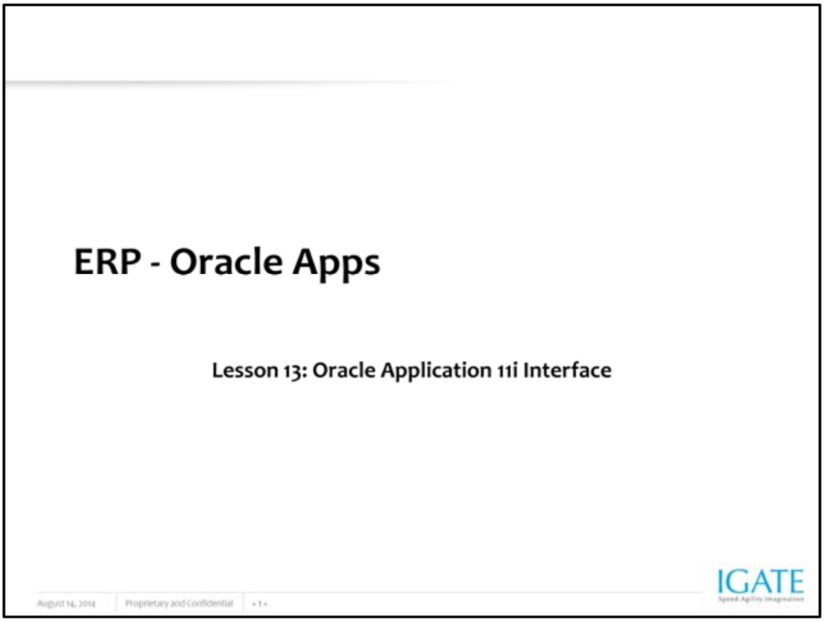
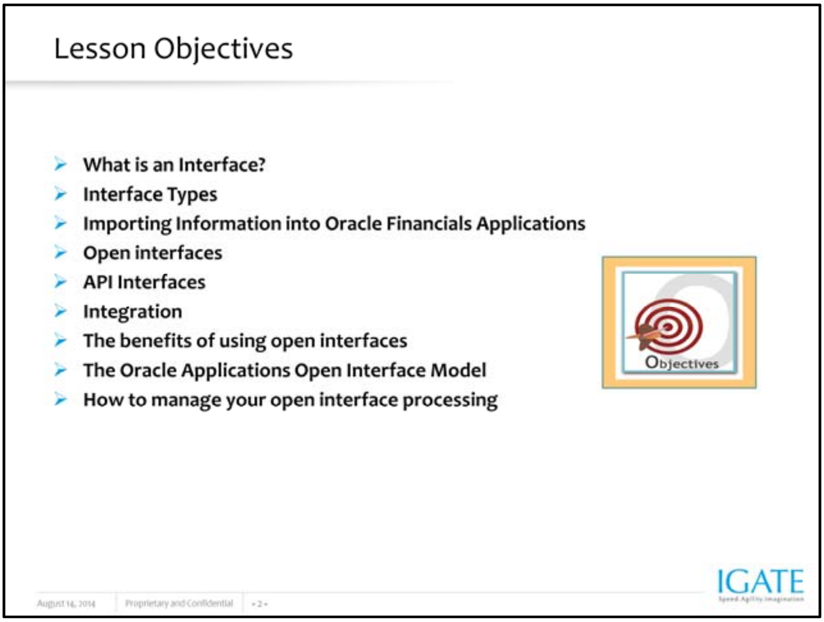


ERP - Oracle Apps

Lesson 13: Oracle Application 11i Interface

August 14, 2014 | Proprietary and Confidential | - 1 -



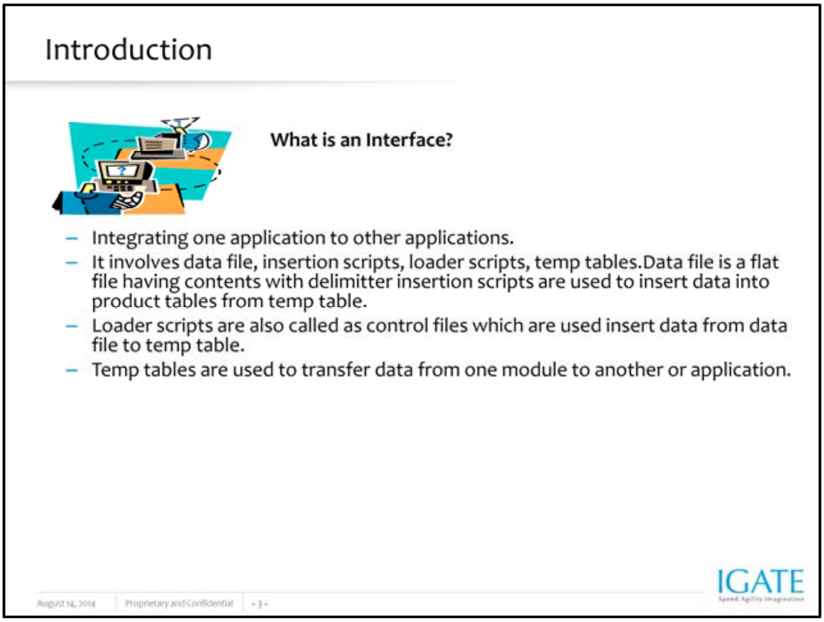
Lesson Objectives

- What is an Interface?
- Interface Types
- Importing Information into Oracle Financials Applications
- Open interfaces
- API Interfaces
- Integration
- The benefits of using open interfaces
- The Oracle Applications Open Interface Model
- How to manage your open interface processing

Objectives

August 14, 2014 | Proprietary and Confidential | - 2 -

IGATE



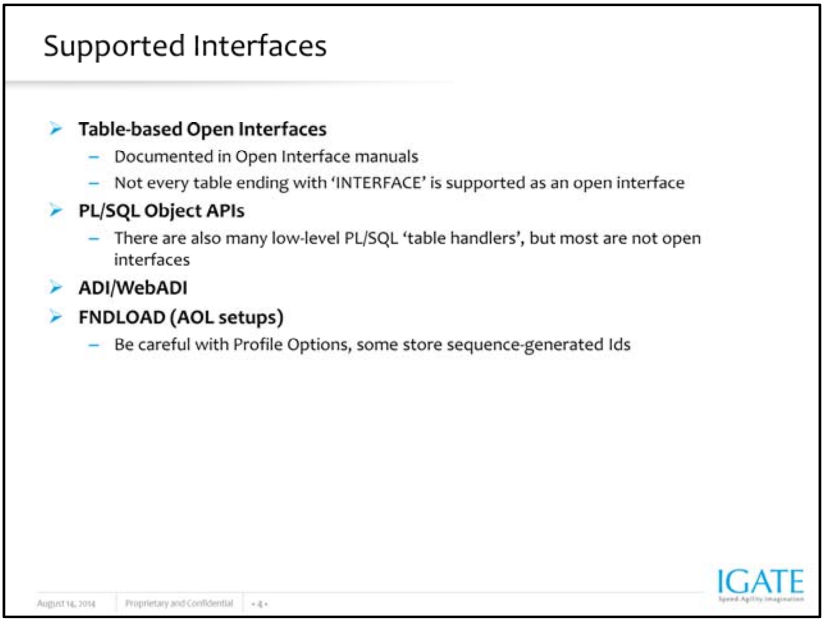
Introduction

What is an Interface?

- Integrating one application to other applications.
- It involves data file, insertion scripts, loader scripts, temp tables.Data file is a flat file having contents with delimitter insertion scripts are used to insert data into product tables from temp table.
- Loader scripts are also called as control files which are used insert data from data file to temp table.
- Temp tables are used to transfer data from one module to another or application.

August 14, 2014 | Proprietary and Confidential | - 3 -

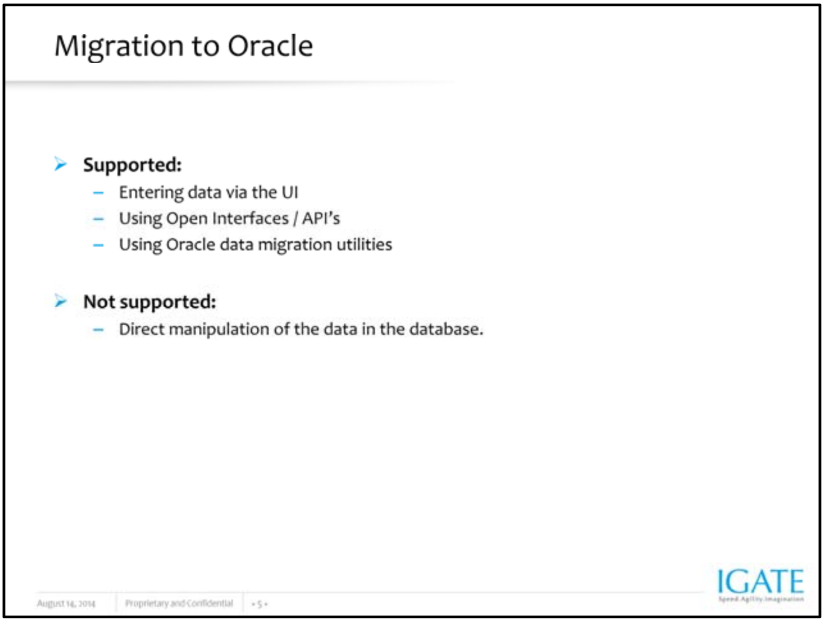
IGATE
Speed.Agility.Imagination



Supported Interfaces

- **Table-based Open Interfaces**
  - Documented in Open Interface manuals
  - Not every table ending with 'INTERFACE' is supported as an open interface
- **PL/SQL Object APIs**
  - There are also many low-level PL/SQL 'table handlers', but most are not open interfaces
- **ADI/WebADI**
- **FNDLOAD (AOL setups)**
  - Be careful with Profile Options, some store sequence-generated Ids

August 14, 2014 | Proprietary and Confidential | - 4 -

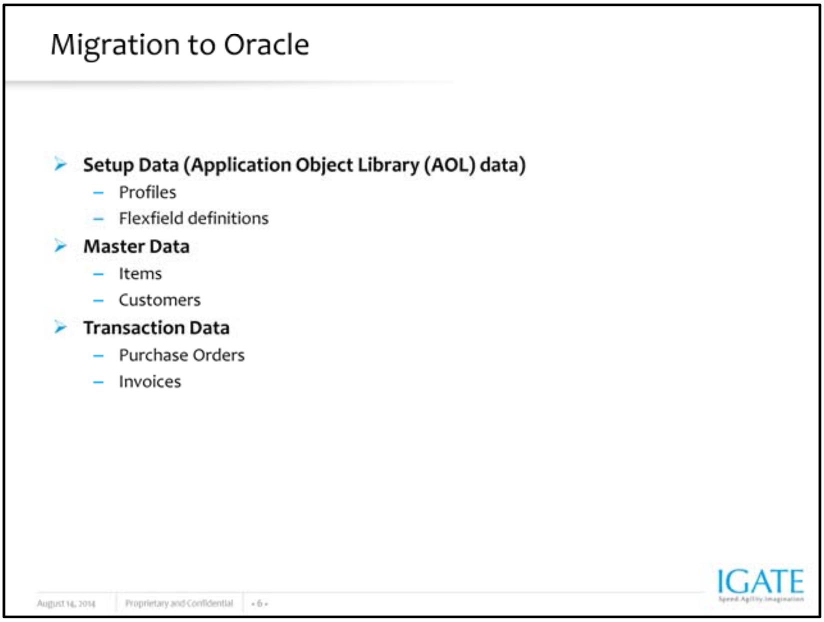


Migration to Oracle

- **Supported:**
  - Entering data via the UI
  - Using Open Interfaces / API's
  - Using Oracle data migration utilities
- **Not supported:**
  - Direct manipulation of the data in the database.

August 14, 2014 | Proprietary and Confidential | - 5 -

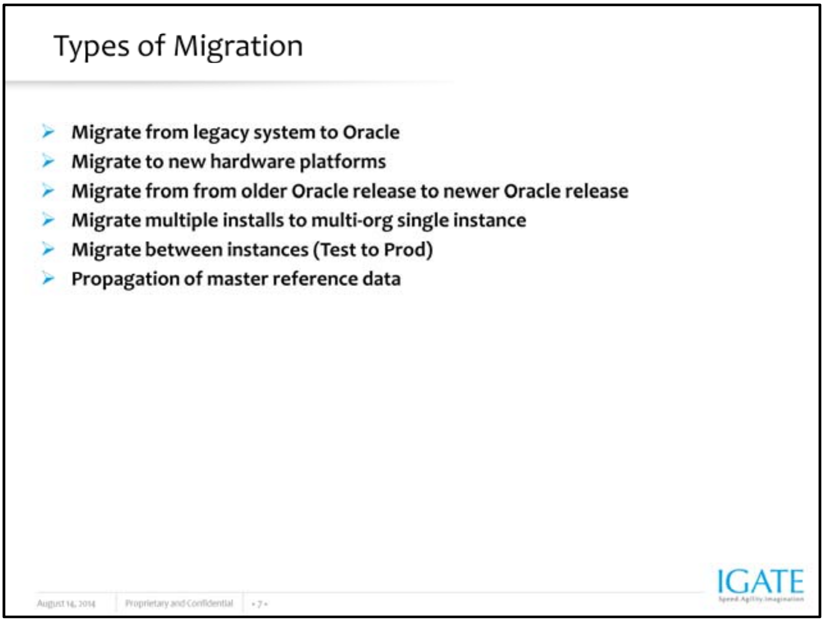
IGATE
Speed. Agility. Imagination



Migration to Oracle

- **Setup Data (Application Object Library (AOL) data)**
  - Profiles
  - Flexfield definitions
- **Master Data**
  - Items
  - Customers
- **Transaction Data**
  - Purchase Orders
  - Invoices

August 14, 2014 | Proprietary and Confidential | - 6 -

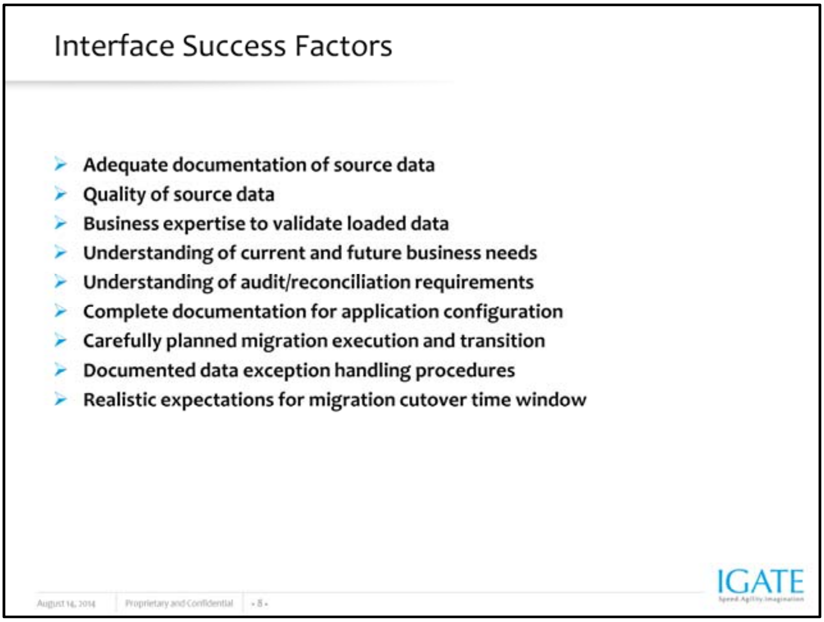


Types of Migration

- Migrate from legacy system to Oracle
- Migrate to new hardware platforms
- Migrate from from older Oracle release to newer Oracle release
- Migrate multiple installs to multi-org single instance
- Migrate between instances (Test to Prod)
- Propagation of master reference data

August 14, 2014 | Proprietary and Confidential | - 7 -

IGATE
Speed.Agility.Imagination

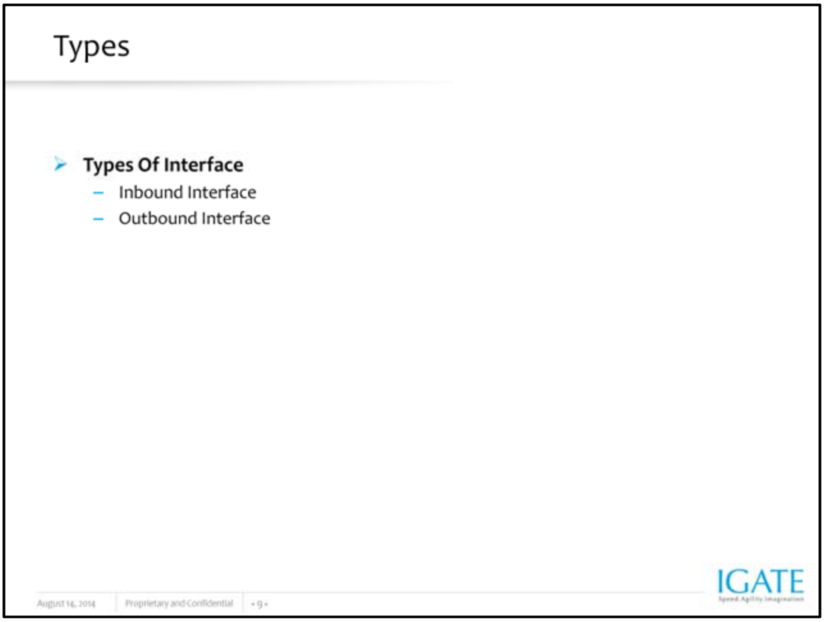


Interface Success Factors

- Adequate documentation of source data
- Quality of source data
- Business expertise to validate loaded data
- Understanding of current and future business needs
- Understanding of audit/reconciliation requirements
- Complete documentation for application configuration
- Carefully planned migration execution and transition
- Documented data exception handling procedures
- Realistic expectations for migration cutover time window

August 14, 2014 | Proprietary and Confidential | - 8 -

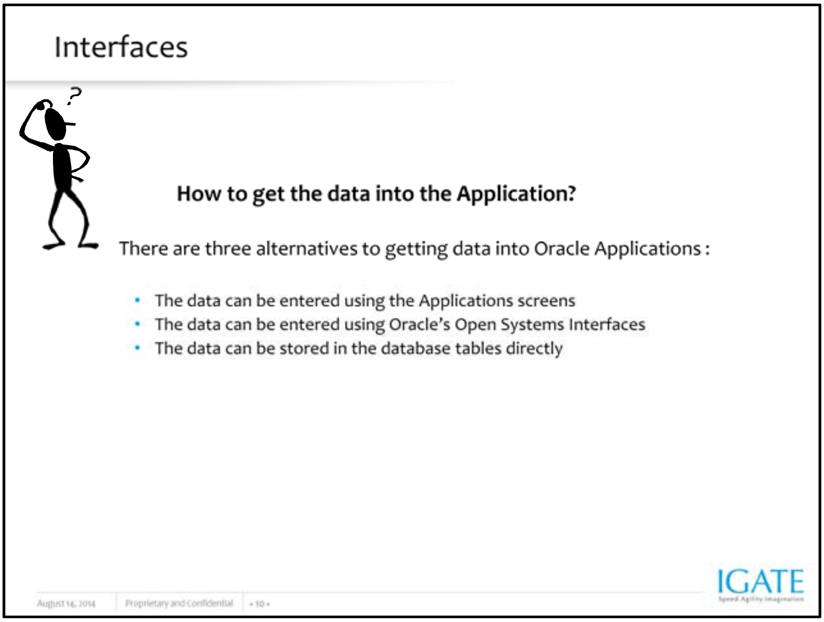
IGATE
Speed.Agility.Imagination



Types

- **Types Of Interface**
  - Inbound Interface
  - Outbound Interface

August 14, 2014 | Proprietary and Confidential | • 9 •



Interfaces

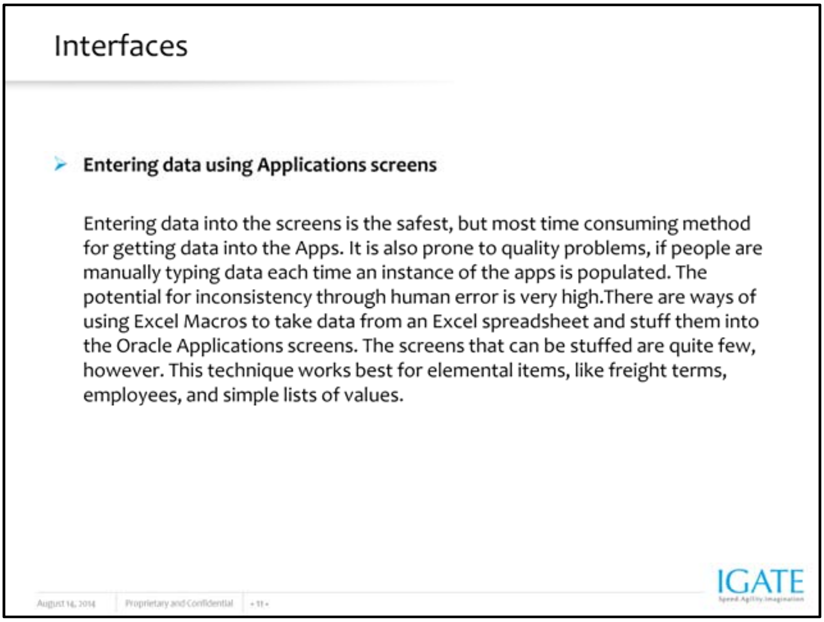
How to get the data into the Application?

There are three alternatives to getting data into Oracle Applications :

- The data can be entered using the Applications screens
- The data can be entered using Oracle's Open Systems Interfaces
- The data can be stored in the database tables directly

August 14, 2014 | Proprietary and Confidential | • 10 •

IGATE
Speed.Agility.Imagination



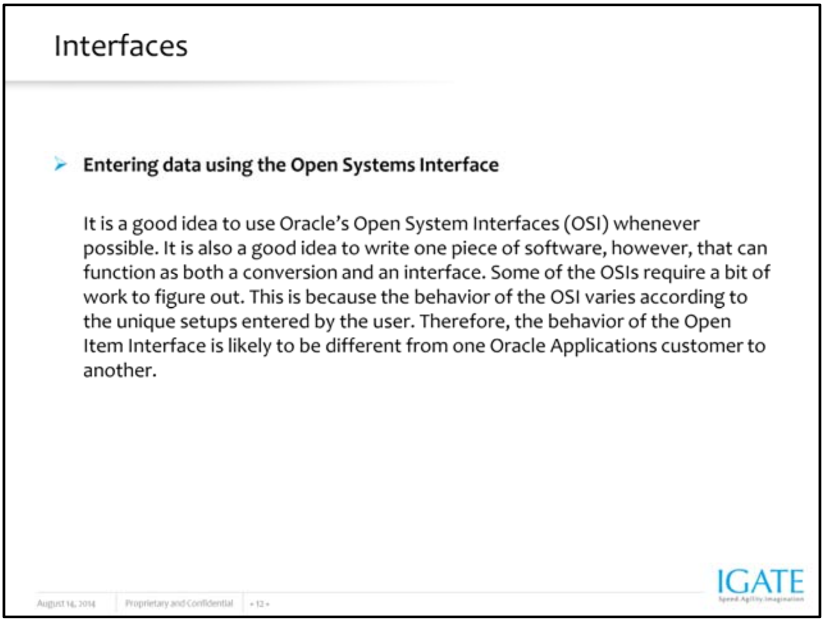
Interfaces

- **Entering data using Applications screens**

  Entering data into the screens is the safest, but most time consuming method for getting data into the Apps. It is also prone to quality problems, if people are manually typing data each time an instance of the apps is populated. The potential for inconsistency through human error is very high.There are ways of using Excel Macros to take data from an Excel spreadsheet and stuff them into the Oracle Applications screens. The screens that can be stuffed are quite few, however. This technique works best for elemental items, like freight terms, employees, and simple lists of values.

August 14, 2014 | Proprietary and Confidential | - 11 -

IGATE
Speed.Agility.Imagination



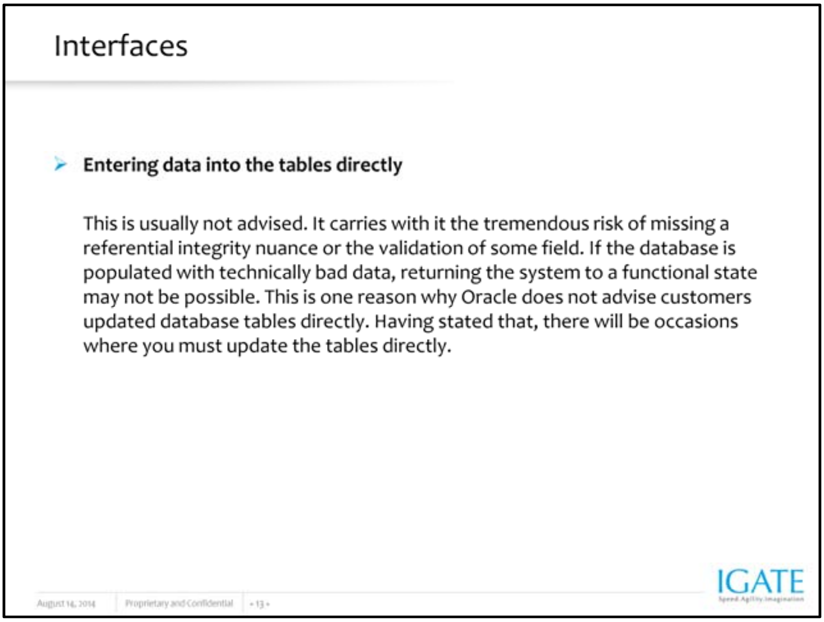
Interfaces

- **Entering data using the Open Systems Interface**

  It is a good idea to use Oracle's Open System Interfaces (OSI) whenever possible. It is also a good idea to write one piece of software, however, that can function as both a conversion and an interface. Some of the OSIs require a bit of work to figure out. This is because the behavior of the OSI varies according to the unique setups entered by the user. Therefore, the behavior of the Open Item Interface is likely to be different from one Oracle Applications customer to another.

August 14, 2014 | Proprietary and Confidential | - 12 -

IGATE
Speed.Agility.Imagination

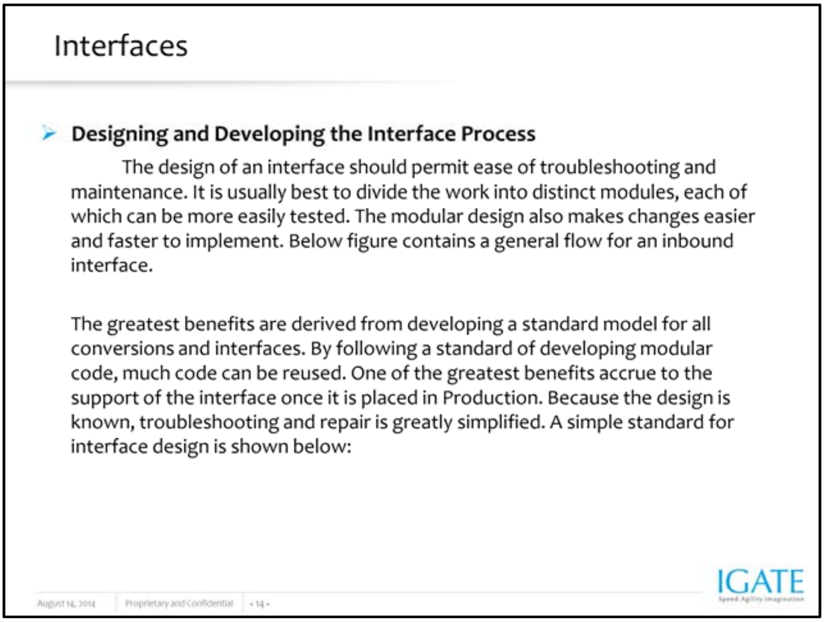


Interfaces

- **Entering data into the tables directly**

  This is usually not advised. It carries with it the tremendous risk of missing a referential integrity nuance or the validation of some field. If the database is populated with technically bad data, returning the system to a functional state may not be possible. This is one reason why Oracle does not advise customers updated database tables directly. Having stated that, there will be occasions where you must update the tables directly.

August 14, 2014 | Proprietary and Confidential | - 13 -

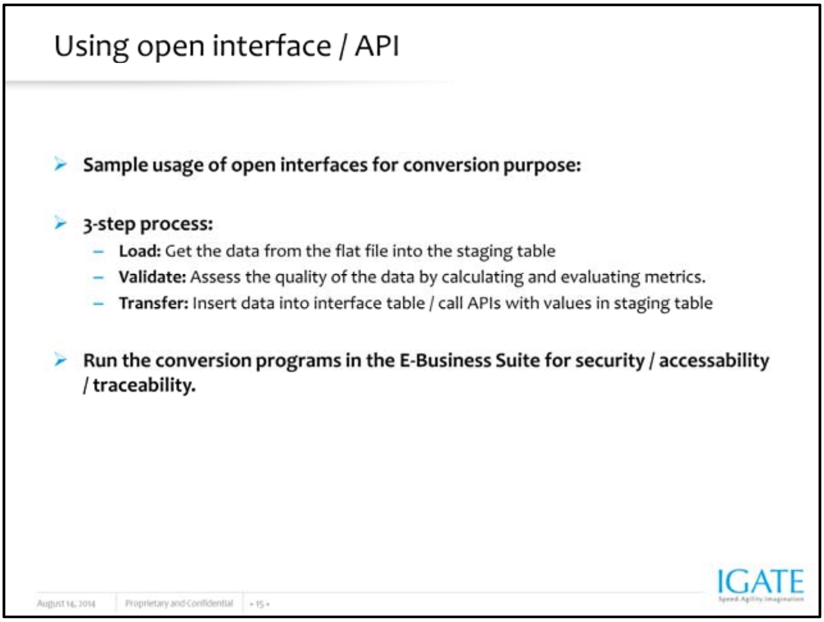


Interfaces

- **Designing and Developing the Interface Process**

  The design of an interface should permit ease of troubleshooting and maintenance. It is usually best to divide the work into distinct modules, each of which can be more easily tested. The modular design also makes changes easier and faster to implement. Below figure contains a general flow for an inbound interface.

  The greatest benefits are derived from developing a standard model for all conversions and interfaces. By following a standard of developing modular code, much code can be reused. One of the greatest benefits accrue to the support of the interface once it is placed in Production. Because the design is known, troubleshooting and repair is greatly simplified. A simple standard for interface design is shown below:

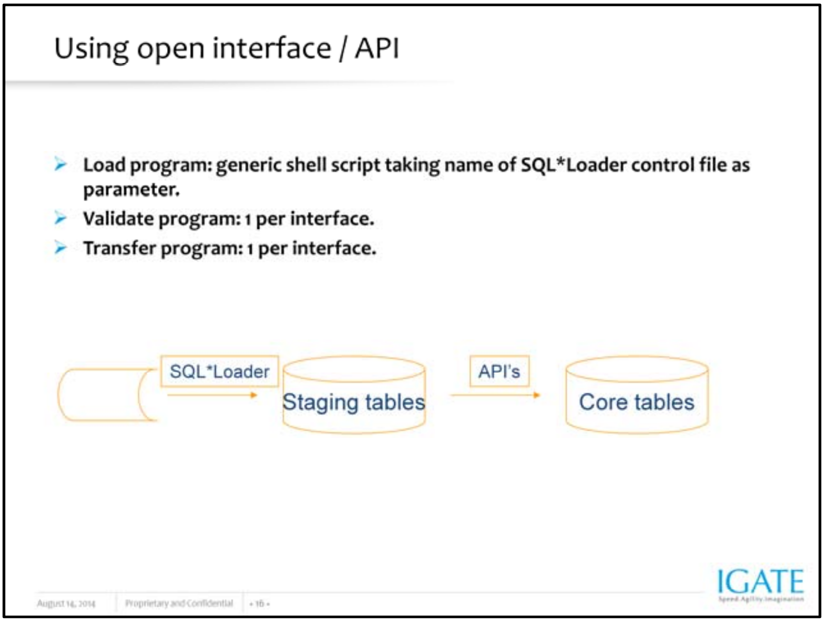


Using open interface / API

- **Sample usage of open interfaces for conversion purpose:**
- **3-step process:**
  - **Load:** Get the data from the flat file into the staging table
  - **Validate:** Assess the quality of the data by calculating and evaluating metrics.
  - **Transfer:** Insert data into interface table / call APIs with values in staging table
- **Run the conversion programs in the E-Business Suite for security / accessability / traceability.**

August 14, 2014 | Proprietary and Confidential | - 15 -

IGATE
Speed.Agility.Imagination



Using open interface / API

- Load program: generic shell script taking name of SQL*Loader control file as parameter.
- Validate program: 1 per interface.
- Transfer program: 1 per interface.

SQL*Loader

Staging tables

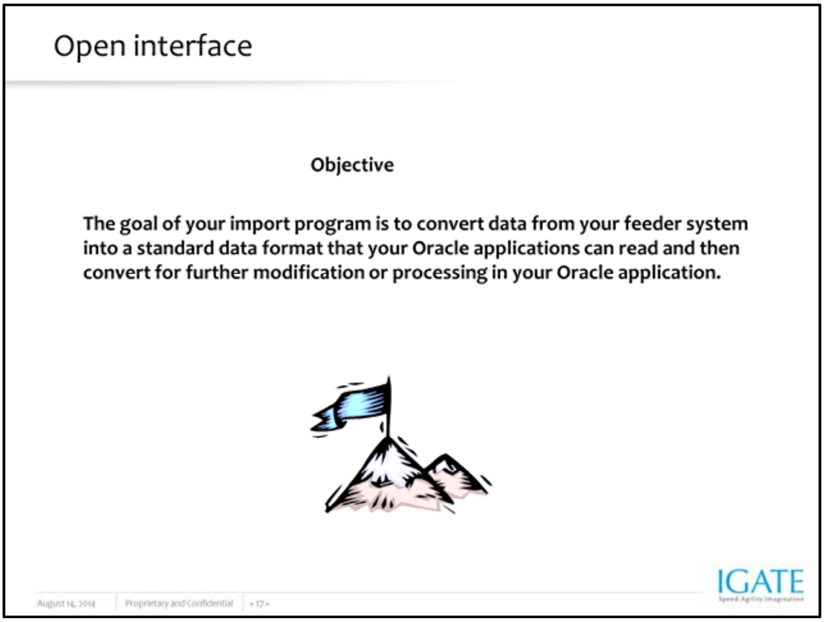
API's

Core tables

August 14, 2014 | Proprietary and Confidential | • 16 •

IGATE

Speed.Agility.Imagination



Open interface

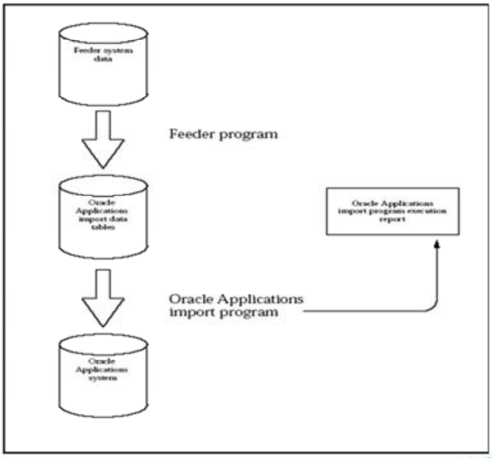
Objective

The goal of your import program is to convert data from your feeder system into a standard data format that your Oracle applications can read and then convert for further modification or processing in your Oracle application.

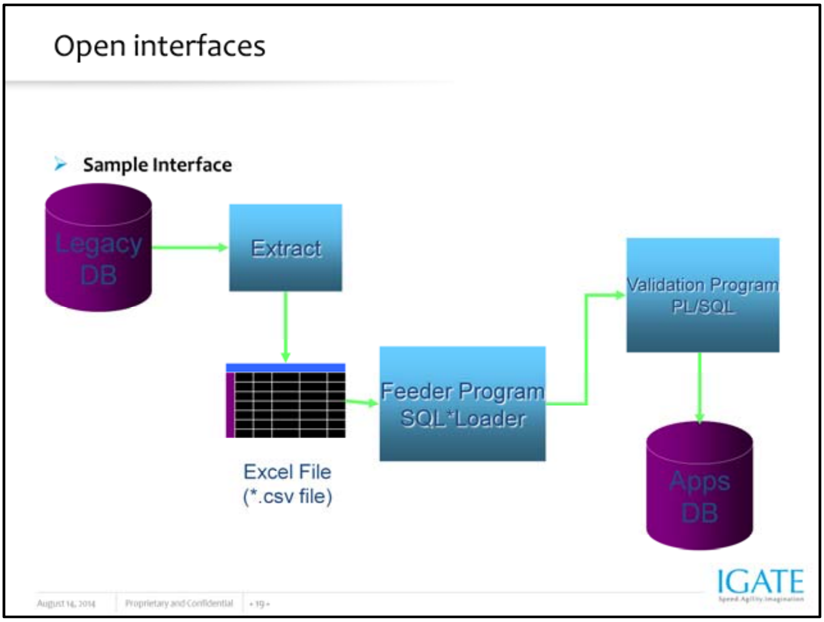
August 14, 2014 | Proprietary and Confidential | - 17 -

Open interfaces

Importing Program


Feeder system data
Feeder program
Oracle Applications import data tables
Oracle Applications import program
Oracle Applications import program execution report
Oracle Applications system

August 14, 2014 | Proprietary and Confidential | - 18 -



Open interfaces

- Sample Interface

Legacy DB

Extract

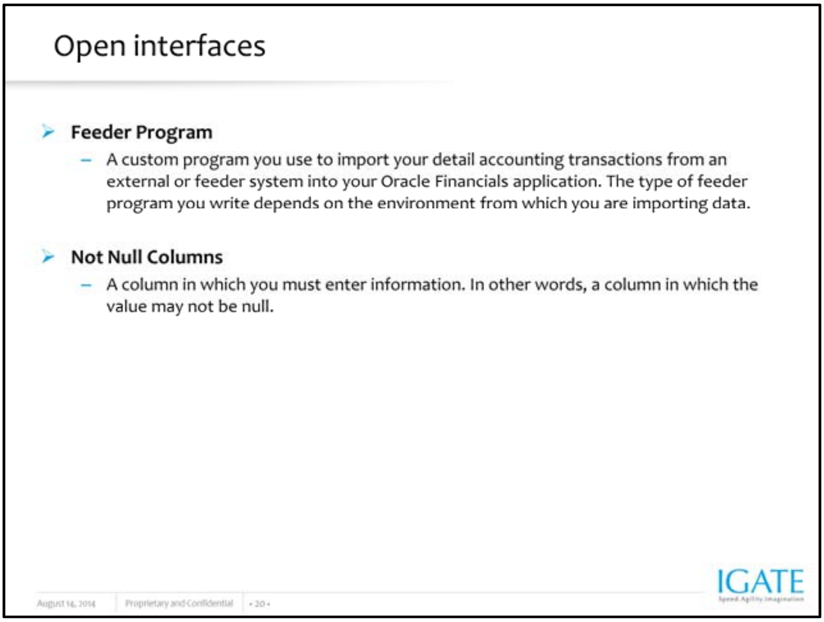
Excel File
(*.csv file)

Feeder Program
SQL*Loader

Validation Program
PL/SQL

Apps DB

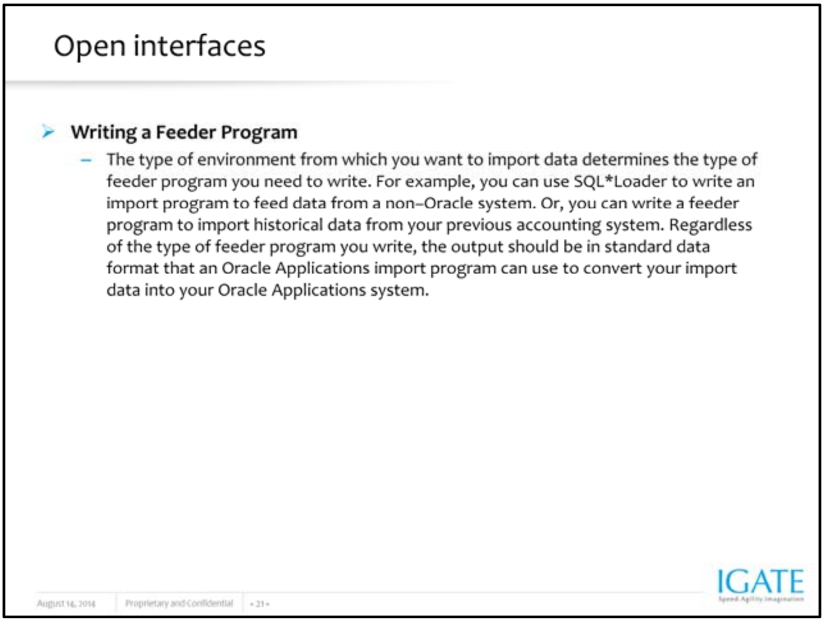
August 14, 2014 | Proprietary and Confidential | - 19 -



Open interfaces

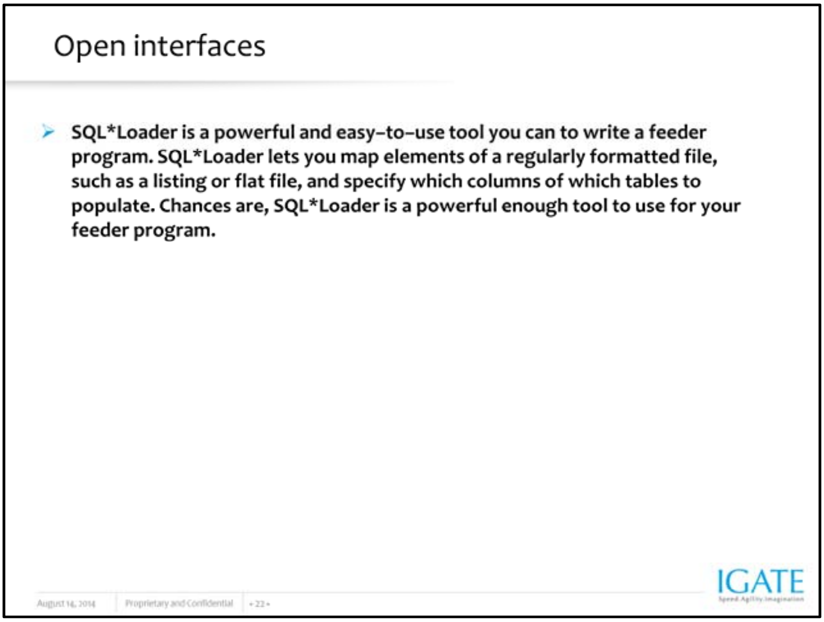
- **Feeder Program**
  - A custom program you use to import your detail accounting transactions from an external or feeder system into your Oracle Financials application. The type of feeder program you write depends on the environment from which you are importing data.
- **Not Null Columns**
  - A column in which you must enter information. In other words, a column in which the value may not be null.

August 14, 2014 | Proprietary and Confidential | - 20 -



Open interfaces

- **Writing a Feeder Program**
  - The type of environment from which you want to import data determines the type of feeder program you need to write. For example, you can use SQL*Loader to write an import program to feed data from a non–Oracle system. Or, you can write a feeder program to import historical data from your previous accounting system. Regardless of the type of feeder program you write, the output should be in standard data format that an Oracle Applications import program can use to convert your import data into your Oracle Applications system.

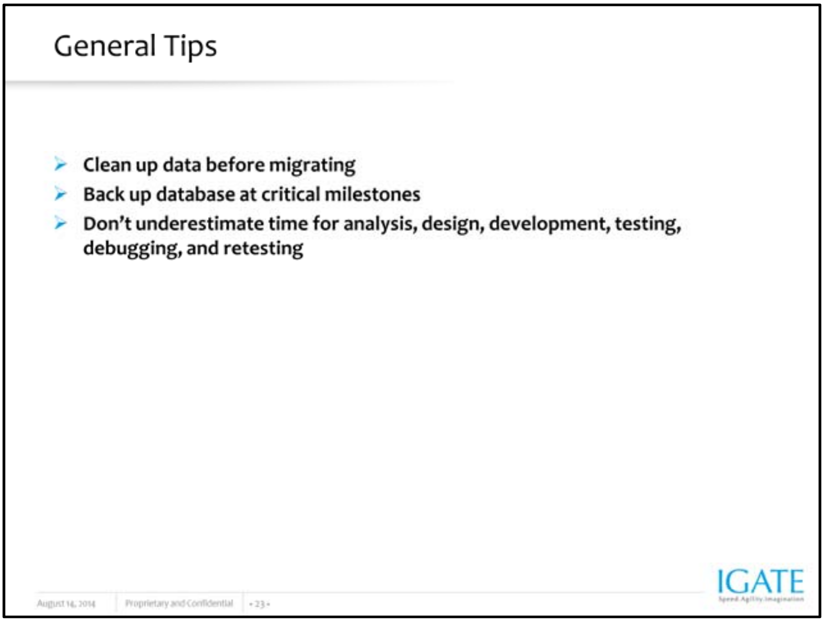


Open interfaces

- SQL*Loader is a powerful and easy–to–use tool you can to write a feeder program. SQL*Loader lets you map elements of a regularly formatted file, such as a listing or flat file, and specify which columns of which tables to populate. Chances are, SQL*Loader is a powerful enough tool to use for your feeder program.

August 14, 2014 | Proprietary and Confidential | • 22 •

IGATE
Speed.Agility.Imagination



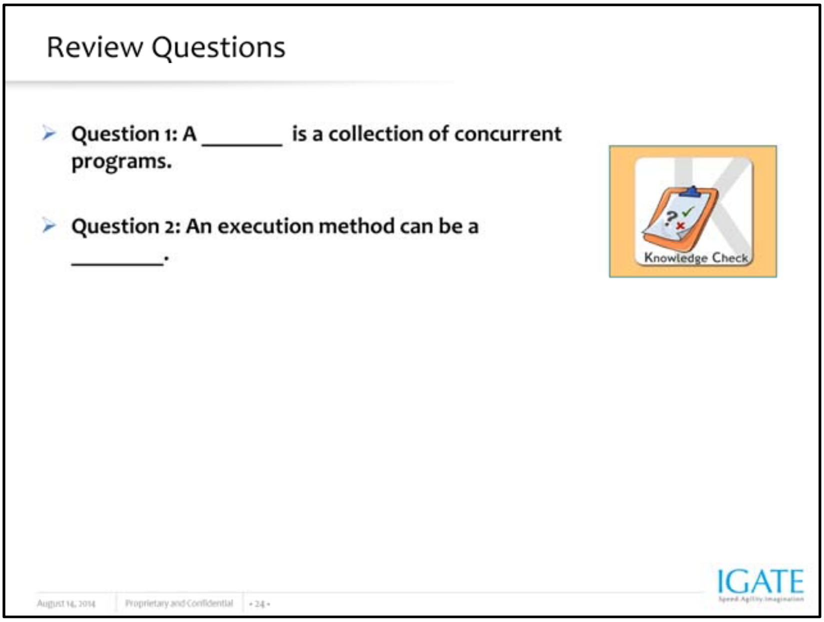
General Tips

- Clean up data before migrating
- Back up database at critical milestones
- Don’t underestimate time for analysis, design, development, testing, debugging, and retesting

August 14, 2014 | Proprietary and Confidential | - 23 -

IGATE
Speed.Agility.Imagination

s



Review Questions

- Question 1: A ________ is a collection of concurrent programs.

- Question 2: An execution method can be a _________.

Knowledge Check

August 14, 2014 | Proprietary and Confidential | - 24 -

IGATE
Speed.Agility.Imagination