Oracle Tax Analytics

Configuration Guide

Release 1.1.0.0

E24214-01

August 2011

Oracle Tax Analytics Configuration Guide

E24214-01

Contents

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Preface

This guide describes how to configure Oracle Tax Analytics.

This preface contains these topics:

- **Audience**
- **Related Documents**
- **Conventions**

Audience

Oracle Tax Analytics Configuration Guide is intended for system administrators configuring Oracle Tax Analytics.

To use this document you should have:

- Experience installing and configuring application servers and other software
- Administrative privileges on the host where you are installing the software

Related Documents

For more information, see these Oracle documents:

- *Oracle Tax Analytics Installation Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1

Overview

This chapter provides an overview of the configuration of Oracle Tax Analytics, and contains the following main topics:

- **What is Oracle Tax Analytics?**
- **Road Map to Configuring Oracle Tax Analytics**

What is Oracle Tax Analytics?

Oracle Tax Analytics is a prebuilt business intelligence solution that supports the following applications as source systems:

- Oracle Enterprise Taxation and Policy Management version 2.3.0
- Oracle Enterprise Taxation Management version 2.2.0

If you already own one of these applications, you can purchase Oracle Business Intelligence Enterprise Edition and Oracle Tax Analytics to work with the application.

Oracle Tax Analytics consists of the following components:

Component	Description
Oracle Data Integrator E-LT tool	Oracle Data Integrator (ODI) performs the extract, load and transform (E-LT) operations for the data warehouse.
Prebuilt Oracle Integration content	This content includes E-LT repository objects, such as mappings, sessions, and workflows, and is contained in the Oracle Data Integrator repositories.
Prebuilt metadata content	This metadata content is contained in the Oracle Tax Analytics repository file (OracleTaxAnalytics.rpd).
Prebuilt reports and dashboard content	This content is contained in the Oracle BI Presentation Services Catalog.
Oracle Tax Analytics Warehouse	This is the prebuilt data warehouse that holds data extracted, transformed, and loaded from the transactional database. For more information, see the *Oracle Tax Analytics Installation Guide*.

Road Map to Configuring Oracle Tax Analytics

To configure and customize Oracle Tax Analytics, do the following:

- Perform any required configuration steps for the tax analytics that you deploy from the appropriate chapters below:
    - For all applications, see **Chapter 2**: **Configuring Common Areas and Dimensions**.
    - For Tax Account Analytics, see **Chapter 3**: **Configuring Oracle Tax Accounting Analytics**.
    - For Tax Form Line Analytics, see **Chapter 4**: **Configuring Oracle Tax Form Line Items Analytics**.
    - For customizing the target and threshold for any of the provided KPIs, see **Chapter 5**: **Configuring Scorecard After Full Load**.
- If you want to modify the out-of-the-box Oracle Tax Analytics Repository (the RPD file), see **Chapter 6**: **Configuring the Oracle Business Intelligence Repository**.

Chapter 2

Configuring Common Areas and Dimensions

This section contains mandatory and additional configuration tasks that apply to Oracle Tax Analytics deployed with the Oracle Enterprise Taxation and Policy Management source system. It contains the following main topics:

- **Source-Independent Configuration Steps**
- **Configuring the Oracle Tax Analytics Project Common Areas Before a Full Load**

> **Note:** Many configuration tasks described in this guide require you to manually enter values obtained from querying your source system. These values are unique to your source system and implementation. To retrieve these values successfully, you need to have a good technical understanding of your source system. If you need assistance in obtaining values from your source system, you should consult with someone in your organization who possesses this knowledge, or consult the Oracle Support Services team for your source system. Make sure to check your configuration entries carefully to avoid data loss during the E-LT process. After configuring the variables in any subject area, the respective scenarios need to be regenerated.

Source-Independent Configuration Steps

This section contains configuration steps that apply to Oracle Tax Analytics deployed with the Oracle Enterprise Taxation and Policy Management source system. It contains the following topics:

- **Configuring Calendars**

Configuring Calendars

This section explains how to set up the different types of calendar that are supported by Oracle Tax Analytics, and contains the following topics:

- **Overview of Calendars in Oracle Tax Analytics**
- **About Configuring Calendars**
- **Notes on Configuring Calendars**
- **How to Set Up the Gregorian Calendar Date Range**
- **How to Reload the Time Dimension Tables After the Data Warehouse Is Loaded**

Overview of Calendars in Oracle Tax Analytics

Oracle Tax Analytics version 1.1.0 supports the following calendar formats:

- **Gregorian.** Regular calendar that starts on January 1st and ends on December 31st.

About Calendar Tables

This section describes the tables used for Time Dimension calendars.

Gregorian calendar tables:

- W_DAY_D

For more information about the configuration tables and context tables, see **About Configuring Calendars** on page 2-2.

About Calendar Categories

Calendars are categorized into two types:

- OLTP sourced (also known as Source Calendars)

  OLTP sourced calendars are calendars that are defined in ERP sources and brought into the warehouse via ETL maps.

- Warehouse generated (also known as Generated Calendars)

  Generated calendars are Gregorian calendars generated in the warehouse based on configuration files in the procedures.

About Configuring Calendars

This section explains how to configure the different types of supported calendar and contains the following topics:

- **About Setting up Gregorian Calendars**
- **Prerequisites to Populating W_DAY_D and W_MONTH_D Tables**

About Setting up Gregorian Calendars

Whatever calendar type you want to deploy, you must set up the start date and end date range for the Gregorian calendar. For more information, see **How to Set Up the Gregorian Calendar Date Range** on page 2-6.

Prerequisites to Populating W_DAY_D and W_MONTH_D Tables

W_DAY_D is the base table that represents the Day dimension in the Oracle Tax Analytics Warehouse. This table needs to be populated as a prerequisite for the multiple Gregorian calendar tables.

There are three parameters $$DAY_D_START_DT, $$DAY_D_END_DT and DAY_D_FIRST_WEEK_DAY that need to be setup to load the calendar data in W_DAY_D. The procedure use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W_DAY_D, the fact and aggregate tables are loaded by their respective interfaces.

W_MONTH_D is the base table that represents the Month dimension in the Oracle Tax Analytics Warehouse. This table needs to be populated as a prerequisite for the multiple Gregorian calendar tables.

There are two parameters $$MONTH_D_START_DT and $$MONTH_D_END_DT that need to be setup to load the calendar data in W_MONTH_D. The procedure use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W_MONTH_D, the aggregate tables are loaded by their respective interfaces.

> **Note:** The parameters $$DAY_D_START_DT, $$DAY_D_END_DT, $$MONTH_D_START_DT and $$MONTH_D_END_DT need to include all dates covered by any of the Gregorian calendars brought into the warehouse. These parameters are the boundaries for the date dimension and related tables.

Notes on Configuring Calendars

When you set up calendars, note the following:

The W_DAY_D table controls how generated calendars are created.

The following table outlines columns in the W_DAY_D table, which is loaded from the procedure Populate_Day_Dim_Prc.

Column Name	Column Description
ROW_WID	The ID of the calendar that is being configured. This is the primary key for this table.
CALENDAR_DATE	The date of the calendar that is being configured.
DAY_DT	The date of the calendar that is being configured.
PER_NAME_MONTH	The type of the calendar year and month that is being configured e.g. '1980 / 01'
PER_NAME_QTR	The type of the calendar year and quarter that is being configured e.g. '1980 / Q1'

Column Name	Column Description
PER_NAME_WEEK	The type of the calendar year and Week of the year that is being configured e.g. '1980 / Week 01'. This variable is used for selecting the start day of the Week for the selected years. Select a number from the below range (1-7), based on how your week should start. This is valid through out the year i.e., from start date to end date. Sunday – 1 Monday – 2 Tuesday – 3 Wednesday – 4 Thursday – 5 Friday – 6 Saturday – 7
PER_NAME_YEAR	The type of the calendar year that is being configured e.g. '1980'
CAL_WEEK_END_DT_WID	The offset which identifies the max (per_name_week) group by per_name_week, per_name_qtr and per_name_year date of the calendar.
CAL_QTR_END_DT_WID	The offset which identifies the max (per_name_qtr) group by per_name_week, per_name_qtr and per_name_year date of the calendar.
CAL_YEAR_END_DT_WID	The offset which identifies the max (per_name_year) group by per_name_week, per_name_qtr and per_name_year date of the calendar.
Other standard column	DATASOURCE_NUM_ID is '1' sent through a variable

The following table outlines Task Level Parameters needed for Generated Calendars.

Parameter Name	Parameter Description
$$DAY_D_START_DT	Task: This variable is used for selecting the start date of the Day Dimension table for loading the data into W_DAY_D table. The format should be YYYYMMDD
$$DAY_D_END_DT	Task: This variable is used for selecting the end date of the Day Dimension table for loading the data into W_DAY_D table. The format should be YYYYMMDD

Parameter Name	Parameter Description
$$DAY_D_FIRST_WEEK_DAY	This variable is used for selecting the start day of the Week for the selected years. Select a number from the below range (1-7), based on how your week should start. This is valid through out the year i.e., from start date to end date.

This DAY_D_START_DT is a variable used for selecting the start date of the Day Dimension table for loading the data. The format should be YYYYMMDD

This DAY_D_END_DT is a variable used for selecting the end date of the Day Dimension table for loading the data. The format should be YYYYMMDD

This DAY_D_FIRST_WEEK_DAY is a variable used for selecting the start day of the week. If there is a week (starting on a Sunday and ending on a Saturday) that falls across two calendar years, the week is counted in both years. For example, the week that starts on 12/30/2007 will be counted in both 2007 and 2008. In 2007, the week start date will 12/30/2007 and the end date will be 12/31/2007. In 2008, this will be the first week with start date as 01/01/2008 and end date as 01/05/2008.

The W_MONTH_D table controls how generated calendars are created.

The following table outlines columns in the W_MONTH_D table, which is loaded from the procedure Populate_Month_Dim_Prc

Column Name	Column Description
ROW_WID	The ID of the calendar that is being configured. This is the primary key for this table.
CAL_MONTH	The date of the calendar month that is being configured e.g. '1'.
CAL_QTR	The date of the calendar quarter that is being configured. e.g. '1'
CAL_YEAR	The type of the calendar year that is being configured e.g. '1980'
M_END_CAL_DT_WID	The type of the calendar used for loading month end calendar date and that is being configured
M_STRT_CAL_DT_WID	The type of the calendar used for loading month start calendar date and that is being configured
PER_NAME_MONTH	The type of the calendar year and month that is being configured e.g. '1980 / 01'
PER_NAME_QTR	The type of the calendar year and quarter that is being configured e.g. '1980 / Q1'
PER_NAME_YEAR	The type of the calendar year that is being configured e.g. '1980'

Column Name	Column Description
MONTH_AGO_WID	The type of the calendar used for loading previous month calendar date and that is being configured. e.g. '198003'
QUARTER_AGO_WID	The type of the calendar used for loading previous quarter calendar date and that is being configured. e.g. '197912'
YEAR_AGO_WID	The type of the calendar used for loading previous year calendar date and that is being configured. e.g. '197903'

The following table outlines Task Level Parameters needed for Generated Calendars.

Parameter Name	Parameter Description
$$MONTH_D_START_DT	Task: This variable is used for selecting the start date of the Month Dimension table for loading the data into W_MONTH_D table. The format should be YYYYMM
$$MONTH_D_END_DT	Task: This variable is used for selecting the end date of the Month Dimension table for loading the data into W_MONTH_D table. The format should be YYYYMM

This MONTH_D_START_DT is a variable used for selecting the start date of the Month Dimension table for loading the data. The format should be YYYYMM

This MONTH_D_END_DT is a variable used for selecting the end date of the Month Dimension table for loading the data. The format should be YYYYMM.

How to Set Up the Gregorian Calendar Date Range

1. Create variables DAY_D_START_DT, DAY_D_END_DT for selecting the dates which the customer wants, and the given date format should be in YYYYMMDD.
2. Create a variable DAY_D_FIRST_WEEK_DAY for selecting the number on which day the week should get started this is a customer choice, ex., '1' (Sunday -1, Monday -2, and so on,... Saturday - 7)
3. Use the $$DAY_D_START_DT and $$DAY_D_END_DT parameters to specify the date range of the calendar that you want to generate.

   Note: The tasks that load the day dimension will run as part of the Execution Plan for your Subject Areas. Please note that there are no separate Subject Areas for common dimensions. They are included in the core Subject Areas.

To set up the Gregorian calendar for: W_MONTH_D

1. Create variables MONTH_D_START_DT, MONTH_D_END_DT for selecting the dates which the customer wants, and the given date format should be in YYYYMM.
2. Use $$MONTH_D_START_DT and $$MONTH_D_END_DT parameters to specify the date range of the calendar that you want to generate.

   Note: The tasks that load the month dimension will run as part of the Execution Plan for your Subject Areas. Please note that there are no separate

Subject Areas for common dimensions. They are included in the core Subject Areas.

How to Reload the Time Dimension Tables After the Data Warehouse Is Loaded

The data in the time dimension is loaded once during the initial full load.

You might want to extend the range of data that you have in your time dimension some time after the warehouse is in production. In order to achieve this, please follow the steps below

1. To set up the Gregorian calendar for W_DAY_D, use the $$DAY_D_START_DT and $$DAY_D_END_DT parameters to specify the date range of the calendar that you want to generate.

   Note: The tasks that load the day dimension will run as part of the Execution Plan for your Subject Areas.

2. To set up the Gregorian calendar for W_MONTH_D, use $$MONTH_D_START_DT and $$MONTH_D_END_DT parameters to specify the date range of the calendar that you want to generate.

   Note: The tasks that load the month dimension will run as part of the Execution Plan for your Subject Areas.

Configuring the Oracle Tax Analytics Project Common Areas Before a Full Load

This section describes the configuration required for Oracle Tax Analytics applications before a full load.

To configure Oracle Tax Analytics before a full data load:

1. Set the variable “GET_WAREHOUSE_STAGING_SCHEMA_NAME:” in accordance with the staging schema name.
2. Set the variable “GET_WAREHOUSE_TARGET_SCHEMA_NAME” in accordance with the warehouse schema name.
3. Set the variable “COMMIT_FREQUENCY” in accordance with the load on the ETL which will let full load to commit the transactions in between the load.
4. As already mentioned in time dimension section the following variables need to be set before running the full load to load the confirmed dimensions W_DAY_D and W_MONTH_D
   - DAY_D_END_DT
   - DAY_D_START_DT
   - DAY_D_FIRST_WEEK_DAY
   - MONTH_D_START_DT
   - MONTH_D_END_DT
5. Refresh the variable “GET_WAREHOUSE_STAGING_SCHEMA_NAME” if the truncate staging tables procedure need to be executed.
6. Refresh the variable “GET_WAREHOUSE_TARGET_SCHEMA_NAME” if the truncate warehouse tables procedure need to be executed.
7. Execute the procedure “Truncate BISTG Tables” if staging tables need to be truncated.
8. Execute the procedure “Truncate BIADM Tables” if warehouse tables need to be truncated.

Chapter 3

Configuring Oracle Tax Accounting Analytics

This section describes how to configure Oracle Tax Accounting Analytics. It contains the following main topics:

- **Overview of Oracle Tax Accounting Analytics**
- **Configuration Required for Oracle Tax Accounting Analytics Before a Full Load**

Overview of Oracle Tax Accounting Analytics

Oracle Tax Accounting Analytics comprises revenue and tax payer accounting functionality based on accounting processing function capability in Oracle Enterprise Taxation and Policy Management. This section describes how Oracle Enterprise Taxation and Policy Management financial transactions are created and how they affect the Oracle Enterprise Taxation and Policy Management accounting process. It contains the following topics:

- **Financial Transaction Sources**
- **Obligation and Account Balances**
- **Financial Transaction General Ledger**

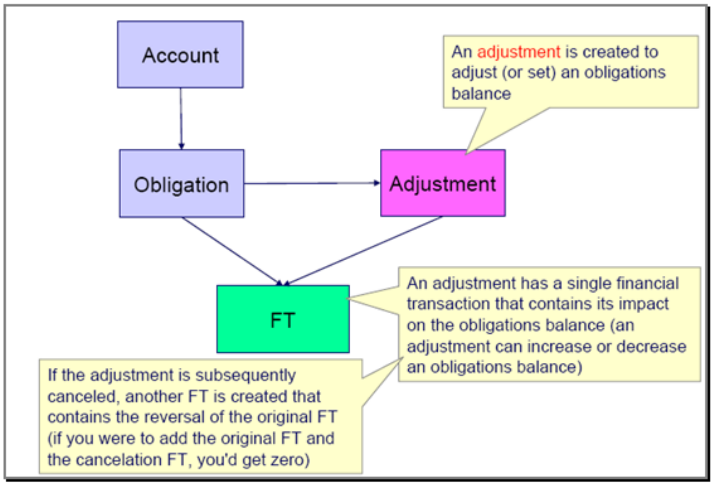
Financial Transaction Sources

Financial Transactions are created by the system on the following events:

- **Adjustments are Created/Cancelled**
- **Payment Segments Are Created/Cancelled**

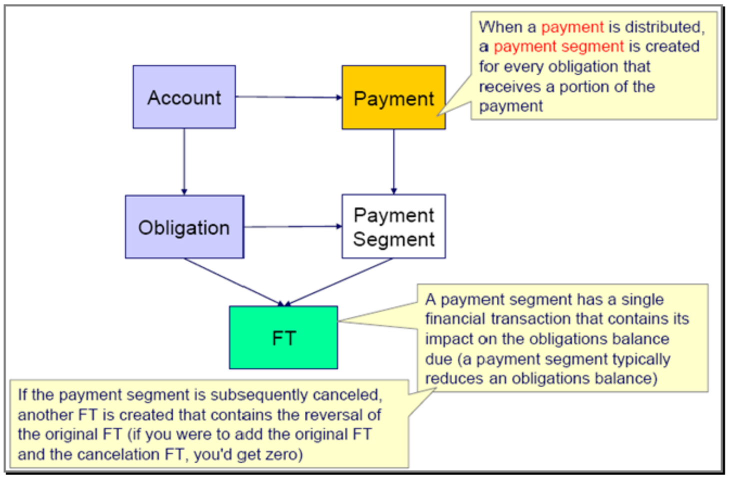
Adjustments are Created/Cancelled

Adjustments are means for the application to adjust or set obligation balances. Adjustments can be created manually and automatically. Best examples of adjustments are assessment creation when a return is filed, assessment of penalty/interest, offsets and balance adjustments are done, etc. To impact the obligation balances, adjustments create financial transaction. The following diagram shows the relationship between an adjustment and a financial transaction (FT).


Account
Obligation
Adjustment
FT
An adjustment is created to adjust (or set) an obligations balance
An adjustment has a single financial transaction that contains its impact on the obligations balance (an adjustment can increase or decrease an obligations balance)
If the adjustment is subsequently canceled, another FT is created that contains the reversal of the original FT (if you were to add the original FT and the cancelation FT, you'd get zero)

Payment Segments Are Created/Cancelled

When the taxpayer makes the payment, it gets distributed to the taxpayer's obligations. Each of these distributed payments is called payment segments. The payment segment has corresponding financial transactions (FTs) to reflect the payment amount to the total obligation balance. The following diagram shows the relationship between a payment and a financial transaction (FT).



When a payment is distributed, a payment segment is created for every obligation that receives a portion of the payment

A payment segment has a single financial transaction that contains its impact on the obligations balance due (a payment segment typically reduces an obligations balance)

If the payment segment is subsequently canceled, another FT is created that contains the reversal of the original FT (if you were to add the original FT and the cancelation FT, you'd get zero)

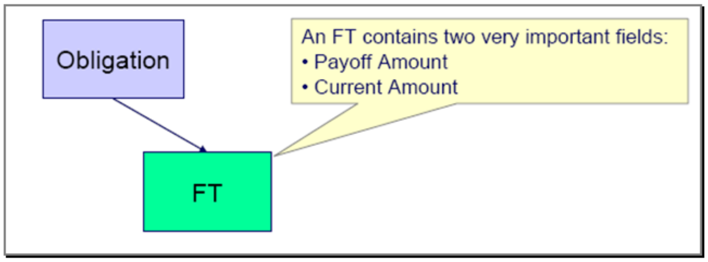
Note: FTs are never cancelled. When source transactions are cancelled, another FT is created to reverse the original. This applies to payment segment and adjustment cancellations.

Obligation and Account Balances

The ETM obligation object does not keep the obligation's due balance. The obligation object has financial transaction information. Getting the sum of all its financial transaction derives the due balance.

In the same manner as obligation, the account object also does not keep its due balance. To calculate the due balance for an account, the sum of all the obligation-linked financial transaction is derived.

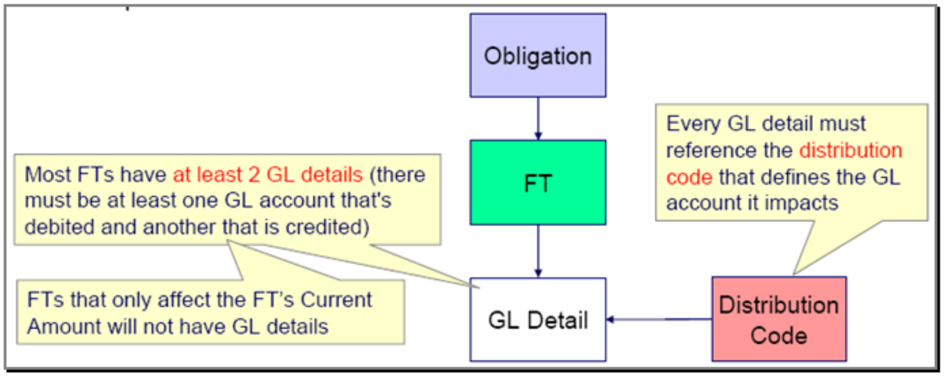
Each of the financial transaction contains two balance values, Current (how much the taxpayer thinks they owe) and Pay-off balances (how much the taxpayer needs to pay upon closing the account). The following diagrams shows the relationship between a financial transaction and an obligation.



An FT contains two very important fields:
- Payoff Amount
- Current Amount

Financial Transaction General Ledger

The financial transaction effect on the general ledger is reflected on its GL details. Each GL detail references a distribution code to determine which GL account it impacts. The following diagram shows the relationship between a financial transaction (FT) and a general ledger (GL) detail.


Obligation
FT
GL Detail
Distribution Code
Most FTs have at least 2 GL details (there must be at least one GL account that's debited and another that is credited)
FTs that only affect the FT's Current Amount will not have GL details
Every GL detail must reference the distribution code that defines the GL account it impacts

The three fact tables of the accounting dimensional design are the obligation fact (OBLIG_FACT), financial transaction fact (FT_FACT) and financial transaction general ledger fact (FT_GL_FACT). Numerical facts can be analyzed by attributes of conformed dimensions, person, tax type, obligation type, account and currency.

The financial transaction fact additionally links to the adjustment type, financial transaction type, GL account type dimensions. While the financial transaction general ledger fact can be further analyzed by general ledger type dimension attributes.

Configuration Required for Oracle Tax Accounting Analytics Before a Full Load

This section contains configuration steps that you need to perform on Oracle Tax Accounting Analytics before you do a full data load.

To configure the Tax Accounting Analytics before a full load, configure the following settings for the FT collections data:

1. Physical and logical architecture should be defined under XML technology as shown in the following table:

Data Server	XML_SOURCE_COLLECTION
JDBC Driver	com.sunopsis.jdbc.driver.xml.SnpsXmlDriver
JDBC URL	jdbc:snps:xml?d=../DirA/xml/W_FT_BAL_CREDIT.dtd&s=FT

   The path given above can be changed in accordance with the project settings. In the path mentioned here, the DTD file is included in the packaging. Note that the FT given above should remain as given.

2. All the variables given in the following table should exist, and should be customized according to the project.

Variable Name	Data Types	Action	Default Value	Select Query	Schema	Configuration
ACCT_LOOP	Numeric	Latest Value	0			Not Required

Variable Name	Data Types	Action	Default Value	Select Query	Schema	Configuration
ACCT_CHECK	Numeric	Latest Value		select count(*) - #ACCT_LOOP from #GET_WAREHOUSE_STAGING_SCHEMA_NAME.W_FT_XML_FILE_NAMES_S	ORACLE_OTOA_TARGET_STG	Not Required
ACCT_XML_FILE_DIR	Alpha Numeric	Historize	..//v100_ta_sf1//xml//			The variable default value needs to be changed (See **To Configure a Variable**, below
ACCT_XMLFILENAME	Alpha Numeric	Latest Value		select xml from (select rownum rnum,'"' || '#ACCT_XML_FILE_DIR' || XmlName || '"' xml from #GET_WAREHOUSE_STAGING_SCHEMA_NAME.W_FT_XML_FILE_NAMES_S) where rnum = #ACCT_LOOP	ORACLE_OTOA_TARGET_STG	Not Required
ACCT_XML_FILE_PATH	Alpha Numeric	Historize	..//v100_ta_sf1//xml			The variable default value needs to be changed (See **To Configure a Variable**, below)
ACCT_FILEEXT	Alpha Numeric	Historize	XML			The variable default value needs to be changed (See **To Configure a Variable**, below)
ACCT_FILESTART	Alpha Numeric	Historize	W_FT_BAL_CREDIT			The variable default value needs to be changed (See **To Configure a Variable**, below)

To Configure a Variable

1. Open the variable.
2. Change the path.
3. Click **Apply,** and then click **OK.**

You should configure variables according to the following examples:

- **ACCT_XML_FILE_DIR:** This is the path for placing all XML files to be handled for collections data.

  For example:
  ..//v100_ta_sf1//xml//

- **ACCT_XML_FILE_PATH:** This is the path where all XML files are placed.

  For example:
  ..//v100_ta_sf1//xml

- **ACCT_FILESTART:** This is the name of the File that is generated from the source system. The files that come from the source may have anything appended but it should start with the given name in this variable, so that it will fetch all the filenames starting with the name given in this variable.

  For Example:
  W_FT_BAL_CREDIT

- **ACCT_FILEEXT:** This is the extension of the file that you get from the source system. It can either be XML or xml.

> **Note:** All the variables are case sensitive Paths and the filenames should be in the same case as that of unix system.

Chapter 4

Configuring Oracle Tax Form Line Items Analytics

This section describes how to configure Oracle Tax Form Line Items Analytics. It contains the following main topics:

- **Overview of Oracle Tax Form Line Items Analytics**
- **Required Configuration Before a Full Load for Oracle Tax Form Line Items Analytics**

Overview of Oracle Tax Form Line Items Analytics

Every taxpayer (individual or business) file their taxes based on the tax type that is applicable. Every tax type requires at least one or more forms to be filed/completed. Each of the form contains related characteristics/information for respective tax/form type. These are referred to as form line items. These line items provides detailed information about taxpayer eg income, exceptions, credits, payments, refunds etc.

Based on the business requirements, there is a need to provide structures (facts/dimensions) along with ETL process for respective structures for the following five tax types:

1. Individual Income
2. Business Income
3. Sales and Use
4. Employer Withholding
5. VAT

For each tax type, two dimensions and one fact table are provided with an ability to map these form line items by customers to pre-wired X_CUSTOM fields in dimensions and fact tables. In addition, the fact table contains foreign keys to Form Type dimension and Person dimension. Also Fact table contains fixed measures for each tax type.

Required Configuration Before a Full Load for Oracle Tax Form Line Items Analytics

This section contains configuration steps that you need to perform on Oracle Tax Forms Analytics before you do a full data load. Configuring the Tax Form Line Items Analytics before a full load requires the following:

The Form Lines table CI_FORM_LINE from the source system should be configured as follows:

1. Set BI_APPL_FLG to C1AP
2. Set BI_TARGET_TBL_NAME to:
   - DIM1 for Dimension1
   - DIM2 for Dimension2
   - FACT for Fact as per the required form lines
3. BI_TARGET_FLD_NAME can be set to:
   - X_CUSTOM1 or X_CUSTOM2 or…. X_CUSTOM10 for Dimension1
   - X_CUSTOM1 or X_CUSTOM2 or…. X_CUSTOM10 for Dimension2
   - X_CUSTOM1 or X_CUSTOM2 or…. X_CUSTOM15 for FACT

   These are assumed as unknown columns and are named as X_CUSTOM

Configuration should be based on form type and form line.

For Example:

FORM_TYPE_CD	BI_TARGET_ TBL_NAME	BI_TARGET_ FLD_NAME	BI_APPL_FLG	DESCRIPTION
INDIV2008	DIM1	X_CUSTOM1	C1AP	This will be mapped to X_CUSTOM1 of W_IND_INC_D1 (Individual Income Dimension1)

FORM_TYPE_CD	BI_TARGET_ TBL_NAME	BI_TARGET_ FLD_NAME	BI_APPL_FLG	DESCRIPTION
SALESUSE2008	DIM2	X_CUSTOM1	C1AP	This will be mapped to X_CUSTOM1 of W_SALES_USE_D2 (SALES USE Dimension2)
INDIV2008	FACT	X_CUSTOM1	C1AP	This will be mapped to X_CUSTOM1 of W_IND_INC_F (Individual Income Fact)

4. Fixed attributes should also be configured in CI_FORM_LINE based on the form type and form line

   Fixed column of Individual Income dimension1 is: FILING STATUS

   Each of the tax types have the following fixed measures:

   - INDIVIDUAL INCOME:
     - TOTAL_EXEMPTIONS
     - ADJUSTED_GROSS_INCOME
     - TAX_ASSESSED
     - TOTAL_INCOME
     - TAXABLE_INCOME
     - NBR_OF_DEPENDANTS
     - TOTAL_CREDITS
   - SALES USE:
     - TOTAL_GROSS_SALES
     - TOTAL_EXEMPT_TRANSACTIONS
     - TAXABLE_TRANSACTIONS
     - TOTAL_ASSESSED_TAX
   - BUSINESS INCOME:
     - TOTAL_INCOME
     - TAXABLE_INCOME
     - TOTAL_TAX
     - TOTAL_DEDUCATION
   - EMPLOYER WITHHOLDING:
     - TAX_WITHHELD
     - PENALTY
     - INTEREST
     - TOTAL_DUE
   - VAT:
     - VAT_CHARGED_BY_RATE_1
     - VAT_CHARGED_BY_RATE_2
     - VAT_CHARGED_OTHR_STATES_RATE_1
     - VAT_CHARGED_OTHR_STATES_RATE_2

- VAT_CHARGED_ON_EXPORT
- VAT_CHARGED_ON_IMPORT

The following variables should exist, and can be customized according to the project requirements:

Variable Name	Data Types	Action	Default Value	Description
FORMLINEITEMS_BI_FLG	AlphaNumeric	Historize	C1AP	BI Applicable flag
FORMLINEITEMS_D1	AlphaNumeric	Historize	DIM1	Dimension1
FORMLINEITEMS_D2	AlphaNumeric	Historize	DIM2	Dimension2
FORMLINEITEMS_FACT	AlphaNumeric	Historize	FACT	Fact
FORMLINEITEMS_DEF_XML_PATH	AlphaNumeric	Historize	asCurrent	Default path for final node
IND_INC	AlphaNumeric	Historize	IND-INCOME	Individual Income tax type
BUS_INC	AlphaNumeric	Historize	CORP-INCOME	Business Income tax type
SALES_USE	AlphaNumeric	Historize	SALES-USE	SALES USE tax type
EMP_WHLDG	AlphaNumeric	Historize	EMP-WHLDG	Employer Withholding tax type
VAT	AlphaNumeric	Historize	VAT	VAT tax type

Chapter 5

Configuring Scorecard After Full Load

This section contains mandatory and additional configuration tasks that apply to Oracle Tax Analytics deployed with the Oracle Enterprise Taxation and Policy Management source system. It includes the following:

- **Configuring the KPI Threshold Tables**

Configuring the KPI Threshold Tables

Tax Analytics provide the flexibility for the users to set their own custom target and threshold values by using the writeback reports. Each of the KPIs used in the KPI Watchlist will make use of its corresponding threshold table's Target, OK and Warning columns. In order to configure these Target, OK and Warning column values for the KPIs, the writeback reports need to be used for updating their custom values. The following table lists the threshold table for each of the Tax Analytic subject areas:

Tax Analytics	Threshold Table
Forms Subject Area	W_FORMS_THRESHOLDS_MONTH_T
Accounting Subject Area	W_FT_THRESHOLDS_MONTH_T
Payments Subject Area	W_PAY_THRESHOLDS_MONTH_T

The time grain for the above tables is the MONTH level. For example, values entered for any Target column for a particular KPI having its aggregation set as SUM in the RPD will sum up all the month values when KPI are analyzed at Year or Quarter Level in the KPI Watchlist. For KPI threshold columns having aggregation set as Average, it fetches the AVG value when KPI are analyzed at Year or Quarter Level, and for KPI threshold columns having aggregation set as Percentage, it fetches the MAX value when KPI analyzed at Year or Quarter Level in the KPI Watchlist.

Configuring the KPI Threshold Tables

Threshold tables (W_FORMS_THRESHOLDS_MONTH_T, W_FT_THRESHOLDS_MONTH_T, W_PAY_THRESHOLDS_MONTH_T) are created in the data warehouse along with the creation of dimensions and fact tables. Check for the existence of the threshold tables in the database by querying with the threshold table name in the data warehouse schema. Ensure that the stored procedures related to scorecard are already deployed. After the E-LT run, execute the procedures related to Scorecard so that it populates the default records. After the successful execution of stored procedures, verify for the existence of default rows by querying the tables W_FORMS_THRESHOLDS_MONTH_T, W_FT_THRESHOLDS_MONTH_T, W_PAY_THRESHOLDS_MONTH_T in the SQL developer.

Example: Adding a KPI Threshold Value

By using the writeback reports we can add our custom target and threshold values. For adding a KPI threshold value, open the writeback reports.

> **Note:** Before performing this procedure, ensure that you have successfully deployed and executed the stored procedures related to Scorecard, and that you have placed the writeback XML templates at the appropriate locations.

1. Open the writeback report from the following path: /Shared Folders/Tax/Forms/Tax Forms/Writeback/Dynamic Thresholds and Targets(KPIs) - Tax Forms.

2. Select the Year value from the prompt for which you want to enter your custom values.

3. Click **Update**.

   This turns all the Target, OK and Warning columns to editable boxes. Revert, Update, and Done buttons are available.

4. If you want to update # Received - Target value to your custom value for Month '2010 / 01', update the value in the box with your custom value.

You can simultaneously enter values in more than one column and then click **Apply**.

5. Click **Done** when you have made all of your changes. You can query the Threshold Table for these values.

Chapter 6

Configuring the Oracle Business Intelligence Repository

This section describes how to configure the Oracle Business Intelligence Repository for use with Oracle Tax Analytics. It contains the following main topics:

- **Setting Up Date-Specific Metrics**
- **Setting Up Additional Time Series Metrics for Oracle Tax Analytics Warehouse**
- **Setting Up Additional Dimension Tables for the Oracle Tax Analytics Warehouse**
- **About the Period Ago Keys for Oracle Tax Analytics Warehouse**
- **About Oracle BI Time Repository Variables**
- **About User Authentication**
- **About the Incremental Deployment of the Oracle Tax Analytics Repository**
- **Configuring the Database and Connection Pools**
- **Deploying the Oracle Tax Analytics RPD and Catalog**
- **Configuring Oracle BI Suite Enterprise Edition**

> **Note:** When you access the OracleTaxAnalytics.rpd repository in the Oracle BI Administration Tool, you must log on with the user name and password Administrator\SADMIN.

Setting Up Date-Specific Metrics

The time dimension in the Oracle BI Repository for Oracle Tax Analytics Warehouse is a standard or canonical time dimension that links to the most important time role in each star schema. The Physical table alias used as a canonical time dimension is W_DAY_D_Common.

If a fact table contains a distinct set of metrics that needs to be reported by different dates, the metadata is organized so that each metric is reported by its causal date.

For example, the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, and each of these metrics need to be reported by the corresponding date-Invoice Date, Fulfill Date, and Payment Date. Additional dates in a fact table that a metric could be queried by are known as Secondary dates. These are available to the end users inside a detailed presentation folder. The detailed presentation folder is typically called the Details folder.

In the following table, each of the metrics reflect the activity related to that event for the entire period, for example, Invoice Amount by Invoice Date, Fulfill Amount by Fulfill date, and Payment Amount by Payment Date.

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

To Implement Date-Specific Metrics

1. Using Oracle BI Administration Tool, open OracleTaxAnalytics.rpd.

   The OracleTaxAnalytics.rpd file is located in the following folder:

   (OBIEE_installed_dir)\instances\(instance)instance1\bifoundation\OracleBIServerComponent\coreapplication_obis1\

2. In the Physical layer, right-click on Oracle Data Warehouse, and create a new physical alias for the fact table.
3. Create Joins for the physical alias which are similar to the base fact table.

   The Join to the date dimension is changed to use the date role in question.

4. Create a new logical table source in the logical fact table that maps the metrics for the physical fact alias.

   The grain of the fact table is the same as the base fact table.

   > **Note:** You need to map each metric to one logical table source at the Detail Level.

Setting Up Additional Time Series Metrics for Oracle Tax Analytics Warehouse

The Oracle BI Repository provides a framework to add Period Ago metrics. The Oracle Tax Analytics Warehouse is configured with premapped period ago metrics; however, you can map other metrics by using the following procedure.

To set up additional time series metrics:

1. Using Oracle BI Administration Tool, open OracleTaxAnalytics.rpd.

   The OracleTaxAnalytics.rpd file is located in the following folder:

   (OBIEE_installed_dir)\instances\(instance)instance1\bifoundation\OracleBIServerComponent\coreapplication_obis1\repository

2. In the Physical layer, right-click on Oracle Data Warehouse, and create a new Period Ago physical alias table.
3. In the Physical layer, create additional tables for each Period Ago alias required.

   For example, Quarter Ago, Year Ago, and so on.

   These aliases need to have the same joins as the base fact table, except for the date join, which you can change in the next step. Setting up this alias is easier to accomplish by copying the base table.
4. Change the join to the date dimension (W_DAY_D) to use the appropriate Period Ago Key.
5. Map the Period Ago metrics in the logical table using the new fact alias by creating a new logical table source under the fact table.
6. Set the content pane levels for the period ago logical table source, to specify the level of the source data.

   These settings are the same as the base fact table.
7. Save and close the OracleTaxAnalytics.rpd file.

Setting Up Additional Dimension Tables for the Oracle Tax Analytics Warehouse

The Oracle Tax Analytics Warehouse is preconfigured to map dimension tables required for analysis. The Physical layer in the Oracle BI Repository provides several other dimension table keys that can be used for certain specific analysis. If you need to set up any of the additional dimensions tables to the Physical layer, perform the following procedure.

To set up additional dimension tables:

1. Validate that the dimension table key is resolved appropriately for the data source that you are using.
2. Using Oracle BI Administration Tool, open OracleTaxAnalytics.rpd.

   The OracleTaxAnalytics.rpd file is located in the following folder:

   (OBIEE_installed_dir)\instances\(instance)instance1\bifoundation\OracleBIServerComponent\coreapplication_obis1\repository
3. Add a dimension table alias in the Physical layer.
4. Join the dimension table alias to the fact table alias using the appropriate keys.
5. Save and close the OracleTaxAnalytics.rpd file.

About the Period Ago Keys for Oracle Tax Analytics Warehouse

The Period Ago Key fields are used to set up the time series metrics like Year Ago, Quarter Ago, and so on. The Period Ago Key fields represent metrics for a prior period, for example, Quarter Ago Revenue, Year Ago Revenue, and so on. The Oracle Tax Analytics Warehouse is preconfigured with a set of fields in the W_DAY_D table. These fields are:

- MONTH_AGO_WID
- QUARTER_AGO_WID
- TRIMESTER_AGO_WID
- WEEK_AGO_WID
- YEAR_AGO_WID

These fields are used in joins to Oracle Tax Analytics Warehouse fact tables to achieve the period ago metrics. The joins in Oracle Tax Analytics Warehouse use the Period Ago fields in the W_DAY_D table.

About Oracle BI Time Repository Variables

The Oracle BI Repository is preconfigured with variables that are used for both reporting and internal usage.

The following table lists some example Oracle BI repository date variables and their descriptions. For a full list of variables, in Oracle BI Administration Tool, choose **Manage**, then **Variables**, to display the Variable Manager, and refer to the Description fields for a brief description.

Variable Name	Description
CURRENT_MONTH	Returns the value of Current Month in the YYYY/MM format.
RUN_DATE	Returns the value of current date in the Mon dd, yyyy format

About User Authentication

You need to configure the user authentication of the Oracle Tax Analytics Warehouse, depending on your requirements. The Oracle Tax Analytics Warehouse supports various authentication modes, for example, repository authentication, database authentication, and LDAP.

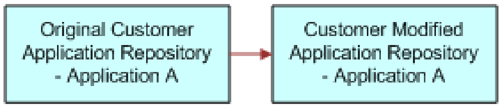
For more information about configuring user authentication, see Oracle Business Intelligence Server Administration Guide.

About the Incremental Deployment of the Oracle Tax Analytics Repository

Oracle Tax Analytics consists of various application families such as Tax Accounting Analytics, Tax Forms Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new applications.

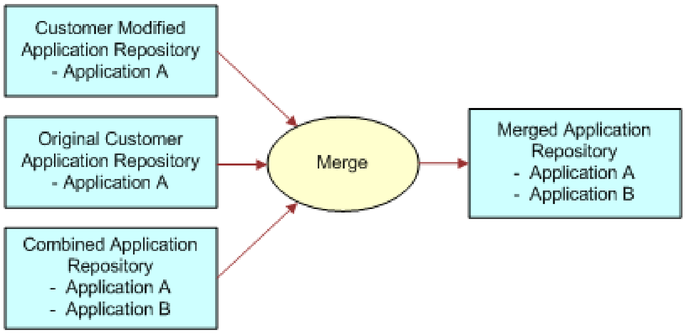
This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

The figure below shows a single Oracle Tax Analytics environment. During installation, you will be asked to specify the application modules you have licensed. The installer will extract the metadata corresponding to this module into one repository file. You can then modify the Oracle BI Repository to suit your business needs.


Original Customer Application Repository - Application A
Customer Modified Application Repository - Application A

When you purchase another Tax Analytics, you need to extract new metadata for all the modules that you have licensed. Use the merge utility in Oracle BI Administration Tool to perform a three-way merge of the original repository, the modified repository, and the combined repository. For more information on merging repositories, see Oracle Business Intelligence Server Administration Guide.

The merged repository preserves your modifications from the original Oracle BI Repository and appends the information with the new Oracle BI Repository, as shown in the figure below.


Customer Modified Application Repository - Application A
Original Customer Application Repository - Application A
Combined Application Repository - Application A - Application B
Merge
Merged Application Repository - Application A - Application B

You can repeat this merging procedure to add more Oracle Tax Analytics to the Oracle BI Repository.

Configuring the Database and Connection Pools

To configure the database and connection pools:

1. Open the RPD in Offline Mode.
2. Navigate to **Manage, Variables, Repository, Static.**
3. Edit the OLAP_DSN variable.
4. In the Default Initializer field of the variable dialog box, enter the name of the target data warehouse database instance.
5. Click **OK.**
6. Edit the OLAP_USER variable.
7. In the Default Initializer field of the variable dialog box, enter the name of the database schema for the target data warehouse.
8. Click **OK.**
9. Open the **Oracle Data Warehouse** database in the physical layer of the RPD.
10. Edit the **Oracle Data Warehouse Connection Pool**.

    On the General tab, enter the password for OLAP_USER.
11. Edit the connection pool **Oracle Data Warehouse Repository Init Blocks Connection Pool**.

    On the General tab, enter the password for OLAP_USER.
12. Edit the connection pool **AnalyticsWriteback**.

    On the General tab, enter the password for OLAP_USER.

Deploying the Oracle Tax Analytics RPD and Catalog

To deploy the Tax Analtyics RPD and catalog:

1. Place the Tax Analytics catalog in the following directory: \InstallHome\instances\instance1\bifoundation\OracleBIPresentationServicesComponent \coreapplication_obips1\catalog
2. Place the OracleTaxAnalytics.rpd file in the following directory: $ORACLE_INSTANCE/bifoundation/ OracleBIServerComponent / $COMPONENT_NAME/ repository/:

For example:
\InstallHome\instances\instance1\bifoundation\OracleBIServerComponent\coreapplication_obips1\repository

3. Copy all the content within the WritebackXMLs folder to the following location in your BIEE install:

   \InstallHome\instances\instance1\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1\analyticsRes

   This folder contains all the custom files (writeback templates) that are consumed by Tax Analytics dashboards and reports.

   > **Note:** If some of the subfolders (for example, customMessages) already exist within the analyticsRes folder on your system, then copy only the XML files from the WritebackXMLs\customMessages folder and place in the install path of the customMessages folder.

Configuring Oracle BI Suite Enterprise Edition

To set the default RPD and Webcat paths using Enterprise Manager, log in to the Enterprise Manager administration screen using the url http://localhost:7001/em. (Use the host name and port number associated with your installation.)

Setting the Default RPD

To set the default RPD:

1. Use the login you created in the Oracle BIEE installation to log in to the Enterprise Manager console.
2. Expand the Business Intelligence node and choose Coreapplication.
3. Select the Deployment tab, and then the Repository tab.
4. Select **Lock and Edit Configuration** to change the default settings.
5. Under Upload BI Server Repository, click on **Browse**, and then locate the OracleTaxAnalytics.rpd file.
6. In the Password field, enter Obidemo123 (this is the offline password for OracleTaxAnalytics.rpd).
7. Click **Apply**.
8. Click **Activate Changes.**
9. After activation completes, you should see a new RPD listed. For example, TaxAnalytics_BIXXXX.

   If you still see the old RPD name, close this screen and restart the process.
10. Click **Restart** to apply the changes.

    This takes you to the page where you can restart services.
11. Under Manage System, click on either **Restart** or **Start**.
12. Click **Yes** in the confirmation dialog box to confirm the change.

    A message confirms that the service has successfully restarted.

Setting the Webcat Path

Your installation needs to point to the Tax Analytics catalog. In Enterprise Manager, you need to update the path to the catalog. For example:

$ORA-CLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/
$COMPONENT_NAME/catalog/XXXXXX.

To set the Webcat Path:

1. Use the login you created in the Oracle BIEE installation to log in to the Enterprise Manager console.
2. Click on **Lock and Edit Configuration** to change the default settings to the path to your catalog. For example:

   $ORA-CLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/
   $COMPONENT_NAME/catalog/ TaxAnalytics.
3. Click on **Apply**.
4. Click **Activate Changes.**
5. Click **Restart** to apply the changes.

   This takes you to the page where you can restart services.
6. Click **Restart**.
7. Click **Yes** in the confirmation dialog box to confirm the change.

   > **Note:** If the services don't start up using Enterprise Manager, you can try to restart them using the OPMN process management tool.