Oracle® Automotive Trading Partner

Toolkit

Technical Reference Manual

Release 11*i*

KEEP UNDER LOCK & KEY

CAUTION!

This document contains proprietary, confidential information that is the exclusive property of Oracle Corporation.

See back cover for important information.

Oracle® Automotive Trading Partner Toolkit Technical Reference Manual

RELEASE 11*i*

March 2000

Oracle® Automotive Trading Partner Toolkit Technical Reference Manual
Release 11*i*

To order this book, ask for Part No. A83735_01

Major Contributors: Subir Hira, Noela Nakos, Lauren Scott, Manish Srivastava, Julie Thai

Contributors: Zoe Mitchell, Elizabeth Looney, Richard Sears, David Reitan

This Technical Reference Manual (TRM) in any form, software or printed matter, contains proprietary information of Oracle Corporation; it is provided under an Oracle Corporation agreement containing restrictions on use and disclosure and is also protected by copyright, patent, and other intellectual property law. Restrictions applicable to this TRM include, but are not limited to: (a) exercising either the same degree of care to safeguard the confidentiality of this TRM as you exercise to safeguard the confidentiality of your own most important Confidential Information or a reasonable degree of care, whichever is greater; (b) maintaining agreements with your employees and agents that protect the Confidential Information of third parties such as Oracle Corporation and instructing such employees and agents of these requirements for this TRM; (c) restricting disclosure of this TRM to those of your employees who have a "need to know" consistent with the purposes for which this TRM was disclosed to you; (d) maintaining this TRM at all times on your premises; (e) not removing or destroying any proprietary or confidential legends or markings placed upon this TRM in whatever form, software or printed matter; and (f) not reproducing or transmitting this TRM in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation. You should not use this TRM in any form, software or printed matter, to create software that performs the same or similar functions as any Oracle Corporation products.

The information in this TRM is subject to change without notice. If you find any problems in the TRM in any form, software or printed matter, please report them to us in writing. Oracle Corporation does not warrant that this TRM is error–free. This TRM is provided to customer "as–is" with no warranty of any kind. This TRM does not constitute Documentation as that term is defined in Oracle's agreements.

Reverse engineering of the Programs (software and documentation) associated with this TRM are prohibited. The Programs associated with this TRM are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail–safe, back–up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

Restricted Rights Legend
This TRM and the Programs associated with this TRM delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the TRM and the Programs associated with this TRM shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, this TRM and the Programs associated with this TRM delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the TRM and the Programs associated with this TRM shall be subject to the restrictions in FAR 52.227–14, Rights in Data — General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle is a registered trademark, and CASE*Exchange, Enabling the Information Age, Hyper*SQL, NLS*Workbench, Oracle7, Oracle8, Oracle 8*i*, Oracle Access, Oracle Application Object Library, Oracle Discoverer, Oracle Financials, Oracle Quality, Oracle Web Customers, Oracle Web Employees, Oracle Work in Process, Oracle Workflow, PL/SQL, Pro*Ada, Pro*C, Pro*COBOL, Pro*FORTRAN, Pro*Pascal, Pro*PL/I, SmartClient, SQL*Connect, SQL*Forms, SQL*Loader, SQL*Menu, SQL*Net, SQL*Plus, and SQL*Report are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

CAUTION

This Technical Reference Manual in any form — software or printed matter — contains proprietary, confidential information that is the exclusive property of Oracle Corporation. If you do not have a valid contract with Oracle for the use of this Technical Reference Manual or have not signed a non–disclosure agreement with Oracle covering this Technical Reference Manual, then you received this document in an unauthorized manner and are not legally entitled to possess or read it.

Use, duplication, and disclosure are subject to restrictions stated in your contract with Oracle Corporation.

Contents

Oracle Proprietary, Confidential Information—Use Restricted by Contract

CHAPTER

1 Introduction

The *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* provides the information you need to understand the underlying structure of Oracle Automotive Trading Partner Toolkit. After reading this manual, you should be able to convert your existing applications data, integrate your existing applications with your Oracle Automotive Trading Partner Toolkit application, and write custom reports for your Oracle Automotive Trading Partner Toolkit application, as well as read data that you need to perform other tasks.

This chapter introduces you to the *Oracle Automotive Trading Partner Toolkit Technical Reference Manual*, and explains how to use it.

Introduction

At Oracle, we design and build applications using Oracle Designer, our systems design technology that provides a complete environment to support developers through all stages of a systems life cycle. Because we use a repository–based design toolset, all the information regarding the underlying structure and processing of our applications is available to us online. Using Oracle Designer, we can present this information to you in the form of a technical reference manual.

This *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* contains detailed, up–to–date information about the underlying structure of Oracle Automotive Trading Partner Toolkit. As we design and build new releases of Oracle Automotive Trading Partner Toolkit applications, we update our Oracle Designer repository to reflect our enhancements. As a result, we can always provide you with an *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* that contains the latest technical information as of the publication date. Note that after the publication date we may have added new indexes to Oracle Automotive Trading Partner Toolkit to improve performance.

About this Manual

This manual describes the Oracle Applications Release 11i data model, as used by Oracle Automotive Trading Partner Toolkit; it discusses the database we include with a fresh install of Oracle Applications Release 11i. If you have not yet upgraded to Release 11i, your database may differ from the database we document in this book.

If you have upgraded from a previous release, you might find it helpful to use this manual with the appropriate *Oracle Applications Product Update Notes* manual. The product update notes list database changes and seed data changes in Oracle Automotive Trading Partner Toolkit between releases. The *Oracle Applications Product Update Notes Release 11* manual describes the changes between Release 10.7 and Release 11, and the *Oracle Applications Product Update Notes Release 11i* manual describes the changes between Release 11 and Release 11i.

You can contact your Oracle representative to confirm that you have the latest technical information for Oracle Automotive Trading Partner Toolkit. You can also use Oracle*MetaLink* which is accessible through Oracle's Support Web Center (*http://www.oracle.com/support/elec_sup*).

Finding the Latest Information

The *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* contains the latest information as of the publication date. For the latest information we encourage you to use Oracle*MetaLink* which is accessible through Oracle's Support Web Center (*http://www.oracle.com/support/elec_sup*).

Audience

The *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* provides useful guidance and assistance to:

- Technical End Users
- Consultants
- Systems Analysts
- System Administrators
- Other MIS professionals

This manual assumes that you have a basic understanding of structured analysis and design, and of relational databases. It also assumes that you are familiar with Oracle Application Object Library and your Oracle Automotive Trading Partner Toolkit application. If you are not familiar with the above products, we suggest that you attend one or more of the training classes available through Oracle Education (see: Other Information Sources: page 1 – 8).

How This Manual is Organized

This manual contains two major sections, High–Level Design and Detailed Design.

High–Level Design

This section, Chapter 2, contains database diagrams and lists each database table and view that Oracle Automotive Trading Partner Toolkit uses. This chapter also has a list of modules.

Detailed Design

This section, Chapter 3, contains a detailed description of the Oracle Automotive Trading Partner Toolkit database design, including information about each database table and view you might need for your custom reporting or other data requirements.

How to Use This Manual

The *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* is a single, centralized source for all the information you need to know about the underlying structure and processing of your Oracle Automotive Trading Partner Toolkit application. For example, you can use this manual when you need to:

- Convert existing application data
- Integrate your Oracle Automotive Trading Partner Toolkit application with your other applications systems
- Write custom reports
- Define alerts against Oracle Applications tables
- Configure your Oracle Self–Service Web Applications
- Create views for decision support queries using query tools
- Create business views for Oracle Discoverer

You need not read this manual cover to cover. Use the table of contents and index to quickly locate the information you need.

How Not To Use This Manual

Do not use this manual to plan modifications

You should not use this manual to plan modifications to your Oracle Automotive Trading Partner Toolkit application. Modifying Oracle Automotive Trading Partner Toolkit limits your ability to upgrade to future releases of your Oracle Automotive Trading Partner Toolkit application. In addition, it interferes with our ability to give you the high–quality support you deserve.

We have constructed your Oracle Automotive Trading Partner Toolkit application so that you can customize it to fit your needs without programming, and you can integrate it with your existing applications through interface tables. However, should you require program modifications, you should contact our support team (see: Other Information Sources: page 1 – 8). They can put you in touch with Oracle Services, the professional consulting organization of Oracle. Their team of experienced applications professionals can make the modifications you need while ensuring upward compatibility with future product releases.

Do not write data into non–interface tables

Oracle reserves the right to change the structure of Oracle Applications tables, and to change the meaning of, add, or delete lookup codes and data in future releases. Do not write data directly into or change data in non–interface tables using SQL*Plus or other programming tools because you risk corrupting your database and interfering with our ability to support you.

Moreover, this version of the *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* does not contain complete information about the dependencies between Oracle Automotive Trading Partner Toolkit applications tables. Therefore, you should write data into only those tables we identify as interface tables. If you write data into other non–interface tables, you risk violating your data integrity since you might not fulfill all the data dependencies in your Oracle Automotive Trading Partner Toolkit application.

You are responsible for the support and upgrade of the logic within the procedures that you write, which may be affected by changes between releases of Oracle Applications.

Do not rely on upward compatibility of the data model

Oracle reserves the right to change the structure of Oracle Automotive Trading Partner Toolkit applications tables, and to change the meaning of, add, or delete lookup codes and other data in future releases. We do not guarantee the upward compatibility of the Oracle Automotive Trading Partner Toolkit applications data model. For example, if you write a report that identifies concurrent requests that end in Error status by selecting directly from Oracle Application Object Library tables, we do not guarantee that your report will work properly after an upgrade.

About Oracle Application Object Library

The *Oracle Automotive Trading Partner Toolkit Technical Reference Manual* may contain references to tables that belong to Oracle Application Object Library. Oracle Application Object Library is a collection of pre–built application components and facilities for building Oracle Applications and extensions to Oracle Applications. Oracle Application Coding Standards use the Oracle Application Object Library and contains shared components including but not limited to — forms, subroutines, concurrent programs and reports, database tables and objects, messages, menus, responsibilities, flexfield definitions and online help.

☞ **Attention:** Oracle does not support *any* customization of Oracle Application Object Library tables or modules, not even by Oracle consultants. (Oracle Application Object Library tables generally have names beginning with FND_%.)

Accordingly, this manual does not contain detailed information about most Oracle Application Object Library tables used by your Oracle Automotive Trading Partner Toolkit application.

A Few Words About Terminology

The following list provides you with definitions for terms that we use throughout this manual:

Relationship

A relationship describes any significant way in which two tables may be associated. For example, rows in the Journal Headers table may have a one–to–many relationship with rows in the Journal Lines table.

Database Diagram

A database diagram is a graphic representation of application tables and the relationships between them.

Summary Database Diagram

A summary database diagram shows the most important application tables and the relationships between them. It omits tables and relationships that contribute little to the understanding of the application data model. Typically, a summary database diagram shows tables that contain key reference and transaction data.

Module

A module is a program or procedure that implements one or more business functions, or parts of a business function, within an application. Modules include forms, concurrent programs and reports, and subroutines.

Application Building Block

An application building block is a set of tables and modules (forms, reports, and concurrent programs) that implement closely–related database objects and their associated processing. Said another way, an application building block is a logical unit of an application.

QuickCodes

QuickCodes let you define general purpose, static lists of values for window fields. QuickCodes allow you to base your program logic on lookup codes while displaying user–friendly names in a list of values window. QuickCodes simplify name and language changes by letting you change the names your end users see, while the codes in your underlying programs remain the same.

Form

A form is a module comprised of closely related windows that are used together to perform a task. For example, the Enter Journals form in Oracle General Ledger includes the Enter Journals window, the Batch window, and the More Actions window among others. The Enter Journals window is the main window, and from it, you can use buttons to navigate to other windows in the form. The form name usually corresponds to the main window in the form, and is frequently a window you open directly from the Navigator.

Other Information Sources

There are additional information sources, including other documentation, training and support services, that you can use to increase your knowledge and understanding of Oracle Designer, Oracle Application Object Library, and your Oracle Automotive Trading Partner Toolkit application. We want to make these products easy for you and your staff to understand and use.

Oracle Designer Online Documentation

The online help for Oracle Designer describes how you can use Oracle Designer for your development needs.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards*. It also provides information to help you build your custom Developer forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards

This manual contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built using Oracle Forms 6.

Oracle Automotive Trading Partner Toolkit applications User Guide

Your user guide provides you with all the information you need to use your Release 11i Oracle Automotive Trading Partner Toolkit application. Each user guide is organized for fast, easy access to detailed information in a function– and task–oriented organization.

Oracle Self–Service Web Applications Online Documentation

This documentation describes how Oracle Self–Service Web Applications enable companies to provide a self–service and secure Web interface for employees, customers, and suppliers. Employees can change their personal status, submit expense reports, or request supplies. Customers can check on their orders, and suppliers can share production schedules with their trading partners. This documentation is available in HTML only.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Oracle Automotive Trading Partner Toolkit implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle Workflow Guide

This manual explains how to define new workflow business processes as well as customize existing Oracle Applications–embedded workflow processes. You also use this guide to complete the setup steps necessary for any Oracle Applications product that includes workflow–enabled processes.

Oracle Alert User Guide

This manual explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to report and maintain accounting records in more than one currency, use this manual before implementing Oracle Automotive Trading Partner Toolkit. This manual details additional steps and setup considerations for implementing Oracle Automotive Trading Partner Toolkit with this feature.

Multiple Organizations in Oracle Applications

If you use the Oracle Applications Multiple Organization Support feature to use multiple sets of books for one Oracle Automotive Trading Partner Toolkit installation, use this guide to learn about setting up and using Oracle Automotive Trading Partner Toolkit with this feature. This book describes the Oracle Applications organization model, which defines business units and the relationships between them in an arbitrarily complex enterprise. Functional and technical overviews of multiple organizations are presented, as well as information about how to set up and implement this feature set in the relevant Oracle Applications products.

Oracle Manufacturing, Distribution, Sales and Service Open Interfaces Manual

This manual contains up–to–date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes open interfaces found in Oracle Manufacturing.

Oracle Applications Messages Manual

The Oracle Applications Messages Manual contains the text of numbered error messages in Oracle Applications. (Oracle Applications messages begin with the prefix ”APP–”.) It also provides information on the actions you take if you get a message. Note: This manual is available only in HTML format.

Installation and System Administration

Oracle Applications Installation Release Notes

This manual contains a road map to the components of the release, including instructions about where to access the Release 11i documentation set.

Oracle Applications Concepts

Designed to be the first book the user reads to prepare for an installation of Oracle Applications. It explains the technology stack, architecture, features and terminology for Oracle Applications Release 11i. This book also introduces the concepts behind and major uses of Applications–wide features such as MRC, BIS, languages and character sets (NLS, MLS), BIS, Self–Service Web Applications and so on.

Installing Oracle Applications

Describes the One–Hour Install process, the method by which Release 11i will be installed. This manual includes all how–to steps, screen shots and information about Applications–wide post–install tasks.

Using the AD Utilities

This manual contains how–to steps, screen shots and other information required to run the various AD utilities such as AutoInstall, AutoPatch, AD Administration, AD Controller, Relink and so on. It also contains information about when and why you should use these utilities.

Upgrading Oracle Applications

This manual contains all the product specific pre– and post–upgrade steps that are required to upgrade products from Release 10.7 (NCA, SC and character–mode) or Release 11 of Oracle Applications. This manual also contains an overview chapter that describes all the tasks necessary to prepare and complete a upgrade of Oracle Applications.

Oracle Applications System Administrator’s Guide

This manual provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and manage concurrent processing.

Oracle Applications Product Update Notes

This book contains a summary of each new feature we added since Release 11, as well as information about database changes and seed data changes that may affect your operations or any custom reports you have written. If you are upgrading from Release 10.7 you also need to read *Oracle Applications Product Update Notes Release 11*.

Oracle Self–Service Web Applications Implementation Manual

This manual describes the setup steps for Oracle Self–Service Web Applications and the Web Applications Dictionary.

Oracle Applications Implementation Wizard User Guide

If you are implementing more than one Oracle product, you can use the Oracle Applications Implementation Wizard to coordinate your setup activities. This guide describes how to use the wizard.

Other Information

Training

Oracle Education offers a complete set of training courses to help you and your staff master Oracle Applications. We can help you develop a training plan that provides thorough training for both your project team and your end users. We will work with you to organize courses appropriate to your job or area of responsibility.

Training professionals can show you how to plan your training throughout the implementation process so that the right amount of information is delivered to key people when they need it the most. You can attend courses at any one of our many Educational Centers, or you can arrange for our trainers to teach at your facility. In addition, we can tailor standard courses or develop custom courses to meet your needs.

Support

From on–site support to central support, our team of experienced professionals provides the help and information you need to keep your Oracle Automotive Trading Partner Toolkit application working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle server, and your hardware and software environment.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 75 software modules for financial management, supply chain management, manufacturing, project systems, human resources, and sales and service management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers, and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

Thank You

Thanks for using Oracle Automotive Trading Partner Toolkit applications and this technical reference manual!

We appreciate your comments and feedback. At the back of this manual is a Reader's Comment Form that you can use to explain what you like or dislike about your Oracle Automotive Trading Partner Toolkit application or this technical reference manual. Mail your comments to the following address or call us directly at (650) 506–7000.

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, California 94065
U.S.A.

Or, send electronic mail to **appsdoc@us.oracle.com**.

CHAPTER

2

High–Level Design

This chapter presents a high–level design for Oracle Automotive Trading Partner Toolkit that satisfies the business needs we specify during Strategy and Analysis. It contains database diagrams for Oracle Automotive Trading Partner Toolkit application building blocks, lists of database tables and views, and a list of modules.

Overview of High–Level Design

During High–Level Design, we define the application components (tables, views, and modules) we need to build our application. We specify what application components should do without specifying the details of *how* they should do it.

You can refer to this High–Level Design chapter to quickly acquaint yourself with the tables, views, and modules that comprise Oracle Automotive Trading Partner Toolkit applications. And, you can prepare yourself to understand the detailed design and implementation of Oracle Automotive Trading Partner Toolkit.

Summary Database Diagram

The Summary Database Diagram section graphically represents the most important application tables and the relationships between them. It omits tables and relationships that contribute little to the understanding of the application data model. Typically, a summary database diagram shows tables that contain key reference and transaction data.

We prepare a summary database diagram to describe, at a conceptual level, the key information on which our business depends. Later, we refine this summary database diagram, breaking it into multiple database diagrams (generally, one per application building block) to represent all the tables and relationships we need to implement our application in the database.

Review the Summary Database Diagram section to see at a glance the major tables and relationships on which your Oracle Automotive Trading Partner Toolkit application depends.

Database Diagrams

The Database Diagrams section graphically represents all Oracle Automotive Trading Partner Toolkit applications tables and the relationships between them, organized by building block.

Use this section to quickly learn what tables each Oracle Automotive Trading Partner Toolkit application building block uses, and how those tables interrelate. Then, you can refer to the Table and View Definitions sections of Chapter 3 for more detailed information about each of those tables.

Table Lists

The Table List sections list the Oracle Automotive Trading Partner Toolkit applications tables. Because a product might not include at least one table for each type, this Oracle Automotive Technical Reference Manual might not include each of the following sections.

Public Tables

Use the Public Table List section to quickly identify the tables you are most interested in. Then, you can refer to the Table and View Definitions sections of Chapter 3 for more detailed information about those tables.

In addition, this manual may contain full documentation for one or more of the following Application Object Library tables: FND_DUAL, FND_CURRENCIES, and FND_COMMON_LOOKUPS.

Internal Tables

This section includes a list of private, internal tables used by Oracle Automotive Trading Partner Toolkit; we do not provide additional documentation for these tables.

View Lists

The View List sections list the Oracle Automotive Trading Partner Toolkit views, with one section for each type of view. Because a product might not include at least one view for each type, this Oracle Automotive Technical Reference Manual might not include each of the following sections.

Use this section to quickly identify the views you are most interested in. Then, you can refer to the Table and View Definitions sections of Chapter 3 for more detailed information about those views.

Public Views

This section lists views that may be useful for your custom reporting or other data requirements. The list includes a description of the view, and the page in Chapter 3 that gives detailed information about the public view.

Web Views

This section lists views that you may need to configure your Self–Service Web applications. The list includes a description of the view, and the page in Chapter 3 that gives detailed information about the web view.

Internal Views

This section includes each private, internal view that Oracle Automotive Trading Partner Toolkit uses.

Multiple Reporting Currency Views

This list includes views that were created to support the Multiple Reporting Currencies feature.

Module List

The Module List section briefly describes each of the Oracle Automotive Trading Partner Toolkit applications modules. This section lists forms, reports, and concurrent programs.

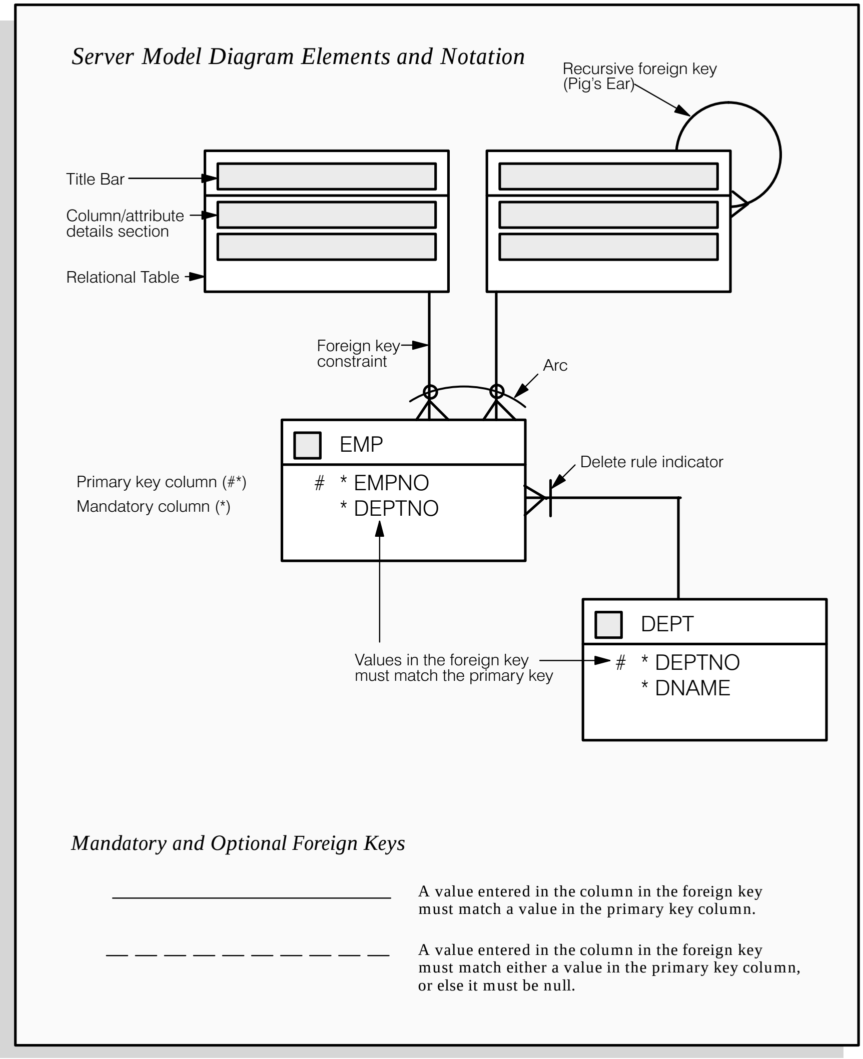
A form is a module comprised of closely related windows that are used together to perform a task. For example, the Enter Journals form in Oracle General Ledger includes the Enter Journals window, the Batch window, and the More Actions window. The Enter Journals window is the main window, and from it, you can use buttons to navigate to other windows in the form. The form name usually corresponds to the main window in the form, and is frequently a window you can open directly from the Navigator.

The Reports and Concurrent Programs lists include processes you can submit from the Submit Requests window or other windows, as well as processes that are submitted automatically by Oracle Automotive Trading Partner Toolkit. Use your user’s guide to learn more about reports and concurrent processes.

Database Diagramming Conventions

We use the following notational conventions in our database diagrams:

Figure 2 – 1 Database Diagram Conventions



Server Model Diagram Elements and Notation
Recursive foreign key (Pig's Ear)
Title Bar
Column/attribute details section
Relational Table
Foreign key constraint
Arc
EMP
Delete rule indicator
Primary key column (#*)
Mandatory column (*)
# * EMPNO
* DEPTNO
DEPT
Values in the foreign key must match the primary key
# * DEPTNO
* DNAME

Mandatory and Optional Foreign Keys

A value entered in the column in the foreign key must match a value in the primary key column.

A value entered in the column in the foreign key must match either a value in the primary key column, or else it must be null.

Tables – are the basic unit of storage in the database. A hand symbol preceding the title in the table's title bar indicates that the table is not owned by this application but shared with another.

Foreign key constraint – is a type of referential integrity constraint for checking the integrity of data entered in a specific column or set of columns. This specified column or set of columns is known as the foreign key.

Delete rule indicator – determines the action to be taken when an attempt is made to delete a related row in a join table. A line through the foreign key constraint, as shown on the above diagram, indicates that this action is restricted.

Arcs – specify that, for any given row in a table, a value must be entered in one of the arc columns. The remaining columns within the arc must be null.

Oracle Automotive Trading Partner Toolkit Summary Database Diagram

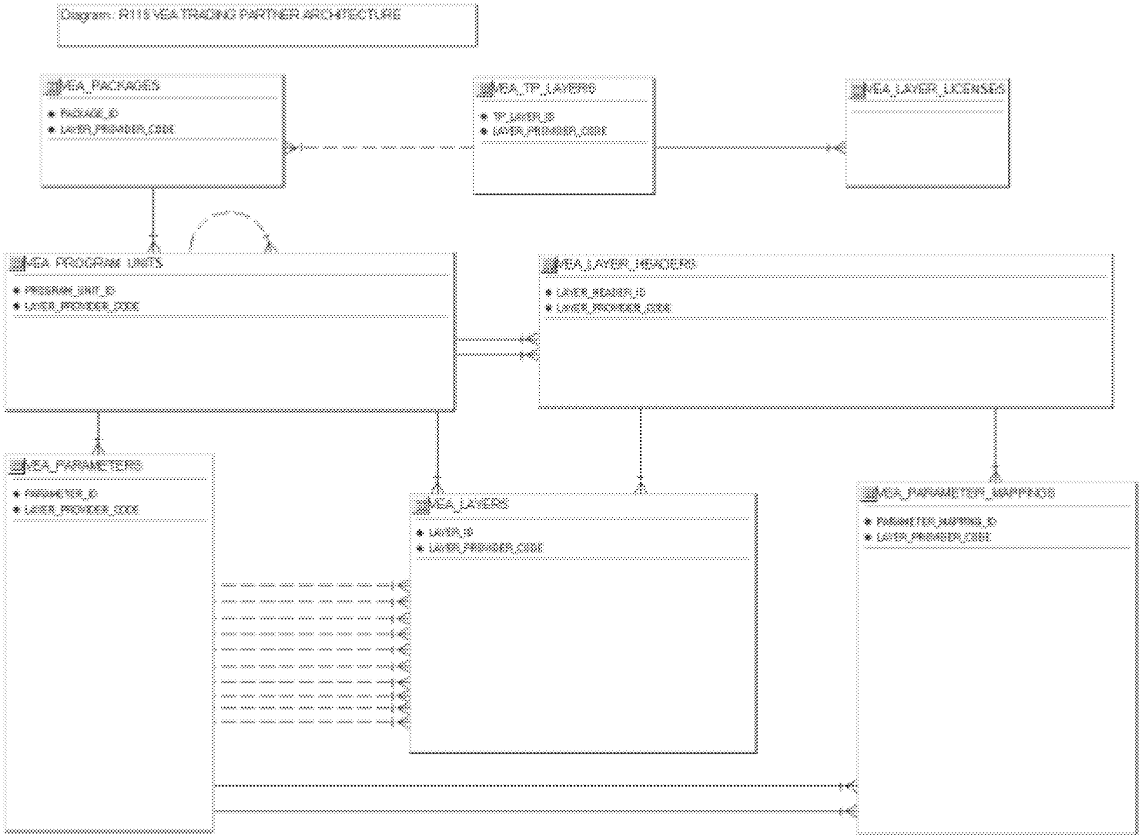


Diagram: R11i VEA TRADING PARTNER ARCHITECTURE
VEA_PACKAGES
* PACKAGE_ID
* LAYER_PROVIDER_CODE
VEA_TP_LAYERS
* TP_LAYER_ID
* LAYER_PROVIDER_CODE
VEA_LAYER_LICENSES
VEA_PROGRAM_UNITS
* PROGRAM_UNIT_ID
* LAYER_PROVIDER_CODE
VEA_LAYER_HEADERS
* LAYER_HEADER_ID
* LAYER_PROVIDER_CODE
VEA_PARAMETERS
* PARAMETER_ID
* LAYER_PROVIDER_CODE
VEA_LAYERS
* LAYER_ID
* LAYER_PROVIDER_CODE
VEA_PARAMETER_MAPPINGS
* PARAMETER_MAPPING_ID
* LAYER_PROVIDER_CODE

Public Table List

This section lists each public database table that Oracle Automotive uses and provides a brief description of each of those tables. The page reference is to the table description in Chapter 3.

Note that ”public” tables are not necessarily intended for write access by custom code; Oracle Corporation supports write access using only standard Oracle Applications forms, reports, and programs, or any SQL write access to tables explicitly documented as API tables. For more information, see the How Not To Use This Manual section of this book’s Introduction.

Oracle Automotive uses the following Public tables:

Table Name	Description
VEA_EXCEPTIONS	Not currently used (See page 3 – 8)
VEA_LAYERS	Branches for base layer program unit (See page 3 – 9)
VEA_LAYER_HEADERS	Header information about links to base layer program units (See page 3 – 12)
VEA_LAYER_LICENSES	Customer licenses for trading partner layers (See page 3 – 13)
VEA_PACKAGES	Packages registered with the trading partner architecture repository (See page 3 – 14)
VEA_PARAMETERS	Signatures of program units registered with the trading partner architecture repository (See page 3 – 16)
VEA_PARAMETER_MAPPINGS	Mapping of trading partner selector program unit parameters with base layer program unit parameters (See page 3 – 18)
VEA_PROGRAM_UNITS	Program units registered with the trading partner architecture repository (See page 3 – 20)
VEA_RELEASE_DETAILS	Not currently used (See page 3 – 22)
VEA_TP_LAYERS	Trading partner layers (See page 3 – 23)

Internal View List

This section lists each private, internal view that Oracle Automotive Trading Partner Toolkit uses.

⚠ **Warning:** Oracle Corporation does not support access to Oracle Applications data using these views, except from standard Oracle Applications forms, reports, and programs.

Oracle Automotive Trading Partner Toolkit uses the following internal views:

- VEA_LAYERS_V

Module List

This section lists each form, report and concurrent program comprising Oracle Automotive.

Forms

VEAADTPL	Activate / Deactivate Layers
VEACRLAL	Create Layer Licenses
VEALAYER	Layer Workbench

CHAPTER

3

Detailed Design

This chapter presents a detailed design for implementing Oracle Automotive Trading Partner Toolkit. It contains detailed definitions of tables and views that you may need to reference to write custom reports or use for other data extraction.

Overview of Detailed Design

During Detailed Design, we specify in detail how each applications component should work. We prepare detailed definitions of tables and views.

You can refer to this Detailed Design chapter to gain a detailed understanding of the underlying structure and processing of Oracle Automotive Trading Partner Toolkit that enables you to:

- Convert existing application data
- Integrate your Oracle Automotive Trading Partner Toolkit application with your other applications systems
- Write custom reports
- Define alerts against Oracle Applications tables
- Create views for decision support queries using query tools
- Configure your Oracle Self–Service Web Applications

Table and View Definitions

The Table and View Definitions section contains a detailed definition of Oracle Automotive Trading Partner Toolkit applications tables. For each table, it provides information about primary keys, foreign keys, QuickCodes, indexes, triggers, and sequences. It also gives you a detailed description of each column and its characteristics. In addition, it provides the SQL statement that defines each view. Review this section to get a detailed understanding of what tables your Oracle Automotive Trading Partner Toolkit application contains, and how it uses them to hold and access the information it needs.

Table and View Definitions

This section contains a detailed description of each Oracle Automotive Trading Partner Toolkit table and view that you may need to reference. For each table, it presents detailed information about:

- Primary keys
- Foreign keys
- Column descriptions
- Indexes
- Oracle sequences
- Triggers
- View derivations

Because Oracle does not support customization of Oracle Application Object Library tables, we do not provide you with detailed information about them. Consequently, this section does not document all the FND_% tables your Oracle Automotive Trading Partner Toolkit application uses.

The following sections appear in each table or view description:

Foreign Keys

To help you understand the relationships between tables, we list each foreign key contained in a table. For each foreign key in a table, we list the primary key table name (the table to which a foreign key refers), its corresponding primary key columns, and the foreign key columns that refer to those primary key columns.

When the primary key table has a composite primary key, we list each column of the composite key sequentially.

If a table contains two or more distinct foreign keys that refer to the same primary key table, we repeat the primary key table name and list each of the distinct foreign keys separately.

QuickCodes Columns

When a database column contains a QuickCodes value, which we implement using a foreign key to FND_LOOKUPS, MFG_LOOKUPS, or to some other lookup table, we list the QuickCodes type (lookup type) to which the QuickCodes value must belong and a complete list of QuickCodes values and meanings. Some QuickCodes can be defined by you in the application. These values are designated as User–defined.

Column Descriptions

We list the important characteristics of each column in a table or view. These characteristics include whether the column is part of the table's primary key, whether Oracle8i requires a value for this column, and the data type of the column. We also give you a brief description of how your Oracle Automotive Trading Partner Toolkit application uses the column.

When a column is part of a table's primary key, we append the notation (PK) to the name of that column.

To help you understand which columns your Oracle Automotive Trading Partner Toolkit application uses and which columns it does not use, we alert you to any unused column. When no module uses a database column, we show one of the following legends in the Description column:

Not currently used Your Oracle Automotive Trading Partner Toolkit application does not use this column, although the column might be used in a future release.

No longer used Your Oracle Automotive Trading Partner Toolkit application no longer uses this column. AutoInstall installs this column. Subsequent versions of your Oracle Automotive Trading Partner Toolkit application might not include this column.

No longer installed Your Oracle Automotive Trading Partner Toolkit application no longer uses this column. If you *upgraded* your software from an earlier version, you may still have this column, depending upon whether you chose to delete it during an upgrade process. If you *install* your Oracle Automotive Trading Partner Toolkit application, you do not have this column.

Standard Who Columns

Most Oracle Automotive Trading Partner Toolkit applications tables contain standard columns to support \ **R**ow **W**ho. When your program or SQL*Plus command selects a row from a table, use these columns to determine who last updated the row. If your program or SQL*Plus command updates or inserts a row in an interface table, you must populate each of the five standard Who columns:

LAST_UPDATE_DATE Date when a user last updated this row

LAST_UPDATED_BY User who last updated this row (foreign key to FND_USER.USER_ID)

CREATION_DATE Date when this row was created

CREATED_BY User who created this row (foreign key to FND_USER.USER_ID)

LAST_UPDATE_LOGIN Operating system login of user who last updated this row (foreign key to FND_LOGINS.LOGIN_ID). You should set this to NULL, or to 0 if NULL is not allowed

Since every table containing Who columns has several foreign keys to the tables FND_USER and FND_LOGINS, we do not include the foreign key columns LAST_UPDATED_BY, CREATED_BY, or LAST_UPDATE_LOGIN in a table’s list of foreign keys.

Additional Who Columns for Concurrent Programs

Some Oracle Automotive Trading Partner Toolkit applications tables also contain several additional Who columns to distinguish between changes a user makes with a form and changes a concurrent program makes. When a concurrent program updates or inserts a row in a table, the concurrent program populates the following additional Who columns:

REQUEST_ID Concurrent request ID of program that last updated this row (foreign key to FND_CONCURRENT_REQUESTS.REQUEST_ID)

PROGRAM_APPLICATION_ID Application ID of program that last updated this row (foreign key to FND_APPLICATION.APPLICATION_ID)

PROGRAM_ID Program ID of program that last updated this row (foreign key to FND_CONCUR-

	RENT_PROGRAM.CONCURRENT_PROGRAM_ID)
PROGRAM_UPDATE_DATE	Date when a program last updated this row

Since every table containing these additional Who columns has several foreign keys to the tables FND_CONCURRENT_REQUESTS, FND_APPLICATION, and FND_CONCURRENT_PROGRAM, we do not include the foreign key columns REQUEST_ID, PROGRAM_APPLICATION_ID, or PROGRAM_ID in a table's list of foreign keys.

Indexes

If an Oracle Automotive Trading Partner Toolkit applications table uses an Oracle8i index, we list the database columns that comprise that index, in sequential order.

> **Note:** The indexes we document in this manual correspond to unique keys we specified during product development and testing. In some cases, we may add additional indexes during the porting process to fine–tune performance on specific platforms; therefore, there may be minor differences between the indexes documented in this book and the indexes for production versions of Oracle Automotive Trading Partner Toolkit.

Sequences

Your Oracle Automotive Trading Partner Toolkit application uses Oracle8i sequence generators to generate unique integers. If any table column gets its value from an Oracle8i sequence generator, we list the name of the corresponding sequence generator and the name of the column that stores the unique integer.

Database Triggers

If a table has one or more active database triggers, we provide a brief explanation of each database trigger and when it fires.

View Derivation

For each Oracle Automotive Trading Partner Toolkit applications view you may need to reference, we include important elements from the SQL statement that defines or creates a view. By studying this view definition, you can understand exactly how a view derives its contents.

VEA_EXCEPTIONS

Not currently used.

Column Descriptions

Name	Null?	Type	Description
EXCEPTION_ID (PK)	NOT NULL	NUMBER	Exception identifier
RELEASE_ID	NOT NULL	NUMBER	Release identifier
LAYER_PROVIDER_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier
MESSAGE_NAME	NOT NULL	VARCHAR2(30)	Name of the message
EXCEPTION_LEVEL	NOT NULL	VARCHAR2(1)	Indicates severity of the message
MESSAGE_TEXT	NOT NULL	VARCHAR2(2000)	Text of the message
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DESCRIPTION	NULL	VARCHAR2(1000)	Description of the message

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_EXCEPTIONS_U1	UNIQUE	1	EXCEPTION_ID

Sequences

Sequence	Derived Column
VEA_EXCEPTIONS_S	EXCEPTION_ID

Oracle Proprietary, Confidential Information—Use Restricted by Contract

VEA_LAYERS

The VEA_LAYERS table contains information about branches with each base layer program unit. Each row contains information about

– Which trading partner layer program unit is linked to base layer program unit

– Branching conditions under which trading partner layer program unit is to be called in place of base layer program unit

– Whether branch is active or inactive Layer developers create branches using layer workbench and generate utility generate code to conditionally execute active branches , that is , trading partner layer program units using this information.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_LAYER_HEADERS	LAYER_PROVIDER_CODE LAYER_HEADER_ID	LAYER_PROVIDER_CODE LAYER_HEADER_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER3_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER4_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER9_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER10_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER1_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER2_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER5_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER6_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER7_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_LP_CODE TPS_PARAMETER8_ID
VEA_PROGRAM_UNITS	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID	PROGRAM_UNIT_LP_CODE NEW_PROGRAM_UNIT_ID

Column Descriptions

Name	Null?	Type	Description
LAYER_ID (PK)	NOT NULL	NUMBER	Branch identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
LAYER_HEADER_ID	NOT NULL	NUMBER	Layer header identifier
NEW_PROGRAM_UNIT_ID	NOT NULL	NUMBER	Trading partner layer program unit identifier
PROGRAM_UNIT_LP_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier for trading partner layer program unit
SEQUENCE_NUMBER	NOT NULL	NUMBER	Sequence of the branch within the base layer program unit
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column

CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DESCRIPTION	NULL	VARCHAR2(1000)	Branch description
ACTIVE_FLAG	NULL	VARCHAR2(1)	Indicates whether branch is active or inactive, Y or N flag
TPS_PARAMETER1_ID	NULL	NUMBER	Trading partner selector program unit parameter1 identifier
TPS_PARAMETER1_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter1 value
TPS_PARAMETER2_ID	NULL	NUMBER	Trading partner selector program unit parameter2 identifier
TPS_PARAMETER2_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter2 value
TPS_PARAMETER3_ID	NULL	NUMBER	Trading partner selector program unit parameter3 identifier
TPS_PARAMETER3_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter3 value
TPS_PARAMETER4_ID	NULL	NUMBER	Trading partner selector program unit parameter4 identifier
TPS_PARAMETER4_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter4 value
TPS_PARAMETER5_ID	NULL	NUMBER	Trading partner selector program unit parameter5 identifier
TPS_PARAMETER5_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter5 value
TPS_PARAMETER6_ID	NULL	NUMBER	Trading partner selector program unit parameter6 identifier
TPS_PARAMETER6_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter6 value
TPS_PARAMETER7_ID	NULL	NUMBER	Trading partner selector program unit parameter7 identifier
TPS_PARAMETER7_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter7 value
TPS_PARAMETER8_ID	NULL	NUMBER	Trading partner selector program unit parameter8 identifier
TPS_PARAMETER8_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter8 value
TPS_PARAMETER9_ID	NULL	NUMBER	Trading partner selector program unit parameter9 identifier
TPS_PARAMETER9_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter9 value
TPS_PARAMETER10_ID	NULL	NUMBER	Trading partner selector program unit parameter10 identifier
TPS_PARAMETER10_VALUE	NULL	VARCHAR2(500)	Trading partner selector program unit parameter10 value

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_LAYERS_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	LAYER_ID

Sequences

Sequence	Derived Column
VEA_LAYERS_S	LAYER_ID

VEA_LAYER_HEADERS

The VEA_LAYER_HEADERS table contains information about

– how base layer program units are linked with callout program units

– Which trading partner selector program unit is to be executed to derive the branch criteria Layer developers or customers specify this information using layer workbench and it is used by generate utility to generate code for calling callout program units and trading partner selector program units.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_PROGRAM_UNITS	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID	PROGRAM_UNIT_LP_CODE PROGRAM_UNIT_ID
VEA_PROGRAM_UNITS	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID	TPS_PROGRAM_UNIT_LP_CODE TPS_PROGRAM_UNIT_ID

Column Descriptions

Name	Null?	Type	Description
LAYER_HEADER_ID (PK)	NOT NULL	NUMBER	Layer header identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
PROGRAM_UNIT_ID	NOT NULL	NUMBER	Program unit identifier
PROGRAM_UNIT_LP_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier for program unit
TPS_PROGRAM_UNIT_ID	NOT NULL	NUMBER	Trading partner selector program unit identifier
TPS_PROGRAM_UNIT_LP_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier for Trading partner selector program unit
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
CONDITION_TYPE	NULL	VARCHAR2(1)	Indicates whether callout is before(B) or after(A) the base layer program unit
DESCRIPTION	NULL	VARCHAR2(1000)	Descriptive text for the header information

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_LAYER_HEADERS_U1	UNIQUE	1 2	LAYER_PROVIDER_CODE LAYER_HEADER_ID

Sequences

Sequence	Derived Column
VEA_LAYER_HEADERS_S	LAYER_HEADER_ID

VEA_LAYER_LICENSES

The VEA_LAYER_LICENSES table contains information about customers licensed to use trading partner layers. Layer developers can licenses trading partner layers to customers using the layer workbench and layer manage utility uses the license information to install layers only for licensed customers.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_TP_LAYERS	LAYER_PROVIDER_CODE TP_LAYER_ID	LAYER_PROVIDER_CODE TP_LAYER_ID

Column Descriptions

Name	Null?	Type	Description
LAYER_PROVIDER_CODE	NOT NULL	VARCHAR2(500)	Layer developer identifier
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
CUSTOMER_NAME	NULL	VARCHAR2(100)	Name of the customer licensed to the trading partner layer
DESCRIPTION	NULL	VARCHAR2(1000)	License description
TP_LAYER_ID	NULL	NUMBER	Trading partner layer identifier

VEA_PACKAGES

The VEA_PACKAGES table contains information about following types of packages registered with the trading partner architecture repository

– Base layer packages

– Trading partner selector packages

– Trading partner layer packages

– Callout packages

– Packages generated by the trading partner architecture

Each row contains information about files containing the package code, application each package belongs to and whether package is client side or server side. The import utility parses package files and registers information about requisite packages in the repository. Layer workbench displays this information and generate utility uses this information to find out packages to be generated and corresponding file names.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_TP_LAYERS	LAYER_PROVIDER_CODE TP_LAYER_ID	LAYER_PROVIDER_CODE TP_LAYER_ID

Column Descriptions

Name	Null?	Type	Description
PACKAGE_ID (PK)	NOT NULL	NUMBER	Package Identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
CLIENT_SERVER_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether package is client(C) or server(S) side package
GENERATE_FLAG	NULL	VARCHAR2(1)	Not currently used
TPA_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether package will be generated by the trading partner architecture, Y or N Flag
NAME	NOT NULL	VARCHAR2(30)	Package name
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
SPECIFICATION_FILENAME	NULL	VARCHAR2(50)	Specification or library file name
BODY_FILENAME	NULL	VARCHAR2(50)	Body file name

LABEL	NULL	VARCHAR2(100)	Package label
VERSION_NUMBER	NULL	VARCHAR2(100)	Current version number for packages generated by the architecture
DESCRIPTION	NULL	VARCHAR2(1000)	Package description
APPLICATION_SHORT_NAME	NULL	VARCHAR2(3)	Short name of the application , package belongs to
TP_LAYER_ID	NULL	NUMBER	Identifier for the trading partner layer, package belongs to

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_PACKAGES_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PACKAGE_ID
VEA_PACKAGES_U2	UNIQUE	1	APPLICATION_SHORT_NAME
		2	CLIENT_SERVER_FLAG
		3	NAME

Sequences

Sequence	Derived Column
VEA_PACKAGES_S	PACKAGE_ID

VEA_PARAMETERS

The VEA_PARAMETERS table contains information about signatures of program units registered with the trading partner architecture repository

– Base layer program units

– Trading partner selector program units

– Trading partner layer program units

– Callout program units

Each row contains information about whether the parameter type, its datatype, sequence and default value. The import utility parses package files and registers signature information about requisite program units in the repository. Layer workbench displays this information and generate utility uses this information to generate the code to call the registered program units as specified by layer developers.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_PROGRAM_UNITS	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID

Column Descriptions

Name	Null?	Type	Description
PARAMETER_ID (PK)	NOT NULL	NUMBER	Parameter identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
PROGRAM_UNIT_ID	NOT NULL	NUMBER	Program unit identifier
PARAMETER_TYPE	NOT NULL	VARCHAR2(1)	Type of parameter, valid values are I (IN), O (OUT) and B (IN OUT)
PARAMETER_SEQ	NOT NULL	NUMBER	Sequence of parameter within the program unit
NAME	NOT NULL	VARCHAR2(30)	Parameter name
DATATYPE	NOT NULL	VARCHAR2(500)	Datatype of the parameter
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DEFAULT_VALUE	NULL	VARCHAR2(500)	Default value for the parameter
DESCRIPTION	NULL	VARCHAR2(1000)	Parameter description

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_PARAMETERS_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PARAMETER_ID
VEA_PARAMETERS_U2	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PROGRAM_UNIT_ID
		3	NAME
VEA_PARAMETERS_U3	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PROGRAM_UNIT_ID
		3	PARAMETER_SEQ

Sequences

Sequence	Derived Column
VEA_PARAMETERS_S	PARAMETER_ID

VEA_PARAMETER_MAPPINGS

the VEA_PARAMETER_MAPPINGS table contains information about mapping between

- Parameters of base layer program unit
- Parameters of trading partner selector program unit.

It specifies IN / IN OUT parameters of base layer program units which will be passed to IN / IN OUT parameters of trading partner selector program units when called from the code generated by the architecture. Layer developers specify the mapping using layer workbench while selecting a trading partner selector for a base layer program unit and generate utility derive exact calling syntax for trading partner selector program units using this information.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_LAYER_HEADERS	LAYER_PROVIDER_CODE LAYER_HEADER_ID	LAYER_PROVIDER_CODE LAYER_HEADER_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	PROGRAM_UNIT_PARAM_LP_CODE PROGRAM_UNIT_PARAMETER_ID
VEA_PARAMETERS	LAYER_PROVIDER_CODE PARAMETER_ID	TPS_PARAMETER_LP_CODE TPS_PARAMETER_ID

Column Descriptions

Name	Null?	Type	Description
PARAMETER_MAPPING_ID (PK)	NOT NULL	NUMBER	Parameter mapping identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
LAYER_HEADER_ID	NOT NULL	NUMBER	Layer header identifier
TPS_PARAMETER_ID	NOT NULL	NUMBER	Parameter identifier for trading partner selector program unit
TPS_PARAMETER_LP_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier for trading partner selector program unit parameter
PROGRAM_UNIT_PARAMETER_ID	NOT NULL	NUMBER	Parameter identifier for the base layer program unit
PROGRAM_UNIT_PARAM_LP_CODE	NOT NULL	VARCHAR2(30)	Layer developer identifier for the base layer program unit parameter
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DESCRIPTION	NULL	VARCHAR2(1000)	Descriptive text for the parameter mapping

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_PARAMETER_MAPPINGS_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PARAMETER_MAPPING_ID
VEA_PARAMETER_MAPPINGS_U2	UNIQUE	1	LAYER_PROVIDER_CODE
		2	LAYER_HEADER_ID
		3	TPS_PARAMETER_LP_CODE
		4	TPS_PARAMETER_ID

Sequences

Sequence	Derived Column
VEA_PARAMETER_MAPPINGS_S	PARAMETER_MAPPING_ID

VEA_PROGRAM_UNITS

The VEA_PROGRAM_UNITS table contains information about following types of program units registered with the trading partner architecture repository

– Base layer program units

– Trading partner selector program units

– Trading partner layer program units

– Callout program units

– Program units generated by the trading partner architecture

Each row contains information about whether the program unit is customizable, whether it is trading partner selector and type of program unit whether it is procedure or function. It also stores mapping between base layer program unit and corresponding program unit to be generated by the architecture. The import utility parses package files and registers information about requisite program units in the repository. Layer workbench displays this information and generate utility uses this information to find out program units to be generated and base layer program units it is linked to.

Foreign Keys

Primary Key Table	Primary Key Column	Foreign Key Column
VEA_PACKAGES	LAYER_PROVIDER_CODE PACKAGE_ID	LAYER_PROVIDER_CODE PACKAGE_ID
VEA_PROGRAM_UNITS	LAYER_PROVIDER_CODE PROGRAM_UNIT_ID	LAYER_PROVIDER_CODE TPA_PROGRAM_UNIT_ID

Column Descriptions

Name	Null?	Type	Description
PROGRAM_UNIT_ID (PK)	NOT NULL	NUMBER	Program unit identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
PACKAGE_ID	NOT NULL	NUMBER	Package identifier
PROGRAM_UNIT_TYPE	NOT NULL	VARCHAR2(1)	Indicates whether program unit is a procedure(P) or function(F)
PUBLIC_FLAG	NOT NULL	VARCHAR2(1)	Not currently used, always defaulted to Y
CUSTOMIZABLE_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether the program unit is customizable, Y/N flag
TPS_FLAG	NOT NULL	VARCHAR2(1)	Indicates if the program unit is a trading partner selector, Y/N flag
NAME	NOT NULL	VARCHAR2(30)	Name of the program unit
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column

REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	
LABEL	NULL	VARCHAR2(100)	Label of the program unit
RETURN_TYPE	NULL	VARCHAR2(500)	Return type if program unit is a function
TPA_PROGRAM_UNIT_ID	NULL	NUMBER	Identifier for the base layer program unit, if the program unit is generated by the architecture
DESCRIPTION	NULL	VARCHAR2(1000)	Description for the program unit

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_PROGRAM_UNITS_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	PROGRAM_UNIT_ID

Sequences

Sequence	Derived Column
VEA_PROGRAM_UNITS_S	PROGRAM_UNIT_ID

VEA_RELEASE_DETAILS

Not currently used.

Column Descriptions

Name	Null?	Type	Description
RELEASE_ID (PK)	NOT NULL	NUMBER	Release identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
FILE_NAME (PK)	NOT NULL	VARCHAR2(50)	Name of the file being shipped
VERSION_NUMBER	NOT NULL	VARCHAR2(200)	Version number of the file being shipped
APPLICATION_SHORT_NAME (PK)	NOT NULL	VARCHAR2(3)	Short name of the application file belongs to
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DESCRIPTION	NULL	VARCHAR2(1000)	Description for the file
ARU_NUMBER	NULL	VARCHAR2(1000)	ARU number corresponding to the version number of the file
BUG_NUMBER	NULL	VARCHAR2(1000)	Bug number corresponding to the version number of the file

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_RELEASE_DETAILS_U1	UNIQUE	1	LAYER_PROVIDER_CODE
		2	RELEASE_ID
		3	APPLICATION_SHORT_NAME
		4	FILE_NAME

Sequences

Sequence	Derived Column
VEA_RELEASES_S	RELEASE_ID

VEA_TP_LAYERS

The VEA_TP_LAYERS table contains information about trading partner layers. Each row contains information about trading partner layer name, description and whether it is active or inactive. Layer developers create trading partner layers using layer workbench. Trading partner layers can be made active or inactive using runtime option of the trading partner architecture. Generate utility look at this information to generate code to execute only active layers

Column Descriptions

Name	Null?	Type	Description
TP_LAYER_ID (PK)	NOT NULL	NUMBER	Trading partner layer identifier
LAYER_PROVIDER_CODE (PK)	NOT NULL	VARCHAR2(30)	Layer developer identifier
NAME	NOT NULL	VARCHAR2(50)	Trading partner layer name
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who column
CREATION_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who column
LAST_UPDATE_LOGIN	NULL	NUMBER(15)	Standard Who column
REQUEST_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who column
DESCRIPTION	NULL	VARCHAR2(1000)	Trading partner layer description
ACTIVE_FLAG	NULL	VARCHAR2(1)	Indicates whether a trading partner layer is active or inactive, Y/N flag

Indexes

Index Name	Index Type	Sequence	Column Name
VEA_TP_LAYERS_U1	NOT UNIQUE	1	LAYER_PROVIDER_CODE
		2	TP_LAYER_ID
VEA_TP_LAYERS_U2	NOT UNIQUE	1	LAYER_PROVIDER_CODE
		2	NAME

Sequences

Sequence	Derived Column
VEA_TP_LAYERS_S	TP_LAYER_ID

Index

Reader's Comment Form

Oracle Automotive Trading Partner Toolkit Technical Reference Manual A83735_01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

____.

____

____

____

____

____

____

____

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (650) 506–7000 Fax: (650) 506–7200

If you would like a reply, please give your name, address, and telephone number below:

____

____

____

Thank you for helping us improve our documentation.

CAUTION!

This document contains proprietary, confidential information that is the exclusive property of Oracle Corporation.

If you are an Oracle customer, this numbered copy is registered in your site's name. Use, duplication, and disclosure are severely restricted and subject to restrictions stated in your contract with Oracle Corporation.

If you are an Oracle employee, this numbered copy is registered in your name and must be returned if you leave the company. It is your responsibility not to disclose this information to anyone outside of Oracle who is not licensed to use this product. If you have a prospect who needs to see this information, contact mailid APPSAPP to obtain a special nondisclosure agreement.

If you are neither an Oracle customer with a valid contract for the use of this document nor an Oracle employee, then you received this document in an unauthorized manner and are not legally entitled to possess or read it. Please return it immediately to Oracle. You may call collect to the office of the Executive Vice President Applications Division, to arrange its return.

Copy #

ORACLE®