DISTINCT
                                  miv.
category_concat_segs
                                FROM
                                mtl_item_categories_v miv

```

```

WHERE miv.inventory_item_id =

p_inventory_item_id)
                                OR item_category IS NULL)
    ORDER BY sold_to_org_id, item_category) a
    WHERE ROWNUM = 1;

    RETURN l_csr_user_id;
EXCEPTION
    WHEN NO_DATA_FOUND
    THEN
        --insert_debug('CSR rule does not exist. Return
null');

        fnd_file.put_line (
            FND_FILE.LOG,
            'CSR rule does not exist. Return null');
        RETURN NULL;
    END;
ELSIF p_inventory_item_id IS NULL
THEN
    --insert_debug('Item details are not present.Return null');
    fnd_file.put_line (FND_FILE.LOG,
        'Item details are not present.Return
null');
    RETURN NULL;
ELSIF p_sold_to_org_id IS NULL
THEN
    --insert_debug('customer is not yet entered on the order.
Cannot assign CSR');
    fnd_file.put_line (
        FND_FILE.LOG,
        'customer is not yet entered on the order. Cannot
assign CSR');
    RETURN NULL;
END IF;
EXCEPTION
    WHEN OTHERS
    THEN
        --insert_debug('others exception:||sqlerrm);
        fnd_file.put_line (FND_FILE.LOG, 'others exception:' ||
SQLERRM);
        RETURN NULL;
END;

```

```

/* Procedure reassign_csr is called when Assign Customer Service
Rep CP is invoked.

```

```

**
**
** reassign_csr - Reassigns the customer service rep based on
assignment rules for the given inputs.
**
** IN parameters
**     CUSTOMER - sold to org id of the customer
**     CUSTSERVICE - user_id of the customer service rep
**     OVERRIDE - boolean value for override existing csr
** OUT parameters
**     ERRBUF - error
**     RETCODE - return status
**
** 1) prepare sql statement based on given parameters
** 2) open cursor with prepared sql statement
** 3) If override is yes, update the existing csr to null
** 4) Iterate over the cursor and update the csr
**
*/

```

```

PROCEDURE assign_csr (ERRBUF          OUT NOCOPY VARCHAR2,
                     RETCODE         OUT NOCOPY VARCHAR2,
                     CUSTOMER        NUMBER,
                     CUSTSERVICEREP  NUMBER,
                     OVERRIDE        VARCHAR2)
IS
    l_sold_to_org_id      NUMBER;
    l_customer_service_rep NUMBER;
    l_min_line_id         NUMBER;
    l_inventory_item_id   NUMBER;
    l_csr_user_id         NUMBER;
    order_header_id       NUMBER;
    l_override            VARCHAR2 (10);
    orders_to_process     SYS_REFCURSOR;
    l_error               BOOLEAN := FALSE;
    l_headers_all         header_csr_tbl;
    l_ECC_BULK_LOAD_SIZE  NUMBER
        := NVL (fnd_profile.Value_wnps ('OM_ECC_BULK_LOAD_SIZE'),
1000);
    l_query_string VARCHAR2 (1000)
        := 'select order_number,header_id,sold_to_org_id from
oe_order_headers where NVL(transaction_phase_code, ''F'') = ''F'' and
open_flag = ''Y''';
    BEGIN
        l_sold_to_org_id := CUSTOMER;
        l_customer_service_rep := CUSTSERVICEREP;
        l_override := OVERRIDE;

        mo_global.init ('ONT');
        mo_global.set_policy_context ('M', NULL);
        --insert_debug('l_sold_to_org_id :'||l_sold_to_org_id ||'-
l_customer_service_rep :'||l_customer_service_rep||' - l_override
:'||l_override);
        fnd_file.put_line (
            FND_FILE.LOG,
            'l_sold_to_org_id :'
            || l_sold_to_org_id
            || '- l_customer_service_rep :'
            || l_customer_service_rep
            || ' - l_override :'
            || l_override);

        IF l_customer_service_rep IS NOT NULL
        THEN
            IF l_override = 'N'
            THEN
                l_query_string := '';
                fnd_file.put_line (
                    FND_FILE.OUTPUT,
                    fnd_message.get_string ('ONT',
'ONT_ECC_OVERRIDE_NO'));
                fnd_file.put_line (
                    FND_FILE.LOG,
                    fnd_message.get_string ('ONT',
'ONT_ECC_OVERRIDE_NO'));
                l_error := TRUE;
                RETCODE := FND_API.G_RET_STS_ERROR;
                l_error := TRUE;
            ELSE
                l_query_string :=
                    l_query_string
                    || ' and csr_user_id = '
                    || l_customer_service_rep;
                -- csr
+override

```

```

IF l_sold_to_org_id IS NOT NULL
    THEN
        l_query_string :=
            l_query_string
            || ' and sold_to_org_id ='
            || l_sold_to_org_id;
csr+override+cust
    END IF;
    END IF;
    ELSIF l_sold_to_org_id IS NOT NULL
    THEN
        l_query_string :=
            l_query_string
            || ' and sold_to_org_id ='
            || l_sold_to_org_id
            || ',';
cust+override

        IF l_override = 'N'
        THEN
            l_query_string :=
                l_query_string || ' and csr_user_id is null'; --
only cust
        END IF;
        ELSIF l_override = 'N'
        THEN
            l_query_string := l_query_string || ' and csr_user_id is
null'; -- none
        END IF;

        --insert_debug('l_query_string : ' || l_query_string);
        fnd_file.put_line (FND_FILE.LOG,
            'l_query_string : ' || l_query_string);

        IF l_error <> TRUE
        THEN
            OPEN orders_to_process FOR l_query_string;

            LOOP
                FETCH orders_to_process
                BULK COLLECT INTO l_headers_all
                LIMIT l_ECC_BULK_LOAD_SIZE;

                EXIT WHEN l_headers_all.COUNT = 0;

                --insert_debug('l_headers_all.COUNT is: ' || l_headers_all.
COUNT);
                fnd_file.put_line (
                    FND_FILE.LOG,
                    'l_headers_all.COUNT is: ' || l_headers_all.COUNT);

                /* Audit issue - should not update last_update_date for
all the records.
                IF l_override = 'Y' THEN
                    FORALL indx IN 1 .. l_headers_all.COUNT
                        update oe_order_headers_all
                        set csr_user_id = null,
LAST_UPDATED_BY = FND_GLOBAL.USER_ID,
                        LAST_UPDATE_DATE = sysdate,
                        LAST_UPDATE_LOGIN = FND_GLOBAL.USER_ID
                        where header_id = l_headers_all(indx).header_id;
                END IF;
                */

                FOR i IN l_headers_all.FIRST .. l_headers_all.LAST
                LOOP

```

```

l_inventory_item_id := 0;
l_sold_to_org_id := l_headers_all (i).
sold_to_org_id;

--insert_debug('order_header_id :'||l_headers_all
(i).header_id);
BEGIN
    SELECT l.inventory_item_id
    INTO l_inventory_item_id
    FROM oe_order_lines_all l,
    (SELECT MIN (line_id) min_line_id
    FROM oe_order_lines_all
    WHERE header_id =
        l_headers_all (i).header_id)
l_min

    WHERE l.line_id = l_min.min_line_id;
EXCEPTION
    WHEN OTHERS
    THEN
        l_inventory_item_id := -1;
        --insert_debug('sql error for order : '||
l_headers_all(i).order_number);
        fnd_file.put_line (
            FND_FILE.LOG,
            'sql error for order : '
            || l_headers_all (i).order_number);
    END;

--insert_debug('inventory_item_id
:'||l_inventory_item_id ||'l_sold_to_org_id:'||l_sold_to_org_id);
fnd_file.put_line (
    FND_FILE.LOG,
    'inventory_item_id : '
    || l_inventory_item_id
    || ' - l_sold_to_org_id:'
    || l_sold_to_org_id
    || ' - order_header_id : '
    || l_headers_all (i).header_id);

IF l_inventory_item_id > 0
THEN
    l_csr_user_id :=
        oe_ecc_util_pvt.get_csr_details (
            l_sold_to_org_id,
            l_inventory_item_id);

    --insert_debug('updating l_csr_user_id
:'||l_csr_user_id ||' for order number :'||l_headers_all(i).
order_number);

    -- should not update if l_csr_user_id same as
on the order

    BEGIN
        UPDATE oe_order_headers_all
        SET csr_user_id = l_csr_user_id,
            LAST_UPDATED_BY = FND_GLOBAL.USER_ID,
            LAST_UPDATE_DATE = SYSDATE,
            LAST_UPDATE_LOGIN = FND_GLOBAL.
USER_ID

        WHERE          header_id =
                        l_headers_all (i).header_id
                        AND NVL (csr_user_id, 0) <>
                        NVL (l_csr_user_id, 0);
    EXCEPTION
        WHEN APP_EXCEPTIONS.RECORD_LOCK_EXCEPTION
        THEN

```

```

--insert_debug('others exception:'||sqlerrm);
                                fnd_file.put_line (
                                    FND_FILE.LOG,
                                    'RECORD_LOCK_EXCEPTION
exception:'

                                || SQLERRM);
                                fnd_file.put_line (
                                    FND_FILE.LOG,
                                    'RECORD_LOCK_EXCEPTION exception:

error occured for '

                                || l_headers_all (i).header_id
                                || '---'
                                || l_headers_all (i).order_number);
                                WHEN OTHERS
                                THEN
                                    --insert_debug('others
exception:'||sqlerrm);
                                    fnd_file.put_line (
                                        FND_FILE.LOG,
                                        'others exception:' || SQLERRM);
                                    fnd_file.put_line (
                                        FND_FILE.LOG,
                                        'others exception: error occured
for '

                                || l_headers_all (i).header_id
                                || '---'
                                || l_headers_all (i).order_number);
                                END;
                                ELSIF l_override = 'Y'
                                THEN
                                    -- update csr to null only if all lines are
deleted and csr it is not null
                                    BEGIN
                                        UPDATE oe_order_headers_all
                                        SET csr_user_id = NULL,
                                        LAST_UPDATED_BY = FND_GLOBAL.USER_ID,
                                        LAST_UPDATE_DATE = SYSDATE,
                                        LAST_UPDATE_LOGIN = FND_GLOBAL.
USER_ID

                                        WHERE          header_id =
                                                    l_headers_all (i).header_id
                                        AND csr_user_id IS NOT NULL;
                                EXCEPTION
                                    WHEN APP_EXCEPTIONS.RECORD_LOCK_EXCEPTION
                                    THEN
                                        --insert_debug('others
exception:'||sqlerrm);
                                        fnd_file.put_line (
                                            FND_FILE.LOG,
                                            'RECORD_LOCK_EXCEPTION
exception:'

                                        || SQLERRM);
                                        fnd_file.put_line (
                                            FND_FILE.LOG,
                                            'RECORD_LOCK_EXCEPTION exception:

error occured for '

                                        || l_headers_all (i).header_id
                                        || '---'
                                        || l_headers_all (i).order_number);
                                        WHEN OTHERS
                                        THEN
                                            --insert_debug('others
exception:'||sqlerrm);
                                            fnd_file.put_line (
                                                FND_FILE.LOG,
                                                'others exception:' || SQLERRM);

```

```

fnd_file.put_line (
                                FND_FILE.LOG,
                                'others exception: error occurred
for '
                                || l_headers_all (i).header_id
                                || '---'
                                || l_headers_all (i).order_number);
                                END;
                                END IF;
                                END LOOP;

                                COMMIT;
                                END LOOP;

                                CLOSE orders_to_process;

                                RETCODE := FND_API.G_RET_STS_SUCCESS;
                                -- COMMIT; -- should we commit after every update in inner
loop.
                                fnd_file.put_line (FND_FILE.OUTPUT,
                                'Program completed successfully');
                                fnd_file.put_line (FND_FILE.LOG,
                                'Program completed successfully');
                                END IF;
                                EXCEPTION
                                WHEN OTHERS
                                THEN
                                fnd_file.put_line (
                                FND_FILE.OUTPUT,
                                'OTHERS exception while submitting : ' || SQLERRM);
                                fnd_file.put_line (
                                FND_FILE.LOG,
                                'OTHERS exception while submitting : ' || SQLERRM);
                                RETCODE := FND_API.G_RET_STS_ERROR;
                                END assign_csr;
/* ECC V3 ER end */

END OE_ECC_UTIL_PVT;
/

```

データ・ロードおよびメタデータ・ロード のシグネチャ

データ・ロードおよびメタデータ・ロードのシグネチャ

```
CREATE OR REPLACE PACKAGE APPS.icx_ecc_util_pvt AUTHID CURRENT_USER AS
/* $Header: ICXECCUS.pls 120.0.1 2018/11/29 11:10:32 jdoe noship $ */
    g_icx_ecc_ins_op CONSTANT VARCHAR2(30) := 'INSERT';
    g_icx_ecc_upd_op CONSTANT VARCHAR2(30) := 'UPDATE';
    g_icx_ecc_del_op CONSTANT VARCHAR2(30) := 'DELETE';
    g_icx_ecc_ups_op CONSTANT VARCHAR2(30) := 'UPSERT';
    g_icx_exc_op CONSTANT VARCHAR2(30) := 'EXECUTE';
    g_icx_dsc_op CONSTANT VARCHAR2(30) := 'DESCRIBE';
    g_icx_dsc_att_op CONSTANT VARCHAR2(30) := 'DESCRIBE|ATTRIBUTE';
    g_icx_dsc_pre_op CONSTANT VARCHAR2(30) :=
'DESCRIBE|PRECEDENCE_RULE';
    g_icx_exc_op CONSTANT VARCHAR2(30) := 'EXECUTE';
    g_icx_exc_att_op CONSTANT VARCHAR2(30) := 'EXECUTE|ATTRIBUTE';
    g_icx_exc_pre_op CONSTANT VARCHAR2(30) := 'EXECUTE|PRECEDENCE_RULE';
/*
    ** GET_ECC_DATA_LOAD_INFO - Retrieves the data load details for the
    given data set key
    **
    ** IN parameters
    **   p_dataset_key - data set key
    **   p_load_type - Indicating Full or incremental load
    (F/I/Custom).For custom there are no data load rules defined
    **   p_ds_last_success_run - Returns the last successful etl run for
    the data set key
    ** OUT parameters
    **   x_ecc_ds_meta_rec - ecc_ds_meta_rec
    **   x_return_status - return status
    */
    -----
    PROCEDURE get_ecc_data_load_info (
        p_dataset_key          IN VARCHAR2,
        p_load_type            IN VARCHAR2,
        p_ds_last_success_run  IN TIMESTAMP,
        p_languages            IN VARCHAR2,
        p_addl_params          IN ecc_sec_field_values,
        x_ecc_ds_meta_rec      OUT NOCOPY ecc_ds_meta_rec,
        x_return_status        OUT NOCOPY VARCHAR2
    );
    -----
/*
    ** GET_SQL_TEXT - Retrieves the sql text for a data set
    ** IN parameters
    **   p_dataset_key - data set key
    **   p_load_type - Indicating Full or incremental load
    (F/I/Custom).For custom there are no data load rules defined
    **   p_operation - Indicates the operations INSERT,UPDATE,DELETE and
    UPSERT
    **   p_data_type - Indicates the type of data
    **   p_ds_last_success_run - Returns the last successful etl run for
    the data set key
    ** RETURNS sql text
    */
    -----
    FUNCTION get_sql_text (
        p_dataset_key          IN VARCHAR2,
        p_load_type            IN VARCHAR2,
        p_operation            IN VARCHAR2,
        p_data_type            IN VARCHAR2,
        p_ds_last_success_run  IN TIMESTAMP,
        p_languages            IN VARCHAR2
    ) RETURN CLOB;
```

```

PROCEDURE create_ecc_desc_view (
    p_category_id IN NUMBER
);

PROCEDURE create_ecc_desc_views;

PROCEDURE drop_ecc_desc_views;

PROCEDURE get_desc_metadata_load_info (
    p_dataset_key      IN VARCHAR2,
    p_dataset_attrs    IN ecc_sec_field_values DEFAULT NULL,
    p_languages         IN VARCHAR2,
    x_ecc_ds_meta_rec   OUT NOCOPY ecc_ds_meta_rec,
    x_return_status     OUT NOCOPY VARCHAR2
);

FUNCTION makencname (
    p_attribute_name IN VARCHAR2
) RETURN VARCHAR2;

PROCEDURE submit_ecc_data_load (
    errbuf      OUT NOCOPY VARCHAR2,
    retcode     OUT NOCOPY VARCHAR2,
    p_system_name IN VARCHAR2,
    p_load_type  IN VARCHAR2,
    p_languages  IN VARCHAR2,
    p_log_level  IN VARCHAR2,
    p_trace_enabled IN VARCHAR2
);

END icx_ecc_util_pvt;
/

```

サンプル・コンカレント・プログラム

サンプル・コンカレント・プログラム

```
PROCEDURE SUBMIT_ECC_DATA_LOAD(
    ERRBUF          OUT NOCOPY VARCHAR2 ,
    RETCODE         OUT NOCOPY VARCHAR2 ,
    p_system_name   IN VARCHAR2,
    p_load_type     IN VARCHAR2,
    p_languages     IN VARCHAR2,
    p_log_level     IN VARCHAR2,
    p_trace_enabled IN VARCHAR2
) IS
    l_app_short_name varchar2(10) := 'icx';
    l_request_id number;
    l_req_data varchar2(10);
    l_req_data1 varchar2(10);
    l_load_type varchar2(200);
BEGIN

    l_req_data := fnd_conc_global.request_data;
    SELECT decode(p_load_type, 'FULL_LOAD', 'METADATA_LOAD',
p_load_type) INTO l_load_type FROM dual;

    IF l_req_data = 'END' THEN
        fnd_file.put_line(FND_FILE.OUTPUT, 'Executed the sub request: ICX
ECC Data Load for load type : ' || p_load_type);
        fnd_file.put_line(FND_FILE.LOG, 'Executed the sub request: ICX ECC
Data Load for load type : ' || p_load_type);
        RETURN;
    END IF;

    IF (l_req_data = 'END-META' and p_load_type = 'FULL_LOAD') THEN
        fnd_file.put_line(FND_FILE.OUTPUT, 'Executed the sub request:
ICX ECC Data Load for load type : ' || l_load_type);
        fnd_file.put_line(FND_FILE.LOG, 'Executed the sub request: ICX
ECC Data Load for load type : ' || l_load_type);
        l_request_id := fnd_request.submit_request(
            application => 'FND',
            program      => 'ECCRUNDL',
            description  => 'ECC - Run Data Load',
            start_time   => sysdate,
            argument1    => p_system_name,
            argument2    => l_app_short_name,
            argument3    => null,
            argument4    => p_load_type,
            argument5    => p_languages,
            argument6    => p_trace_enabled,
            argument7    => p_log_level,
            sub_request  => true);

        fnd_file.put_line(FND_FILE.OUTPUT, 'ECC Load for load type : '
|| p_load_type || ' Job Request ID: ' || l_request_id);
        fnd_file.put_line(FND_FILE.LOG, 'ECC Load for load type : ' ||
p_load_type || ' Job Request ID: ' || l_request_id);

        IF l_request_id = 0 THEN
            --
            -- If request submission failed, exit with error.
            --
            ERRBUF := fnd_message.get;
            RETCODE := 2;
        ELSE
            --
            -- Here we set the globals to put the program into the
            -- PAUSED status on exit, and to save the state in
```

```

-- request_data.
--
        fnd_conc_global.set_req_globals(conc_status => 'PAUSED',
request_data => 'END');

        ERRBUF := 'Sub-Request submitted!';
        RETCODE := 0 ;

        END IF;
        RETURN;
    END IF;
    -- Submit l_load_type which can be either METADATA_LOAD or
    INCREMENTAL_LOAD.
    l_request_id := fnd_request.submit_request(
        application => 'FND',
        program      => 'ECCRUNDL',
        description  => 'ECC - Run Data Load',
        start_time   => sysdate,
        argument1    => p_system_name,
        argument2    => l_app_short_name,
        argument3    => null,
        argument4    => l_load_type,
        argument5    => p_languages,
        argument6    => p_trace_enabled,
        argument7    => p_log_level,
        sub_request  => true);

    fnd_file.put_line(FND_FILE.OUTPUT,'ECC Load for load type : ' ||
l_load_type || ' Job Request ID:' || l_request_id);
    fnd_file.put_line(FND_FILE.LOG,'ECC Load for load type : ' ||
l_load_type || ' Job Request ID:' || l_request_id);

    IF l_request_id = 0 THEN
        --
        -- If request submission failed, exit with error.
        --
        ERRBUF := fnd_message.get;
        RETCODE := 2;
    ELSE
        --
        -- Here we set the globals to put the program into the
        -- PAUSED status on exit, and to save the state in
        -- request_data.
        --
        IF(l_load_type = 'METADATA_LOAD' AND p_load_type='FULL_LOAD')
THEN
            l_req_data1 := 'END-META';
        ELSE
            l_req_data1 := 'END';
        END IF;

        fnd_conc_global.set_req_globals(conc_status => 'PAUSED',request_data
=> l_req_data1);

        ERRBUF := 'Sub-Request submitted!';
        RETCODE := 0 ;

        END IF;
        RETURN;
    EXCEPTION
    WHEN OTHERS THEN
        RETCODE := 1;
        fnd_file.put_line(FND_FILE.OUTPUT,'OTHERS exception while submitting
: ICX ECC Data Load' || sqlerrm);
        fnd_file.put_line(FND_FILE.LOG,'OTHERS exception while submitting :
ICX ECC Data Load' || sqlerrm);

```

```
END SUBMIT_ECC_DATA_LOAD;
```

データ・セット ECC_SPEC_ID の構造

概要

この項では、データ・セットの詳細と ECC_SPEC_ID の構造について製品別に説明します。

Enterprise Asset Management

データ・セット・キー: eam-wo

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs,  
wip_operations,  
wip_operation_resources,  
wip_op_resource_instances
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id,  
wo.operation_seq_num,  
wor.resource_seq_num,  
wor.resource_id,  
wori.instance_id,  
wori.serial_number
```

ECC_SPEC_ID 連結構造

```
( wdj.wip_entity_id || '-' || wo.operation_seq_num || '-' || wor.  
resource_seq_num || '-' || wor.resource_id || '-INSTANCE-' || wori.  
instance_id || '-EQUIPMENT-' || wori.serial_number
```

データ・セット・キー: eam-wr

Oracle E-Business Suite 表の名前

```
wip_eam_work_requests
```

ECC_SPEC_ID 列

work_request_number

ECC_SPEC_ID 連結構造

なし。

データ・セット・キー: eam-mat

Oracle E-Business Suite 表の名前

mtl_system_items_kfv,
mtl_cross_references_b,
mtl_mfg_part_numbers,
mtl_secondary_inventories

ECC_SPEC_ID 列

msi.inventory_item_id,
msi.organization_id,
mcr.cross_reference_id,
mpn.manufacturer_id,
mpn.mfg_part_num,
msinv.secondary_inventory_name,
msi.concatenated_segments (asset_bom)

ECC_SPEC_ID 連結構造

(msi.inventory_item_id || '-' || msi.organization_id || '-CROSS_REF-'
|| mcr.cross_reference_id || '-MANUFACTURER-' || mpn.manufacturer_id || '-
MFG_PART_NUM-' || mpn.mfg_part_num || '-SUB_INV-' || msinv.
secondary_inventory_name || '-ASSET_BOM-' || NVL(asset_bom, 'NO_VAL'))

データ・セット・キー: eam-asset

Oracle E-Business Suite 表の名前

csi_item_instances,
wip_discrete_jobs,
wip_eam_period_balances

ECC_SPEC_ID 列

cii.instance_id,
wdj.wip_entity_id,
period_name

ECC_SPEC_ID 連結構造

((wdj.wip_entity_id || '-' || workorder_cost.period_name) or To_char
(cii.instance_id)

Projects

データ・セット・キー: pa-ds-cost-prj-task

Oracle E-Business Suite 表の名前

```
pa_exp_item_mls_v
pa_cost_dist_lines_v
pa_expenditure_items_all
pa_project_asset_line_details
pa_project_asset_lines_all
pa_project_types_all
pa_project_statuses pa_tasks
pa_budget_versions
pa_budget_lines
pa_resource_assignments
pa_budget_types pbt
pa_expenditures_all
pa_expenditure_groups_all
pa_transaction_interface_all
hr_all_organization_units_tl
```

ECC_SPEC_ID 列

```
ppa.org_id, ppa.project_id, ppa.task_id
```

ECC_SPEC_ID 連結構造

```
'PROJ_TASK'|| TO_CHAR(ppa.org_id)|| '-'|| TO_CHAR(ppa.project_id)||
 '-'|| TO_CHAR(ppa.task_id)
```

データ・セット・キー: pa-ds-cost-eicdl

Oracle E-Business Suite 表の名前

```
pa_exp_item_mls_v
pa_cost_distribution_lines_all
pa_expenditure_groups_all
pa_system_linkages
gl_code_combinations_kfv
hr_all_organization_units
pa_cc_tp_schedules xla_events
xla_ae_headers
xla_accounting_errors
```

ECC_SPEC_ID 列

```
ei.expenditure_item_id cdl.line_num cdl.line_type
```

ECC_SPEC_ID 連結構造

```
'PROJ-'|| TO_CHAR(ei.expenditure_item_id)|| '-'|| TO_CHAR(cdl.
line_num)|| '-'|| cdl.line_type
```

データ・セット・キー: pa-ds-cost-asset

Oracle E-Business Suite 表の名前

```
pa_implementations_all
gl_sets_of_books
pa_project_types_all
pa_projects_all
pa_project_assets_all
pa_project_asset_assignments
pa_project_asset_lines_all
pa_project_asset_line_details
hr_all_organization_units_tl
```

ECC_SPEC_ID 列

```
ppal.project_asset_id
pp.project_id
ppal.project_asset_line_id
ppal.task_id
ppal.project_asset_line_detail_id
ppald.proj_asset_line_dtl_uniq_id
```

ECC_SPEC_ID 連結構造

```
'PROJ-' || TO_CHAR(NVL(ppal.project_asset_id, pp.project_id)) || '-'
|| TO_CHAR(NVL(ppal.project_asset_line_id, pp.project_id)) || '-' ||
TO_CHAR(NVL(ppal.task_id, 99999)) || '-' || TO_CHAR(NVL(ppal.
project_asset_line_detail_id, pp.project_id)) || '-' || TO_CHAR(NVL
(ppald.proj_asset_line_dtl_uniq_id, pp.project_id))
```

データ・セット・キー: pa-ds-cost-source-imp

Oracle E-Business Suite 表の名前

```
pa_transaction_interface_all
pa_transaction_sources
pa_project_types_all
pa_projects_all
pa_tasks
hr_all_organization_units_tl
gl_code_combinations_kfv
hr_locations
po_vendors
po_headers_all
po_lines_all
po_distributions_all
rcv_shipment_headers
rcv_transactions
rcv_shipment_lines
rcv_receiving_sub_ledger
rcv_accounting_event_types
rcv_accounting_events
ap_invoices_all
ap_invoice_lines_all
ap_invoice_distributions_all
ap_payment_hist_dists
ap_invoice_payments_all
ap_prepay_app_dists
```

ECC_SPEC_ID 列

```
ps.transaction_id  
ps.distribution_id  
ts.transaction_source
```

ECC_SPEC_ID 連結構造

```
'PROJ-' || TO_CHAR(ps.transaction_id) || '-' || TO_CHAR(ps.  
distribution_id) || '-' || TO_CHAR(ts.transaction_source),
```

データ・セット・キー: pa-ds-cost-bcbalances

Oracle E-Business Suite 表の名前

```
hr_all_organization_units_tl  
per_all_people_f  
pa_bc_packets  
pa_bc_balances  
pa_periods_all  
pa_budget_entry_methods  
pa_budget_versions  
gl_period_statuses  
pa_budgetary_control_options  
pa_budget_types  
pa_tasks  
pa_project_types_all  
pa_project_statuses  
pa_projects_all
```

ECC_SPEC_ID 列

```
bgt.project_id  
bgt.encumb_type  
bgt.task_id bgt.top_task_id  
bgt.parent_member_id  
bgt.resource_list_member_id  
bgt.start_date  
bgt.end_date
```

ECC_SPEC_ID 連結構造

```
'PBC-' || TO_CHAR(bgt.project_id) || '-' || TO_CHAR(bgt.encumb_type)  
|| '-' || TO_CHAR(bgt.task_id) || '-' || TO_CHAR(bgt.top_task_id) ||  
'-' || TO_CHAR(bgt.parent_member_id) || '-' || TO_CHAR(bgt.  
resource_list_member_id) || '-' || TO_CHAR(bgt.start_date, 'DD-MON-  
YYYY') || '-' || TO_CHAR(bgt.end_date, 'DD-MON-YYYY')
```

データ・セット・キー: pa-ds-cost-bcreresults

Oracle E-Business Suite 表の名前

```
pa_expenditure_types
po_headers_all
po_lines_all
po_distributions_all
pa_bc_commitments_all
ap_invoices_all
ap_invoice_lines_all
ap_invoice_distributions_all
po_vendors
gl_ledgers
pa_implementations_all
pa_projects_all
pa_tasks
po_req_distributions_all
po_requisition_lines_all
po_requisition_headers_all
pa_bc_commitments_all
pa_budget_versions
pa_budgetary_control_options
pa_expenditures_all
pa_expenditure_items_all
pa_cost_distribution_lines_all
pa_bc_packets
pa_budget_types
pa_project_statuses
pa_periods_all
hr_all_organization_units_tl
```

ECC_SPEC_ID 列

```
det.record_identifier
det.project_id
det.task_id
det.bc_commitment_ei_id
det.burden_cost_flag
det.document_type
det.document_header_id
det.document_distribution_id
```

ECC_SPEC_ID 連結構造

```
'PBC-' || det.record_identifier || TO_CHAR(det.project_id) || '-' ||
TO_CHAR(det.task_id) || '-' || TO_CHAR(det.bc_commitment_ei_id) || '-'
|| TO_CHAR(det.burden_cost_flag) || '-' || TO_CHAR(det.document_type) ||
 '-' || TO_CHAR(det.document_header_id) || '-' || TO_CHAR(det.
document_distribution_id)
```

Payables

データ・セット・キー: ap-trx

Oracle E-Business Suite 表の名前

```
ap_payment_schedules_all
```

ECC_SPEC_ID 列

invoice_id
payment_num

ECC_SPEC_ID 連結構造

invoice_id || '_' || payment_num

データ・セット・キー: ap-hold

Oracle E-Business Suite 表の名前

ap_holds_all

ECC_SPEC_ID 列

invoice_id
hold_id

ECC_SPEC_ID 連結構造

invoice_id || '_' || hold_id

データ・セット・キー: ap-paid

Oracle E-Business Suite 表の名前

ap_payment_schedules_all

ECC_SPEC_ID 列

invoice_payment_id

ECC_SPEC_ID 連結構造

なし。

Receivables

データ・セット・キー: ar-trx

Oracle E-Business Suite 表の名前

ar_payment_schedules_all

ECC_SPEC_ID 列

payment_schedule_id

ECC_SPEC_ID 連結構造

'TRX-' || payment_schedule_id

データ・セット・キー: ar-billproc

Oracle E-Business Suite 表の名前

```
ar_adjustments_all adjust
ra_customer_trx_all trx
ar_transmissions_all transms
```

ECC_SPEC_ID 列

```
adjust.adjustment_id
trx.customer_trx_id
transms.transmission_id
```

ECC_SPEC_ID 連結構造

```
'ARADJUST-' || adjust.adjustment_id
'ARIMCPTRX-' || trx.customer_trx_id
'ARLOCKBOX-' || transms.transmission_id
```

データ・セット・キー: ar-pmtproc

Oracle E-Business Suite 表の名前

```
ar_batches_all batches
ar_cash_receipts_al receipt
ar_batches_all batches
```

ECC_SPEC_ID 列

```
batches.batch_id
receipt.cash_receipt_id
batches.batch_id
```

ECC_SPEC_ID 連結構造

```
'ARRCPTBATCH-' || batches.batch_id
'ARCASHRCPT-' || receipt.cash_receipt_id
'ARRMITBATCH-' || batches.batch_id
```

データ・セット・キー: ar-history

Oracle E-Business Suite 表の名前

```
ar_cash_receipts_al receipt
ar_payment_schedules_all ps
```

ECC_SPEC_ID 列

```
receipt.cash_receipt_id
ps.payment_schedule_id
```

ECC_SPEC_ID 連結構造

```
'ARRCIPTHSTRY-' || receipt.cash_receipt_id
'ARTRXHSTRY-' || ps.payment_schedule_id
```

データ・セット・キー: ar-dispute

Oracle E-Business Suite 表の名前

ra_cm_requests_all cmreq

ECC_SPEC_ID 列

cmreq.request_id

ECC_SPEC_ID 連結構造

'CMREQ' || cmreq.request_id

Assets

データ・セット・キー: fa-asset

Oracle E-Business Suite 表の名前

fa_additions_b ad
fa_books bk
fa_asset_invoices ai
fa_distribution_history dh
gl_code_combinations
expense_cc

ECC_SPEC_ID 列

ad.asset_id
bk.transaction_header_id_in
NVL (ai.source_line_id, 0)
dh.distribution_id
expense_cc.code_combination_id

ECC_SPEC_ID 連結構造

ad.asset_id || '-' || bk.transaction_header_id_in || '-' || NVL (ai.
source_line_id, 0) || '-' || dh.distribution_id || '-' || expense_cc.
code_combination_id

データ・セット・キー: fa-recon

Oracle E-Business Suite 表の名前

Xla_Ae_Lines Lines

ECC_SPEC_ID 列

FA_AEL.TRX_HDR_ID FA_AEL.ADJUSTMENT_LINE_ID
LINES.AE_HEADER_ID
LINES.AE_LINE_NUM
Lines.GL_SL_LINK_ID

ECC_SPEC_ID 連結構造

```
FA_AEL.TRX_HDR_ID || '-' || FA_AEL.ADJUSTMENT_LINE_ID || '-' || LINES.  
AE_HEADER_ID || '-' || LINES.AE_LINE_NUM || '-' || lines.GL_SL_LINK_ID
```

データ・セット・キー: fa-masstrans

Oracle E-Business Suite 表の名前

```
FA_MASS_TRANSFERS mt  
fa_mass_revaluations mr,  
FA_MASS_REVALUATION_RULES mrr  
fa_mass_changes mc  
FA_MASS_RECLASS mr  
FA_MASS_RETIREMENTS mt  
fa_mass_additions ma
```

ECC_SPEC_ID 列

```
NVL(mt.CONCURRENT_REQUEST_ID, 0)  
mt.MASS_TRANSFER_ID  
NVL(mr.last_request_id, 0)  
mr.mass_reval_id  
mrr.ASSET_ID  
NVL(mc.concurrent_request_id, 0)  
mc.mass_change_id  
NVL(mr.concurrent_request_id, 0)  
mr.MASS_RECLASS_ID  
NVL(mt.RETIRE_REQUEST_ID, 0)  
mt.MASS_RETIREMENT_ID  
ma.MASS_ADDITION_ID
```

ECC_SPEC_ID 連結構造

```
NVL(mt.CONCURRENT_REQUEST_ID, 0) || '-' || mt.MASS_TRANSFER_ID  
NVL(mr.last_request_id, 0) || '-' || mr.mass_reval_id || '-' || mrr.ASSET_ID  
NVL(mc.concurrent_request_id, 0) || '-' || mc.mass_change_id  
NVL(mr.concurrent_request_id, 0) || '-' || mr.MASS_RECLASS_ID  
NVL(mt.RETIRE_REQUEST_ID, 0) || '-' || mt.MASS_RETIREMENT_ID  
ma.MASS_ADDITION_ID || ''
```

データ・セット・キー: fa-clr

Oracle E-Business Suite 表の名前

```
ap_invoices_all inv  
ap_invoice_lines_all  
invl pa_project_asset_lines_all pal  
pa_project_assets_all paa
```

ECC_SPEC_ID 列

```
inv.invoice_id  
invl.line_number  
invd.invoice_distribution_id  
paa.project_id  
paa.project_asset_id  
pal.project_asset_line_id  
pal.project_asset_line_detail_id
```

ECC_SPEC_ID 連結構造

```
'INV'|| '-'|| inv.invoice_id|| '-'|| invl.line_number|| '-'|| invd.  
invoice_distribution_id  
'PROJ'|| '-'|| paa.project_id|| '-'|| paa.project_asset_id|| '-'|| pal.  
project_asset_line_id|| '-'|| pal.project_asset_line_detail_id
```

Order Management

データ・セット・キー: ont-lines

Oracle E-Business Suite 表の名前

```
oe_order_lines_all
```

ECC_SPEC_ID 列

```
line_id
```

ECC_SPEC_ID 連結構造

```
line_id
```

データ・セット・キー: ont-headers

Oracle E-Business Suite 表の名前

```
oe_order_headers_all
```

ECC_SPEC_ID 列

```
header_id
```

ECC_SPEC_ID 連結構造

```
header_id
```

Service

データ・セット・キー: cs-service

Oracle E-Business Suite 表の名前

```
cs_incidents_all_b,cs_incidents_all_tl,
cs_incident_types_tl,cs_incident_statuses_tl,
jtf_rs_resource_extns,jtf_rs_groups_b,
jtf_rs_groups_tl,cs_incident_urgencies_tl,
cs_incident_severities_tl,hz_parties,
hz_cust_accounts,fnd_territories_tl,
cs_hz_sr_contact_points,mtl_system_items_b_kfv,
mtl_system_items_tl,mtl_categories_kfv,
csi_item_instances,csi_systems_tl,
okc_k_headers_all_b,okc_statuses_tl,
okc_k_lines_b,okc_k_lines_tl,
oks_cov_types_b,oks_cov_types_tl,
jtf_tasks_b,jtf_task_references_b,
fnd_lookup_values,jtf_task_statuses_b,
jtf_task_statuses_tl,hz_relationships,
per_all_people_f,per_phones,hz_locations,
hz_party_sites,
jtf_rs_resource_extns_tl
```

ECC_SPEC_ID 列

```
incident_id
```

ECC_SPEC_ID 連結構造

```
incident_id
```

データ・セット・キー: cs-task

Oracle E-Business Suite 表の名前

```
cs_incidents_all_b,hz_parties,
jtf_tasks_b,jtf_tasks_tl,
jtf_task_types_tl,jtf_task_types_b,
jtf_task_statuses_tl,jtf_task_statuses_b,
jtf_task_priorities_tl,jtf_objects_tl,
hz_locations,hz_party_sites,
fnd_lookup_values,jtf_task_assignments,
jtf_task_all_assignments,jtf_rs_teams_tl,
jtf_rs_resource_extns_tl,jtf_rs_groups_tl,
jtf_rs_resource_extns,jtf_rs_salesreps
```

ECC_SPEC_ID 列

```
task_assignment_id
```

ECC_SPEC_ID 連結構造

```
task_assignment_id
```

Service Contracts (OKS)

データ・セット・キー: oks-contract

Oracle E-Business Suite 表の名前

```
oks_k_ecc_header_v hdr
oks_k_ecc_line_v line
oks_k_ecc_subline_v sline
```

ECC_SPEC_ID 列

```
hdr.contract_operating_unit_id,
hdr.contract_id,
line.line_id, sline.sub_line_id
```

ECC_SPEC_ID 連結構造

```
to_char (hdr.contract_operating_unit_id)
|| '-'
|| to_char (hdr.contract_id)
|| '-'
|| to_char (line.line_id)
|| '-'
|| to_char (sline.sub_line_id)
```

データ・セット・キー: oks-contract-sr

Oracle E-Business Suite 表の名前

```
cs_incident_types_tl type_tl
cs_incident_types_b type
cs_incident_severities_tl sev_tl
cs_incident_severities_b sev
cs_incident_statuses_tl status_tl
cs_incident_statuses_b status
cs_incident_urgencies_b urgency
cs_incidents_all_vl inc
hz_parties hzp
okc_k_headers_all_b oks_hdr
okc_statuses_b sts
```

ECC_SPEC_ID 列

```
inc.incident_id,
inc.incident_type_id,
inc.org_id
```

ECC_SPEC_ID 連結構造

```
'SR - '
|| to_char (inc.incident_id)
|| '-'
|| to_char (inc.incident_type_id)
|| '-'
|| to_char (inc.org_id)
```

データ・セット・キー: oks-contract-risk

Oracle E-Business Suite 表の名前

oks_k_ecc_header_v

ECC_SPEC_ID 列

contract_id

ECC_SPEC_ID 連結構造

contract_id

データ・セット・キー: oks-contract-contact

Oracle E-Business Suite 表の名前

okc_contacts oc_sale
okx_party_contacts_v opc
fnd_lookup_values fndv
okc_k_headers_all_b okh
okc_statuses_b sts

ECC_SPEC_ID 列

dnz_chr_id

ECC_SPEC_ID 連結構造

dnz_chr_id

データ・セット・キー: oks-contract-renewal

Oracle E-Business Suite 表の名前

okc_k_headers_all_b okchdr
oks_k_headers_b khdr
okc_statuses_b okcsts
okc_operation_lines oplines
okc_operation_instances opinst
okc_class_operations classops

ECC_SPEC_ID 列

okchdr.id

ECC_SPEC_ID 連結構造

okchdr.id

データ・セット・キー: oks-contract-salescredit

Oracle E-Business Suite 表の名前

```
oks_k_sales_credits linesales
jtf_rs_salesreps linejtfrs
jtf_rs_groups_tl linegrp
oe_sales_credit_types lineoe
okc_k_headers_all_b salelinehdr
okc_k_lines_b saleline
```

ECC_SPEC_ID 列

```
linesales.id
```

ECC_SPEC_ID 連結構造

```
linesales.id
```

データ・セット・キー: oks-contract-customer

Oracle E-Business Suite 表の名前

```
HZ_ORGANIZATION_PROFILES Orgpro ,
HZ_PARTIES Party ,
HZ_CREDIT_RATINGS Credit,
HZ_PARTY_SITES ps,
HZ_LOCATIONS loc,
oks_endeca_tca_finrep_v tca_finrep ,
okc_k_party_roles_b CUSTOMER,
okc_k_headers_all_b c
```

ECC_SPEC_ID 列

```
c.id
```

ECC_SPEC_ID 連結構造

```
c.id
```

Incentive Compensation

データ・セット・キー: cn-cp

Oracle E-Business Suite 表の名前

```
CN_PROCESS_BATCHES_ALL
CN_CALC_SUBMISSION_BATCHES_ALL
FND_CONCURRENT_REQUESTS
```

ECC_SPEC_ID 列

```
CN_PROCESS_BATCHES_ALL.LOGICAL_BATCH_ID
FND_CONCURRENT_REQUESTS.REQUEST_ID
CN_CALC_SUBMISSION_BATCHES_ALL.CALC_SUB_BATCH_ID
```

ECC_SPEC_ID 連結構造

```
CN_PROCESS_BATCHES_ALL.LOGICAL_BATCH_ID || ' ' ECC_SPEC_ID,  
FND_CONCURRENT_REQUESTS.REQUEST_ID || ' ' ECC_SPEC_ID,  
CN_CALC_SUBMISSION_BATCHES_ALL.CALC_SUB_BATCH_ID || ' ' ECC_SPEC_ID,
```

データ・セット・キー: cn-trx

Oracle E-Business Suite 表の名前

```
CN_COMM_LINES_API  
CN_COMMISSION_HEADERS  
CN_COMMISSION_LINES  
CN_PAYMENT_TRANSACTION  
CN_PAYMENT_WORKSHEETS  
CN_PAYRUNS
```

ECC_SPEC_ID 列

```
cn_comm_lines_api_all.COMM_LINES_API_ID  
CN_COMMISSION_LINES.COMMISSION_LINE_ID  
CN_PAYMENT_TRANSACTIONS_ALL.PAYMENT_TRANSACTION_ID  
cn_payment_worksheets_ALL.PAYMENT_WORKSHEET_ID
```

ECC_SPEC_ID 連結構造

```
cn_comm_lines_api_all.COMM_LINES_API_ID || ' API') || ' ' ||  
CN_COMMISSION_LINES.COMMISSION_LINE_ID || ' LINE' || ' ' ||  
CN_PAYMENT_TRANSACTIONS_ALL.PAYMENT_TRANSACTION_ID || ' ' ||  
cn_payment_worksheets_ALL.PAYMENT_WORKSHEET_ID || ' PAY'  
AS ECC_SPEC_ID
```

iProcurement

データ・セット・キー: icx-catalog

Oracle E-Business Suite 表の名前

```
icx_cat_items_ctx_hdrs_tlp ctx
po_system_parameters_all psp,
icx_cat_attribute_values av,
icx_cat_attribute_values_tlp avtlp,
ap_supplier_sites_all pvs,
mtl_categories_kfv mck,
mtl_system_items_kfv msikfv,
mtl_units_of_measure_tl muomtl,
gl_sets_of_books gsob,
financials_system_params_all fsp,
fnd_lookup_values flv,
icx_rvw_reviews_summary_tbl reviews,
icx_cat_categories_tl cat_tl
icx_cat_punchout_items ctx,
icx_cat_pch_item_attrs av,
icx_cat_pch_item_attrs_tlp avtlp,
ap_suppliers vendor,
icx_cat_content_zones_b zones,
icx_cat_punchout_zone_details dtls,
icx_cat_content_zones_tl tl,
por_noncat_templates_all_b b,
por_noncat_templates_all_tl tl,
po_vendors pv
```

ECC_SPEC_ID 列

```
ctx.inventory_item_id,
ctx.po_line_id,
ctx.req_template_name,
ctx.req_template_line_num,
ctx.org_id,
ctx.punchout_item_id,
b.zone_id,
b.template_id
```

ECC_SPEC_ID 連結構造

1. ctx.inventory_item_id || '#' || ctx.po_line_id || '#' || ctx.req_template_name || '#' || ctx.req_template_line_num || '#' || ctx.org_id "ECC_SPEC_ID"
2. TO_CHAR(ctx.punchout_item_id) ecc_spec_id,
3. TO_CHAR(b.zone_id) ecc_spec_id
4. TO_CHAR(b.template_id) ecc_spec_id

Contract Lifecycle Management for Public Sector

Oracle E-Business Suite 表の名前

```
PO_REQUISITION_HEADERS_ALL PRH
PO_REQUISITION_LINES_ALL PRL
PO_REQ_DISTRIBUTIONS_ALL PRD
PO_DRAFTS PD
PO_CLM_ASSIGNMENTS PCA
PO_CLM_ASSGNMT_MILESTONE_DTLS PMD
```

ECC_SPEC_ID 列

```
PRH.REQUISITION_HEADER_ID
PRL.REQUISITION_LINE_ID
PRD.DISTRIBUTION_ID
PCA.ASSIGNMENT_ID
PMD.MILESTONE_CODE
PMD.MILESTONE_TYPE
PRH.PAR_DRAFT_ID
PRH.PAR_LINE_ID
```

ECC_SPEC_ID 連結構造

```
1. REQUISITION_HEADER_ID
   || '-'
   || REQUISITION_LINE_ID
   || '-'
   || DISTRIBUTION_ID
   || '-'
   || ASSIGNMENT_ID
   || '-'
   || MILESTONE_CODE
   || '-'
   || MILESTONE_TYPE
   || '-'
   || PAR_DRAFT_ID
   || '-'
   || PAR_LINE_ID ECC_SPEC_ID

2. ASSIGNMENT_ID
   || '-'
   || MILESTONE_CODE
   || '-'
   || MILESTONE_TYPE ECC_SPEC_ID
```

データ・セット・キー: po-clm-award

Oracle E-Business Suite 表の名前

```
PO_HEADERS_ALL
PO_DRAFTS
PO_LINES_ALL
PO_LINE_LOCATIONS_ALL
PO_DISTRIBUTIONS_ALL
```

ECC_SPEC_ID 列

PO_HEADER_ID
DRAFT_ID
PO_LINE_ID
LINE_LOCATION_ID
PO_DISTRIBUTION_ID

ECC_SPEC_ID 連結構造

PO_HEADER_ID || '-' || DRAFT_ID || '-' || PO_LINE_ID || '-' ||
LINE_LOCATION_ID || '-' || PO_DISTRIBUTION_ID

データ・セット・キー: po-clm-idv

Oracle E-Business Suite 表の名前

PO_HEADERS_ALL
PO_DRAFTS
PO_LINES_ALL

ECC_SPEC_ID 列

PO_HEADER_ID
DRAFT_ID
PO_LINE_ID

ECC_SPEC_ID 連結構造

PO_HEADER_ID || '-' || DRAFT_ID || '-' || PO_LINE_ID

データ・セット・キー: po-clm-par

Oracle E-Business Suite 表の名前

PO_HEADERS_ALL
PO_DRAFTS
PO_LINES_ALL
PO_LINE_LOCATIONS_ALL
PO_DISTRIBUTIONS_ALL

ECC_SPEC_ID 列

PO_HEADER_ID
DRAFT_ID
PO_LINE_ID
LINE_LOCATION_ID
PO_DISTRIBUTION_ID

ECC_SPEC_ID 連結構造

PO_HEADER_ID || '-' || DRAFT_ID || '-' || PO_LINE_ID || '-' ||
LINE_LOCATION_ID || '-' || PO_DISTRIBUTION_ID

データ・セット・キー: po-clm-deliverables

Oracle E-Business Suite 表の名前

OKC_DELIVERABLES
PO_HEADERS_ALL

ECC_SPEC_ID 列

DELIVERABLE_ID
PO_HEADER_ID

ECC_SPEC_ID 連結構造

DELIVERABLE_ID || '-' || PO_HEADER_ID

データ・セット・キー: po-clm-protest

Oracle E-Business Suite 表の名前

PO_PROTESTS
PO_HEADERS_ALL

ECC_SPEC_ID 列

PROTEST_ID
PO_HEADER_ID

ECC_SPEC_ID 連結構造

PROTEST_ID || '-' || PO_HEADER_ID

データ・セット・キー: po-clm-closeout

Oracle E-Business Suite 表の名前

OKC_DELIVERABLES
PO_HEADERS_ALL

ECC_SPEC_ID 列

DELIVERABLE_ID
PO_HEADER_ID

ECC_SPEC_ID 連結構造

DELIVERABLE_ID || '-' || PO_HEADER_ID

データ・セット・キー: po-clm-solicitations

Oracle E-Business Suite 表の名前

pon_auction_headers_all pah
pon_auction_item_prices_all paip

ECC_SPEC_ID 列

```
pah.auction_header_id  
paip.line_number
```

ECC_SPEC_ID 連結構造

```
pah.auction_header_id || '-'  
|| paip.line_number ecc_spec_id
```

データ・セット・キー: po-clm-sol-time

Oracle E-Business Suite 表の名前

```
pon_auction_headers_all pah
```

ECC_SPEC_ID 列

```
pah.auction_header_id
```

ECC_SPEC_ID 連結構造

```
auction_header_id || '-' || duration_type ECC_SPEC_ID
```

データ・セット・キー: po-clm-sol-protests

Oracle E-Business Suite 表の名前

```
pon_auction_headers_all pah  
po_protests prt
```

ECC_SPEC_ID 列

```
pah.auction_header_id  
prt.protest_id
```

ECC_SPEC_ID 連結構造

```
pah.auction_header_id || '-' || prt.protest_id ECC_SPEC_ID
```

データ・セット・キー: po-clm-sol-offers

Oracle E-Business Suite 表の名前

```
pon_auction_headers_all pah  
pon_bid_headers pbh  
pon_bid_item_prices pbip
```

ECC_SPEC_ID 列

```
pah.auction_header_id  
pbh.bid_number  
pbip.line_number
```

ECC_SPEC_ID 連結構造

```
pah.auction_header_id
|| '-'
|| pbh.bid_number
|| '-'
|| pbip.line_number ecc_spec_id
```

データ・セット・キー: po-clm-sol-forms

Oracle E-Business Suite 表の名前

```
pon_auction_headers_all ah,
PON_FORMS_SECTIONS Forms
PON_FORMS_INSTANCES pfi
okc_rep_contracts_all orca
```

ECC_SPEC_ID 列

```
ah.auction_header_id
pfi.form_id
Forms.form_code
orca.contract_number
```

ECC_SPEC_ID 連結構造

1. ah.auction_header_id||'-'||pfi.form_id||'-'||'FEDBIZ'||'-'||Forms.
form_code ECC_SPEC_ID
2. ah.auction_header_id||'-'||'SBCR'||'-'||orca.contract_number
ECC_SPEC_ID

データ・セット・キー: po-clm-sol-deliverables

Oracle E-Business Suite 表の名前

```
pon_auction_headers_all pah
okc_deliverables deliverable
```

ECC_SPEC_ID 列

```
pah.auction_header_id
deliverable.deliverable_id
```

ECC_SPEC_ID 連結構造

```
pah.auction_header_id||'-'||deliverable.deliverable_id as  
ECC_SPEC_ID
```

データ・セット・キー: po-clm-acquisitions

Oracle E-Business Suite 表の名前

```
okc_rep_contracts_all orc
okc_deliverables ord
```

ECC_SPEC_ID 列

```
orc.contract_id  
ord.deliverable_id
```

ECC_SPEC_ID 連結構造

```
orc.contract_id||"-"||ord.deliverable_id as ECC_SPEC_ID
```

Discrete Manufacturing

データ・セット・キー: wip-wodetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs  
mtl_reservations  
oe_order_lines_all
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id  
wdj.organization_id  
mr.supply_source_header_id  
ool1.line_id
```

ECC_SPEC_ID 連結構造

```
(wdj.wip_entity_id||'-'||wdj.organization_id||'-'||mr.  
supply_source_header_id||'-'||ool1.line_id||'-'||JOB')
```

データ・セット・キー: wip-opdetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs  
wip_operations
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id  
wo.operation_seq_num  
wdj.organization_id
```

ECC_SPEC_ID 連結構造

```
(wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||  
wdj.organization_id||'-'||OP')
```

データ・セット・キー: wip-serialedetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs  
mtl_serial_numbers
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id  
wdj.organization_id  
msn.serial_number
```

ECC_SPEC_ID 連結構造

```
(wdj.wip_entity_id || '-' || wdj.organization_id || '-' || msn.  
serial_number || '-SER')
```

データ・セット・キー: wip-compdetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs  
wip_requirement_operations
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id  
wro.operation_seq_num  
wro.inventory_item_id  
wdj.organization_id
```

ECC_SPEC_ID 連結構造

```
(wdj.wip_entity_id || '-' || wro.operation_seq_num || '-' || wro.  
inventory_item_id || '-' || wdj.organization_id || '-COMP')
```

データ・セット・キー: wip-opqualitydetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs  
wip_operations  
wip_move_transactions
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id wdj.organization_id wo.operation_seq_num  
wmt.transaction_id
```

ECC_SPEC_ID 連結構造

```
(wdj.wip_entity_id || '-' || wdj.organization_id || '-' || wo.  
operation_seq_num || '-' || wmt.transaction_id || '-TXN')
```

データ・セット・キー: wip-ospdetails

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs
wip_operations
bom_resources
po_distributions_all
po_releases_all
po_requisition_lines_all
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
prl.requisition_header_id
prl.requisition_line_id
UNION
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
pd.po_header_id
pd.po_line_id
pr.po_release_id
UNION
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
pd.po_header_id
pd.po_line_id
```

ECC_SPEC_ID 連結構造

```
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||prl.
requisition_header_id ||'-'||prl.requisition_line_id||'-'||OSP''
UNION
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||pd.po_header_id
||'-'||pd.po_line_id||'-'||pr.po_release_id||'-'||OSP''
UNION
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||pd.po_header_id
||'-'||pd.po_line_id||'-'||OSP''
```

データ・セット・キー: wip-ospdetails2

Oracle E-Business Suite 表の名前

```
wip_discrete_jobs
wip_operations
bom_resources
po_distributions_all
po_releases_allpo_requisition_lines_all
```

ECC_SPEC_ID 列

```
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
prl.requisition_header_id
prl.requisition_line_id
UNION
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
pd.po_header_id
pd.po_line_id
pr.po_release_id
UNION
wdj.wip_entity_id
wo.operation_seq_num
br.resource_code
wdj.organization_id
pd.po_header_id
pd.po_line_id
```

ECC_SPEC_ID 連結構造

```
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||prl.
requisition_header_id ||'-'||prl.requisition_line_id||'-OSP'
UNION
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||pd.po_header_id
||'-'||pd.po_line_id|| ' '-'||pr.po_release_id||'-OSP'
UNION
wdj.wip_entity_id||'-'||wo.operation_seq_num||'-'||br.
resource_code||'-'||wdj.organization_id||'-'||pd.po_header_id
||'-'||pd.po_line_id||'-OSP'
```

Outsourced Manufacturing

データ・セット・キー: jmf-subcon

Oracle E-Business Suite 表の名前

```
jmf_shikyu_replenishments (jsr)
jmf_subcontract_orders (jso)
jmf_shikyu_adjustments (jsadj)
jmf_shikyu_components (jsc)
qa_results (qr)
```

ECC_SPEC_ID 列

jsr.replenishment_so_line_id
jsr.replenishment_so_header_id
jsr.replenishment_po_header_id
jsr.replenishment_po_line_id
jsr.shikyu_component_id
jso.subcontract_po_shipment_id
jsadj.adjustment_id
jso.subcontract_po_shipment_id
jsc.shikyu_component_id
jso.subcontract_po_shipment_id
qr.plan_id
qr.collection_id
qr.occurrence

ECC_SPEC_ID 連結構造

```
jsr.replenishment_so_line_id
|| jsr.replenishment_so_header_id
|| jsr.replenishment_po_header_id
|| jsr.replenishment_po_line_id
|| jsr.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| jsadj.adjustment_id
|| jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| qr.plan_id
|| qr.collection_id
|| qr.occurrence
```

UNION

```
jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jsadj.adjustment_id
|| jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| qr.plan_id
|| qr.collection_id
|| qr.occurrence
```

UNION

```
jsr.replenishment_so_line_id
|| jsr.replenishment_so_header_id
|| jsr.replenishment_po_header_id
|| jsr.replenishment_po_line_id
|| jsr.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| jsadj.adjustment_id
|| jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| qr.plan_id
|| qr.collection_id
|| qr.occurrence
```

UNION

```
jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jsadj.adjustment_id
|| jso.subcontract_po_shipment_id
|| jsc.shikyu_component_id
|| jso.subcontract_po_shipment_id
|| qr.plan_id
|| qr.collection_id
|| qr.occurrence
```

Process Manufacturing

データ・セット・キー: gmo-batch

ECC_SPEC_ID 連結構造

このデータ・セットの ecc_spec_id は、gmo- quality を除くすべてのデータ・セットの結合です。

データ・セット・キー: gmo-batch-md

Oracle E-Business Suite 表の名前

gme_batch_header(hdr)

ECC_SPEC_ID 列

hdr.BATCH_ID

ECC_SPEC_ID 連結構造

'A' || '-' || TO_CHAR(hdr.BATCH_ID)

データ・セット・キー: gmo-batch-q

Oracle E-Business Suite 表の名前

gme_batch_header(hdr)

ECC_SPEC_ID 列

hdr.BATCH_ID

ECC_SPEC_ID 連結構造

'A' || '-' || TO_CHAR(hdr.BATCH_ID)

データ・セット・キー: gmo-material

Oracle E-Business Suite 表の名前

gme_material_details(batchdtl)

ECC_SPEC_ID 列

batchdtl.BATCH_ID
batchdtl.material_detail_id

ECC_SPEC_ID 連結構造

'B' || '-' || batchdtl.BATCH_ID || '-' || batchdtl.material_detail_id

データ・セット・キー: gmo-material-q

Oracle E-Business Suite 表の名前

gme_material_details (batchdtl)

ECC_SPEC_ID 列

batchdtl.BATCH_ID
batchdtl.material_detail_id

ECC_SPEC_ID 連結構造

'B' || '-' || batchdtl.BATCH_ID || '-' || batchdtl.material_detail_id

データ・セット・キー: gmo-ncm

Oracle E-Business Suite 表の名前

gme_batch_header (hdr)
qa_results (ncm)

ECC_SPEC_ID 列

hdr.BATCH_ID
ncm.plan_id
ncm.collection_id
ncm.occurrence

ECC_SPEC_ID 連結構造

'G' || '-' || hdr.BATCH_ID || '-' || ncm.plan_id || '-' || ncm.
collection_id || '-' || ncm.occurrence

データ・セット・キー: gmo-quality

ECC_SPEC_ID 連結構造

このデータ・セットの ecc_spec_id は、gmo-batch を除くすべてのデータ・セットの結合です。

データ・セット・キー: gmo-res

Oracle E-Business Suite 表の名前

GME_BATCH_STEP_RESOURCES (res)

ECC_SPEC_ID 列

res.BATCH_ID
res.batchstep_id
res.batchstep_activity_id
res.BATCHSTEP_RESOURCE_ID
pp.PROCESS_PARAM_ID

ECC_SPEC_ID 連結構造

```
'F' || '-' || res.BATCH_ID || '-' || res.batchstep_id || '-' || res.  
batchstep_activity_id || '-' || res.BATCHSTEP_RESOURCE_ID. UNION 'C' || '-'  
|| res.BATCH_ID || '-' || res.batchstep_id || '-' || res.  
batchstep_activity_id || '-' || res.BATCHSTEP_RESOURCE_ID || '-' || pp.  
PROCESS_PARAM_ID
```

データ・セット・キー: gmo-res-q

Oracle E-Business Suite 表の名前

```
GME_BATCH_STEP_RESOURCES (res)
```

ECC_SPEC_ID 列

```
res.BATCH_ID  
res.batchstep_id  
res.batchstep_activity_id  
res.BATCHSTEP_RESOURCE_ID  
pp.PROCESS_PARAM_ID
```

ECC_SPEC_ID 連結構造

```
'F' || '-' || res.BATCH_ID || '-' || res.batchstep_id || '-' || res.  
batchstep_activity_id || '-' || res.BATCHSTEP_RESOURCE_ID. UNION 'C' || '-'  
|| res.BATCH_ID || '-' || res.batchstep_id || '-' || res.  
batchstep_activity_id || '-' || res.BATCHSTEP_RESOURCE_ID || '-' || pp.  
PROCESS_PARAM_ID
```

データ・セット・キー: gmo-sample

Oracle E-Business Suite 表の名前

```
gmd_samples (s)  
gmd_spec_results (sr)  
gmd_results (r)
```

ECC_SPEC_ID 列

```
s.BATCH_ID  
s.sample_id  
sr.event_spec_disp_id  
r.result_id
```

ECC_SPEC_ID 連結構造

```
'H' || '-' || s.BATCH_ID || '-' || s.sample_id || '-' || sr.  
event_spec_disp_id || '-' || r.result_id
```

データ・セット・キー: gmo-so

Oracle E-Business Suite 表の名前

```
gme_batch_header (hdr)  
mtl_reservations (rsrv)
```

ECC_SPEC_ID 列

hdr.BATCH_ID
rsrv.reservation_id

ECC_SPEC_ID 連結構造

'I' || '-' || hdr.BATCH_ID || '-' || rsrv.reservation_id

データ・セット・キー: gmo-steps

Oracle E-Business Suite 表の名前

GME_BATCH_STEPS (step)

ECC_SPEC_ID 列

step.BATCH_ID
step.batchstep_id

ECC_SPEC_ID 連結構造

'D' || '-' || step.BATCH_ID || '-' || step.batchstep_id

データ・セット・キー: gmo-steps-q

Oracle E-Business Suite 表の名前

GME_BATCH_STEPS (step)

ECC_SPEC_ID 列

step.BATCH_ID
step.batchstep_id

ECC_SPEC_ID 連結構造

'D' || '-' || step.BATCH_ID || '-' || step.batchstep_id

データ・セット・キー: gmo-test

Oracle E-Business Suite 表の名前

gmd_samples (s)
gmd_spec_results (sr)
gmd_results (r)

ECC_SPEC_ID 列

s.BATCH_ID
s.sample_id
sr.event_spec_disp_id
r.result_id

ECC_SPEC_ID 連結構造

```
'H' || '-' || s.BATCH_ID || '-' || s.sample_id || '-' || sr.  
event_spec_disp_id || '-' || r.result_id
```

データ・セット・キー: gmo-exp

Oracle E-Business Suite 表の名前

```
mtl_lot_numbers (mln)
```

ECC_SPEC_ID 列

```
mln. organization_id  
mln. inventory_item_id  
mln. lot_number
```

ECC_SPEC_ID 連結構造

```
'K' || '-' || mln.organization_id || mln.inventory_item_id || mln.lot_number
```

Logistics (Inventory)

データ・セット・キー: Delivery_Details

Oracle E-Business Suite 表の名前

```
wsh_delivery_details
```

ECC_SPEC_ID 列

```
delivery_detail_id
```

ECC_SPEC_ID 連結構造

```
delivery_detail_id
```

データ・セット・キー: inv-onhand

Oracle E-Business Suite 表の名前

```
mtl_system_items_b_kfv (msib)  
mtl_onhand_quantities_detail (moqd)
```

ECC_SPEC_ID 列

```
msib.organization_id  
msib.inventory_item_id  
moqd.revision  
moqd.subinventory_code  
moqd.locator_id  
moqd.lot_numbe
```

ECC_SPEC_ID 連結構造

```
rmsib.organization_id || '--' || msib.inventory_item_id || '--' || moqd.  
revision || '--' || moqd.subinventory_code || '--' || moqd.locator_id  
|| '--' || moqd.lot_number
```

データ・セット・キー: inv-cyclecounting

Oracle E-Business Suite 表の名前

```
mtl_cycle_count_entries (mcce)  
mtl_cycle_count_headers (mcch)  
mtl_parameters (mp)  
mtl_cycle_count_items (mcci)
```

ECC_SPEC_ID 列

```
mcce.cycle_count_entry_id  
mcch.cycle_count_header_id  
mp.organization_id  
mcci.inventory_item_id
```

ECC_SPEC_ID 連結構造

```
DECODE(mcce.cycle_count_entry_id, NULL, mcch.cycle_count_header_id ||  
'--' || mp.organization_id || '--' || mcci.inventory_item_id, mcce.  
cycle_count_entry_id)
```

Channel Revenue Management

データ・セット・キー: ozf-open-claim

Oracle E-Business Suite 表の名前

```
OZF_CLAIMS_ALL C
```

ECC_SPEC_ID 列

```
C.CLAIM_ID
```

ECC_SPEC_ID 連結構造

```
C.CLAIM_ID || '-' || null || '-' || null
```

データ・セット・キー: ozf-closed-claim

Oracle E-Business Suite 表の名前

```
OZF_CLAIMS_ALL C  
OZF_CLAIM_LINES_ALL L  
OZF_FUNDS_UTILIZED_ALL_B FU  
QP_LIST_HEADERS_B QP
```

ECC_SPEC_ID 列

```
C.CLAIM_ID  
QP.LIST_HEADER_ID  
FU.PRODUCT_ID  
L.ITEM_ID
```

ECC_SPEC_ID 連結構造

```
C.CLAIM_ID || '-' || QP.LIST_HEADER_ID || '-' || NVL(FU.PRODUCT_ID,L.  
ITEM_ID)
```

データ・セット・キー: ozf-budget

OZF Oracle E-Business Suite 表の名前

```
OZF_FUNDS_UTILIZED_ALL_B  
OZF_FUNDS_ALL_B
```

ECC_SPEC_ID 列

```
FUND_ID  
PLAN_TYPE  
PLAN_ID  
BILLTO_CUST_ACCOUNT_ID  
CUST_ACCOUNT_ID  
BILL_TO_SITE_USE_ID  
SHIP_TO_SITE_USE_ID  
PRODUCT_LEVEL_TYPE  
PRODUCT_ID  
ACTIVITY_DATE
```

ECC_SPEC_ID 連結構造

```
fund_id || '-' || plan_type || '-' || plan_id || '-' ||  
billto_cust_account_id || '-' ||  
cust_account_id || '-' || bill_to_site_use_id || '-' ||  
ship_to_site_use_id || '-' ||  
product_level_type || '-' || product_id || '-' || activity_date || '-'  
|| type_flag
```

Cost Management

データ・セット・キー: cst-phc-invt

Oracle E-Business Suite 表の名前

mtl_material_transactions (MMT)
mtl_transactions_interface (MTI)
mtl_material_transactions_temp (MMTT)
WSH_DELIVERY_DETAILS (WSH)
CST_LC_ADJ_INTERFACE (LCM)
gmf_lc_adj_headers_v (glah)
gmf_lc_adj_transactions (glat)
gmf_period_balances (GPB)
gmf_lot_cost_adjustments (lca)
cm_adj_dtl (cad)
gl_aloc_dtl (dtl)
GMF_INVOICE_DISTRIBUTIONS (GID)

ECC_SPEC_ID 列

MMT.rowid
MTI.rowid
MMTT.Rowid
WSH.ROW_ID
LCM.ROWID
GLAT.ROWID
GLAT.ship_header_id
GLAT.ship_line_id
GLAT.adj_transaction_id
GPB.PERIOD_BALANCE_ID
GPB.INVENTORY_ITEM_ID
GPB.ROWID
LCA.ADJUSTMENT_ID
LCA.INVENTORY_ITEM_ID
LCA.ROWID
CAD.COST_ADJUST_ID
CAD.ROWID
DTL.ALLOC_ID
MSI.INVENTORY_ITEM_ID
DTL.LINE_NO
DTL.ALLOCDTL_ID
DTL.ROWID
gid.rowid

ECC_SPEC_ID 連結構造

```
'MMT' || MMT.rowid
UNION ALL
'MMT' || MMT.rowid
UNION ALL
'MTI' || MTI.rowid
UNION ALL
'MMTT' || MMTT.Rowid
UNION ALL
'WSH' || WSH.ROW_ID
UNION ALL
'LCM' || LCM.ROWID
UNION ALL
'LCM' || RT.ROWID || GLAH.ship_header_id || GLAH.ship_line_id || glah.
adjustment_num || glah.component_type || glah.cost_cmpntcls_id || glah.
cost_analysis_code
UNION ALL
'LCM' || GLAT.ROWID || GLAT.ship_header_id || GLAT.ship_line_id ||
GLAT.
adj_transaction_id
UNION ALL
'GPB' || GPB.PERIOD_BALANCE_ID || '-' || GPB.INVENTORY_ITEM_ID || GPB.
ROWID
UNION ALL
'LCA' || LCA.ADJUSTMENT_ID || '-' || LCA.INVENTORY_ITEM_ID || LCA.ROWID
UNION ALL
'CAD' || CAD.COST_ADJUST_ID || CAD.ROWID
UNION ALL
'GLEXP' || DTL.ALLOC_ID || MSI.INVENTORY_ITEM_ID || DTL.LINE_NO || DTL.
ALLOCDTL_ID || DTL.ROWID
UNION ALL
'GID' || gid.rowid
```

データ・セット・キー: st-phc-rcv

Oracle E-Business Suite 表の名前

```
rcv_transactions_interface (rti)
gmf_rcv_accounting_txns (grat)
```

ECC_SPEC_ID 列

```
RTI.rowid
GRAT.rowid
GRAT.rowid
```

ECC_SPEC_ID 連結構造

```
'RTI' || RTI.rowid
UNION ALL
'GRAT' || GRAT.rowid
UNION ALL
'GRAT' || GRAT.rowid
```

データ・セット・キー: cst-phc-gmfmfg

Oracle E-Business Suite 表の名前

```
gme_resource_txns (rt)
```

ECC_SPEC_ID 列

RT.rowid

ECC_SPEC_ID 連結構造

'RSRC' || RT.ROWID

データ・セット・キー: cst-phc-mfg-dis

Oracle E-Business Suite 表の名前

WIP_COST_TXN_INTERFACE (wcti)
WIP_MOVE_TXN_INTERFACE (wmti)
WSM_LOT_MOVE_TXN_INTERFACE (wlmti)
WSM_LOT_SPLIT_MERGES_INTERFACE (wlsmi)
WSM_SPLIT_MERGE_TXN_INTERFACE (wsmti)

ECC_SPEC_ID 列

WCTI.ROWID
WMTI.ROWID
wlmti.rowid
wlsmi.rowid
wsmti.rowid

ECC_SPEC_ID 連結構造

'WCTI' || WCTI.ROWID
UNION ALL
'WMTI' || WMTI.ROWID
UNION ALL
'WSMLMIT' || wlmti.rowid
UNION ALL
'WSMLSMTI' || wlsmi.rowid
UNION ALL
'WSMSMTI' || wsmti.rowid

データ・セット・キー: cst-phc-eam

Oracle E-Business Suite 表の名前

wip_discrete_jobs

ECC_SPEC_ID 列

WDJ.rowid

ECC_SPEC_ID 連結構造

'EAM' || WDJ.ROWID

データ・セット・キー: cst-phc-ovw

Oracle E-Business Suite 表の名前

```
mtl_material_transactions (MMT)
mtl_transactions_interface (MTI)
mtl_material_transactions_temp (MMTT)
WSH_DELIVERY_DETAILS (WSH)
CST_LC_ADJ_INTERFACE (LCM)
gmf_lc_adj_headers_v (glah)
gmf_lc_adj_transactions (glat)
gmf_period_balances (GPB)
gmf_lot_cost_adjustments (lca)
cm_adj_dtl (cad)
gl_alloc_dtl (dtl)
GMF_INVOICE_DISTRIBUTIONS (GID)
rcv_transactions_interface (rti)
gmf_rcv_accounting_txns (grat)
gme_resource_txns (rt)
WIP_COST_TXN_INTERFACE (wcti)
WIP_MOVE_TXN_INTERFACE (wmti)
WSM_LOT_MOVE_TXN_INTERFACE (wlmti)
WSM_LOT_SPLIT_MERGES_INTERFACE (wlsmi)
WSM_SPLIT_MERGE_TXN_INTERFACE (wsmti)
wip_discrete_jobs
```

ECC_SPEC_ID 列

```
MMT.rowid
MTI.rowid
MMTT.Rowid
WSH.ROW_ID
LCM.ROWID
GLAT.ROWID
GLAT.ship_header_id
GLAT.ship_line_id
GLAT.adj_transaction_id
GPB.PERIOD_BALANCE_ID
GPB.INVENTORY_ITEM_ID
GPB.ROWID
LCA.ADJUSTMENT_ID
LCA.INVENTORY_ITEM_ID
LCA.ROWID
CAD.COST_ADJUST_ID
CAD.ROWID
DTL.ALLOC_ID
MSI.INVENTORY_ITEM_ID
DTL.LINE_NO
DTL.ALLOCDTL_ID
DTL.ROWID
gid.rowid
RTI.rowid
GRAT.rowid
GRAT.rowid
RT.rowid
WCTI.ROWID
WMTI.ROWID
wlmti.rowid
wlsmi.rowid
wsmti.rowid
WDJ.rowid
```

ECC_SPEC_ID 連結構造

```
'MMT' || MMT.rowid
UNION ALL
'MMT' || MMT.rowid
UNION ALL
'MTI' || MTI.rowid
UNION ALL
'MMTT' || MMTT.Rowid
UNION ALL
'WSH' || WSH.ROW_ID
UNION ALL
'LCM' || LCM.ROWID
UNION ALL
'LCM' || RT.ROWID || GLAH.ship_header_id || GLAH.ship_line_id || glah.
adjustment_num || glah.component_type || glah.cost_cmpntcls_id || glah.
cost_analysis_code
UNION ALL
'LCM' || GLAT.ROWID || GLAT.ship_header_id || GLAT.ship_line_id ||
GLAT.
adj_transaction_id
UNION ALL
'GPB' || GPB.PERIOD_BALANCE_ID || '-' || GPB.INVENTORY_ITEM_ID || GPB.
ROWID
UNION ALL
'LCA' || LCA.ADJUSTMENT_ID || '-' || LCA.INVENTORY_ITEM_ID || LCA.ROWID
UNION ALL
'CAD' || CAD.COST_ADJUST_ID || CAD.ROWID
UNION ALL
'GLEXP' || DTL.ALLOC_ID || MSI.INVENTORY_ITEM_ID || DTL.LINE_NO || DTL.
ALLOCDTL_ID || DTL.ROWID
UNION ALL
'GID' || gid.rowid
UNION ALL
'RTI' || RTI.rowid
UNION ALL
'GRAT' || GRAT.rowid
UNION ALL
'GRAT' || GRAT.rowid
UNION ALL
'RSRC' || RT.ROWID
UNION ALL
'WCTI' || WCTI.ROWID
UNION ALL
'WMTI' || WMTI.ROWID
UNION ALL
'WSMLMIT' || wlmmti.rowid
UNION ALL
'WSMLSMTI' || wlsmti.rowid
UNION ALL
'WSMSMTI' || wsmti.rowid
UNION ALL
'EAM' || WDJ.ROWID
```

Procurement

データ・セット・キー: po-pcc-agreements

Oracle E-Business Suite 表の名前

```
po_headers_all  
po_lines_all  
okc_deliverables
```

ECC_SPEC_ID 列

```
poh.po_header_id  
pla.po_line_id  
del.deliverable_id
```

ECC_SPEC_ID 連結構造

```
poh.po_header_id || '-' || pla.po_line_id || '-' || del.deliverable_id
```

データ・セット・キー: po-pcc-requisitions

Oracle E-Business Suite 表の名前

```
po_requisition_headers_all  
po_requisition_lines_all  
po_req_distributions_all
```

ECC_SPEC_ID 列

```
req_header.requisition_header_id  
req_line.requisition_line_id  
req_dist.distribution_Id
```

ECC_SPEC_ID 連結構造

```
req_header.requisition_header_id || '-' || req_line.  
requisition_line_id || '-' || req_dist.distribution_Id
```

データ・セット・キー: po-pcc-orders

Oracle E-Business Suite 表の名前

```
po_headers_all  
po_lines_all  
po_line_locations_all  
po_distributions_all  
okc_deliverables
```

ECC_SPEC_ID 列

```
poh.po_header_id  
pla.po_line_id  
plla.line_location_id  
pda.PO_DISTRIBUTION_ID  
del.deliverable_id
```

ECC_SPEC_ID 連結構造

```
poh.po_header_id||'-'||pla.po_line_id||'-'||plla.  
line_location_id||'-'||pda.PO_DISTRIBUTION_ID || '-' ||del.  
deliverable_id
```

データ・セット・キー: po-pcc-orders-metric

Oracle E-Business Suite 表の名前

```
ap_holds_all
```

ECC_SPEC_ID 列

```
aha.INVOICE_ID  
aha.LINE_LOCATION_ID  
aha.HOLD_LOOKUP_CODE
```

ECC_SPEC_ID 連結構造

```
aha.INVOICE_ID ||'-'|| aha.LINE_LOCATION_ID ||'-'||REPLACE(aha.  
HOLD_LOOKUP_CODE,' ','_')
```

データ・セット・キー: po-pcc-orders-prccycletimegraph

Oracle E-Business Suite 表の名前

```
po_headers_all
```

ECC_SPEC_ID 列

```
poh.po_header_id
```

ECC_SPEC_ID 連結構造

```
poh.po_header_id
```

Depot Repair

データ・セット・キー: csd-repairs

Oracle E-Business Suite 表の名前

```
csd_repairs (cr)
cs_incidents_all_b (sr)
fnd_user
hz_parties
hz_cust_accounts
csd_flow_statuses_b
csd_flow_statuses_tl
csd_repair_types_b
csi_item_instances
cs_incident_types_tl
cs_incident_severities_tl
cs_incident_urgencies_tl
cs_incident_statuses_tl
cs_incidents_all_tl
hz_party_site_use
hz_party_sites
hz_locations
fnd_territories_tl
okc_k_headers_all_b
okc_k_lines_b
csd_repair_estimate
csd_repair_actuals
cs_hz_sr_contact_points
hr_organization_information
csd_return_types_tl
jtf_rs_resource_extns
csd_ro_service_codes
csd_ro_diagnostic_codes
csd_diagnostic_codes_b
csd_diagnostic_codes_tl
csd_service_codes_b
csd_service_codes_tl
```

ECC_SPEC_ID 列

```
cr.repair_line_id, sr.incident_id
```

ECC_SPEC_ID 連結構造

```
'CSD-RO-' || cr.repair_line_id || '-' || sr.incident_id
```

Human Resources

データ・セット・キー: PER

Oracle E-Business Suite 表の名前

```
PER_ECC_PEOPLE_V, PER_ECC_ASSIGNMENT_V, PER_ECC_ADDRESS_V,  
PER_ECC_COMPETENCY_V, PER_ECC_PHONE_V,  
PER_ECC_QUALIFICATION_V, PER_ECC_TAG_V,  
PER_ECC_PERFORMANCE_V
```

ECC_SPEC_ID 列

```
person_id, granted_user_id, security_profile_id
```

ECC_SPEC_ID 連結構造

```
person_id || '#' || granted_user_id || '#' || security_profile_id
```

索引

A

アーキテクチャ, 2-1

C

共通のコンポーネント・コントロール, 7-65

コンカレント・プログラム, C-2

D

データ・ロードの概要, 3-1

データ・ロードおよびメタデータ・ロードのシグネチャ,

B-2

データ・ロード・プロセスの定義, 3-2

詳細インサイト・コンポーネント, 7-44

動的属性, 6-1

記述オプション, 6-1

実行オプション, 6-3

動的優先順位ルール, 6-5

E

既存のダッシュボードの拡張, 8-2

既存のデータ・セットの拡張, 8-21

L

レイアウト・コンポーネント, 7-63

M

メタデータ, 5-1, 5-6

アプリケーション, 5-1

データ・セット, 5-4

ページ, 5-2

N

ナビゲーション・コンポーネント, 7-10

O

Oracle Enterprise Command Center Framework の
アーキテクチャ, 2-1

概要, 1-1

P

優先順位ルール動的, 6-5

S

サンプル PL/SQL パッケージ, A-2

セキュリティ

概要, 4-1

U

ユーザー・インタフェース・コンポーネント

共通のコンポーネント・コントロール, 7-65

詳細インサイト, 7-44

レイアウト, 7-63

ナビゲーション, 7-10

概要, 7-1

ビジュアル化, 7-21

V

ビジュアル化コンポーネント, 7-21

