



**IFSworld.com**



# IFS FOUNDATION1™ IFS DATA MIGRATION

IFS ASSOCIATE COURSE – IFS APPLICATIONS™ 10

IFS ACADEMY



# WELCOME

- Timings
- Safety
- Ground rules
- Objectives



## YOUR TRAINER

Name:

Contact number:

Email:

| TOPICS                    | 01<br>INTRODUCTION<br>TO DATA<br>MIGRATION    | 02<br>FILE<br>MIGRATION | IFS |
|---------------------------|---|-------------------------|-----|
| 03<br>SOURCE<br>MIGRATION | 04<br>REPLICATION                             | 05<br>REPLICATION OUT   |     |
| 06<br>EXCEL<br>MIGRATION  | 07<br>EXPORT &<br>IMPORT<br>MIGRATION<br>JOBS |                         |     |

## SAFETY AND GROUND RULES

- Fire Alarm – process – exits – muster
- Phones off or on silent
- Be social, switch off Social Media
- Respect
- Access



### GROUND RULES & HOUSEKEEPING

#### WELCOME TO THE COURSE!

Please take note of the fire, health & safety procedures at this venue.

During the course, the only rules we all need to observe are based around respect for each person, no matter what their social background, culture, sex or abilities are. This will manifest itself in allowing everyone to speak and be heard, not using any stereotypical remarks about any social group etc.

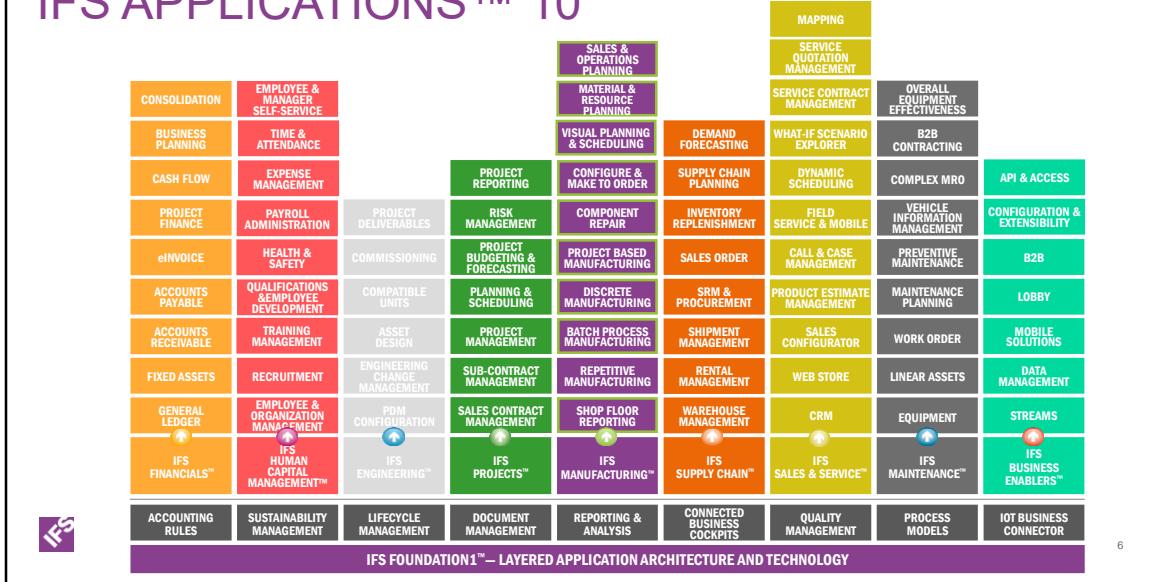
Please respect everyone's point of view, except if it is liable to cause offense. Use of inappropriate language is not acceptable. Thank you for your consideration.

As for communication and mobile devices, at least put your mobile phones to silent and only answer in case of emergency, otherwise use breaks to call back. When answering during the course, please leave the class room without interrupting the course (e.g. by loud speaking)..

For tablets, we cannot see what you are doing but expect that if they are used during the course they are exclusively used for the course, not to read, respond and update Facebook, mail etc.

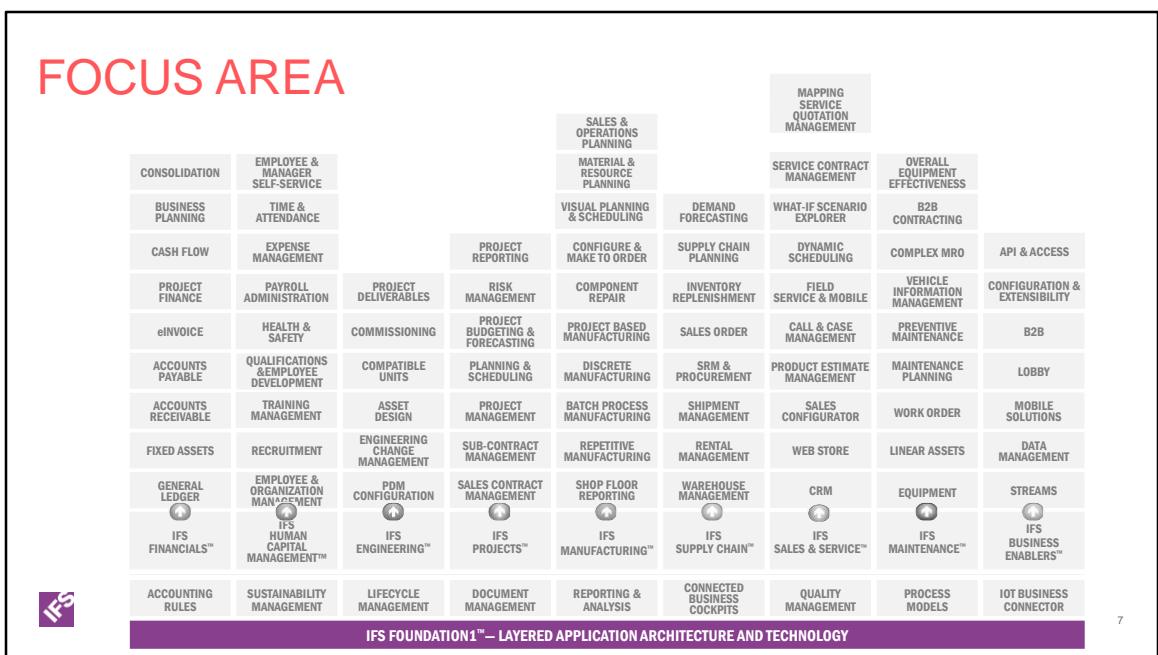
We would like to have your undivided attention.

# IFS APPLICATIONS™ 10



6

# FOCUS AREA



7



# IFS DATA MIGRATION

## COURSE OBJECTIVES

- At the end of this course, the students will have gained insight into IFS Applications™ data migration framework and learn the techniques and best practices to migrate data from and into legacy systems





Link to Application

User Name(s)

User Password(s)



## INTRODUCTION TO DATA MIGRATION

### LESSON OBJECTIVES

By the end of this lesson, students should know:

- The History and Overview of IFS Data Migration
- The Main Data Flow in Data Migration
- Vocabulary used in Data Migration
- The Windows used for data migration in IFS Solution Manager



## BACK TO BASICS

# LOGICAL UNITS

- A Logical Unit contains elements needed to maintain a business object
- Business Object
  - Customer
  - Part Catalog
- LU PartCatalog
  - Table : PART\_CATALOG\_TAB (Storage)
  - Index : PART\_CATALOG\_PK (Storage)
  - View : PART\_CATALOG (Business Logic)
  - Package : PART\_CATALOG\_API (Business Logic)



### The Logical Unit Concept

The Logical Unit (LU) is the basic building block for implementation of the OLA structure. It resembles, and is closely related to a class in Object-Oriented terms. An LU is created to define a business object such as a Customer, Inventory Part, or Customer Invoice, for example. Part of the LU definition will be attributes and behaviors (methods) for that business object. Relationships to other LU's are specified as well.

An LU is defined and implemented across all three layers of the OLA model. A typical LU will consist of an Oracle table in the Data Storage Layer, PL/SQL Procedures and Functions and an Oracle View in the Business Logic Layer, and one or more Application Forms in the Presentation Layer.

## BACK TO BASICS

### PACKAGES

- To manipulate data within a Logical Unit, standard procedures within the server code (Package) must be invoked.
- Procedures for data manipulating
  - Part\_Catalog\_API.New\_\_(info\_, objid\_, objversion\_, attr\_, action\_)
  - Part\_Catalog\_API.Modify\_\_(info\_, objid\_, objversion\_, attr\_, action\_)
- To get out data from a LU, use Get-functions in the package
- Functions to retrieve data
  - Part\_Catalog\_API.Get\_Description( part\_no\_ )
  - Part\_Catalog\_API.Get\_Unit\_Code( part\_no\_ )



Public GET-functions to share data with other LUs/Components.

Private procedures for manipulating data. (New\_\_, Modify\_\_)

Attribute string (attr\_) contains data that have been created/changed on the client.

## BACK TO BASICS

### DATA FLOW



## HISTORY & OVERVIEW

### INTRODUCTION TO DATA MIGRATION

- Initiated by IFS Norway
- Well known as FNDMIG
- IFS Foundation1 owns it since 2004
- Used to move data from previous legacy systems
- Loads data through the business logic APIs
- Assures that business rules, validations, and integrity checks are never bypassed



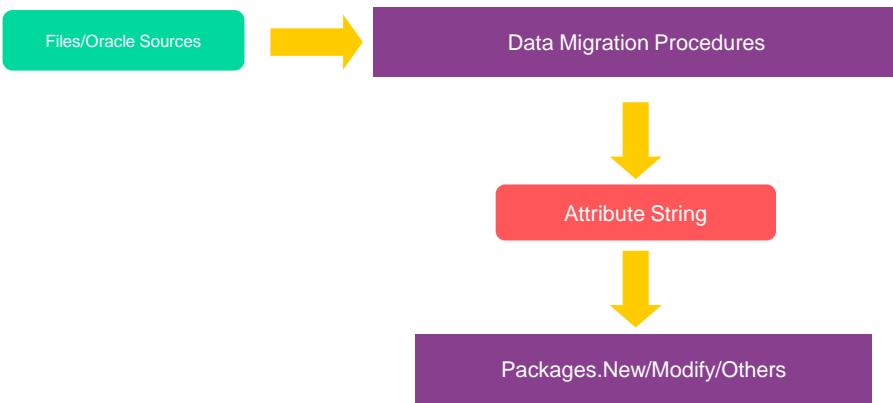
## HISTORY & OVERVIEW

### INTRODUCTION TO DATA MIGRATION

- Moves data into any application core object in IFS Application
- Reads a number of different data sources
  - Text files
  - Tables or Views in Oracle database
  - Other sources via database links
- Directly from Microsoft Excel using the IFS Data Migration Excel Add-In.
- Join information from already migrated data along with external sources
- IFS Application PROD(product) components includes pre-packaged migration definitions



## MAIN FLOW IN DATA MIGRATION



IFS

## BASIC DATA

### IFS DATA MIGRATION

- Procedures
  - Standard procedures for different types of job
- Default Rules
  - Common settings for job rules
- Groups
  - Categorize your jobs
- Users
  - Set user privileges and job access
- Conversion lists
  - Data values from the source may be changed



## BASIC DATA PROCEDURES

- Standard procedures for different kind of jobs
- All procedures are defined as public methods in LU [IntfaceUtil](#)
- If customized procedures are developed in [FNDMIG](#), they can be created in any LU, but must appear in [IntfaceUtil](#) as public methods



# BASIC DATA

## PROCEDURES

Data Migration Basic Data - Procedures

| Procedure Name         | Description                                | Direction        |
|------------------------|--|------------------|
| CHECK_BY_METHOD_NEW    | Insert rows from file                      | File In          |
| CREATE_OUTPUT_FILE     | Write data to file                         | File Out         |
| CREATE_TABLE_FROM_FILE | Create table from file                     | File In          |
| FNDMIG_EXPORT          | Export all Migration definitions to file   | File Out         |
| FNDMIG_IMPORT          | Import all Migration definitions from file | File In          |
| INSERT_BY_METHOD_NEW   | Insert rows from file                      | File In          |
| INSERT_OR_UPDATE       | Insert or update rows from file            | File In          |
| MIGRATE_SOURCE_DATA    | Data migration from table/view             | Source Migration |
| REPLICATION            | Replication                                | Source Migration |
| REPPLIC_DATA_OUT       | Replicate Data Structures                  | ReplicStructure  |
| EXCEL_MIGRATION        | Data Migration from Excel file             | Excel Migration  |



## BASIC DATA

### DEFAULT RULES

- Rules are used to specify individual behaviors of migration jobs
- Rules may be connected to specified procedures, or general use (\*)
- Rules with procedure-name will be automatically inserted to the data migration jobs using this procedure
- Rules with \* as the procedure-name will only be inserted if the `RuleFlag='Active'`



**Hint :** Spend some time to decide default-settings for rules like  
IGNOREXEERROR, SAVEJOBDAYS, SKIPLINES, MAPDBCOL etc.

# BASIC DATA

## DEFAULT RULES

| Data Migration Basic Data - Default Rules |               |               |  |          |                  |
|---|---------------|---------------|--|----------|------------------|
|   | Procedures    | Default Rules | Group  | User     | Pt User          |
|   | KEEPTRIGGER   | REPLICATION   | Do NOT generate standard trigger code                        | Inactive | No value allowed |
|   | ADDCBID       | REPLICATION   | Create extra column detail for OBID                          | Active   | No value allowed |
|   | ULTRAPLUGER   | REPLICATION   | Execute replication with alternate user                      | Inactive | No value allowed |
|   | BUILDFLAGIF   | REPLICATION   | Build IF-statement for updatable columns                     | Active   | No value allowed |
|   | CLEANUP       | REPLICATION   | Clear errors details automatically                           | BEFORE   | Active           |
|   | CREATEDET     | REPLICATION   | Create new column details based on views in MethodList       | 3        | Active           |
|   | IGNOREERROR   | REPLICATION   | Continue procedure even if errors occur                      | Active   | No value allowed |
|   | KEEPTRIGGER   | REPLICATION   | Do NOT generate standard trigger code                        | Inactive | No value allowed |
|   | MAPDBCOL      | REPLICATION   | Remap ID1 client-default to DB-default                       | Inactive | No value allowed |
|   | HONULLVAL     | REPLICATION   | Do not include selected items with null-value in attr-string | Inactive | No value allowed |
|   | SAVEJOB DAYS  | REPLICATION   | Number of days to save job-history                           | 30       | Active           |
|   | SAVEJOBERR    | REPLICATION   | Save only errors in history log                              | Active   | No value allowed |
|   | ADDCBID       | MIGRATE%      | Create extra column detail for OBID                          | Active   | No value allowed |
|   | CHECKPOINT    | MIGRATE%      | Perform methods New and Modify with CHECK option             | Inactive | No value allowed |
|   | CONFIDE       | MIGRATE%      | Start this job first   | Inactive | Mandatory string |
|   | CREATEDET     | MIGRATE%      | Create new column details based on views in MethodList       | 1        | Active           |
|   | IGNOREERROR   | MIGRATE%      | Continue procedure even if errors occur                      | Active   | No value allowed |
|   | KEEFDYNAMPROC | MIGRATE%      | Do not DROP dynamic created procedures                       | Inactive | No value allowed |
|   | MAPDBCOL      | MIGRATE%      | Remap ID1 client-default to DB-default                       | Inactive | No value allowed |
|   | HONULLVAL     | MIGRATE%      | Do not include selected items with null-value in attr-string | Inactive | No value allowed |



**Hint :** Spend some time to decide default-settings for rules like IGNOREEXERROR, SAVEJOB DAYS, SKIPLINES, MAPDBCOL etc.

## BASIC DATA GROUPS

- Makes queries and export/import easier
- Naming convention is recommended
- Hide-flag is used to prevent jobs from appearing in overview or maintenance forms



# BASIC DATA GROUPS

## Data Migration Basic Data - Group

|  | Procedures           | Default Rules              | Group                    | User | Pr User | Conversion Lists |
|--|----------------------|----------------------------|--------------------------|------|---------|------------------|
|  | MIG_UTIL             | Utility jobs               | <input type="checkbox"/> |      |         |                  |
|  | MIG_APPLICATION_USER | Migrate Application Use... | <input type="checkbox"/> |      |         |                  |
|  | MIG_EMPLOYEE         | Migrate Employee Infor...  | <input type="checkbox"/> |      |         |                  |
|  | MIG_CUSTOMER         | Migrate Customer Infor...  | <input type="checkbox"/> |      |         |                  |
|  |                      |                            |                          |      |         |                  |



## BASIC DATA

# USERS

- Users are defined and given privileges
- NoUpdateAllowed means only access to execute jobs; no maintenance
- UpdateAllowed means access to maintain all jobs in Data Migration
- From the Execute Job window, users can only start jobs they are connected to
- Application owner has always access to all jobs



# BASIC DATA

## USERS

| Data Migration Basic Data - User |                 |
|----------------------------------|-----------------|
| Procedures                       | Default Rules   |
| + Identity                       | Privilege       |
| ▶ IFSAPP                         | UpdateAllowed   |
| RUFELK                           | NoUpdateAllowed |

| Data Migration Basic Data - Pr User |  |
|-------------------------------------|--|
| Procedures                          | Default Rules  |
| Identity:                           | <input type="text" value="TFSAPP"/> <input type="button" value="▼"/> |
| Job ID                              | <input checked="" type="checkbox"/> CREATE_INPUT_FILE_DEF            |
|                                     | <input type="checkbox"/> EXPORT_MIG_APPLICATION_USER                 |
|                                     | <input type="checkbox"/> EXPORT_MIG_CUSTOMER                         |
|                                     | <input type="checkbox"/> EXPORT_MIG_EMPLOYEE                         |
|                                     | <input type="checkbox"/> EXPORT_MIG_EQUIPMENT_OBJECT                 |
|                                     | <input type="checkbox"/> EXPORT_MIG_FINANCIAL                        |
|                                     | <input type="checkbox"/> EXPORT_MIG_FND_USER                         |
|                                     | <input type="checkbox"/> EXPORT_MIG_INVENTORY_PART                   |
|                                     | <input type="checkbox"/> EXPORT_MIG_INVENT_LLOCATION                 |
|                                     | <input type="checkbox"/> EXPORT_MIG_MFO_STANDARDS                    |
|                                     | <input type="checkbox"/> EXPORT_MIG_PARTS                            |
|                                     | <input type="checkbox"/> EXPORT_MIG_PRICELIST                        |
|                                     | <input type="checkbox"/> EXPORT_MIG_SUPPLIER                         |
|                                     | <input type="checkbox"/> EXPORT_MIG_UTIL                             |
|                                     | <input type="checkbox"/> FNDMIG_EXPORT                               |
|                                     | <input type="checkbox"/> FNDMIG_EXPORT_SERVER                        |
|                                     | <input type="checkbox"/> FNDMIG_IMPORT                               |
|                                     | <input type="checkbox"/> FNDMIG_IMPORT_SERVER                        |

## BASIC DATA CONVERSION LISTS

- Used to change values from input source
- Wildcards may be used in Old Value, example : AB%, A%100 , A%, %
- One list may be connected to multiple jobs (Job Maintenance/FileMapping or SourceMapping)



# BASIC DATA

## CONVERSION LISTS

Data Migration Basic Data - Conversion Lists 

Procedures Default Rules Group User Pr User Conversion Lists

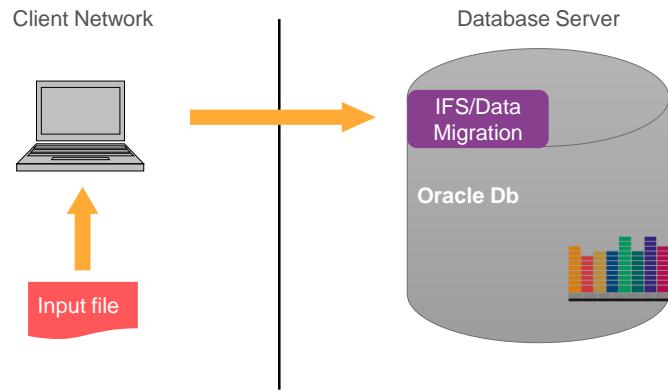
Conversion List ID: UOM Description: Unit of Measures

|   | Old Value | New Value |
|---|-----------|-----------|
| 1 | pcs       |           |
| 2 | l         |           |
| 3 | kg        |           |
| 4 | m         |           |

IES

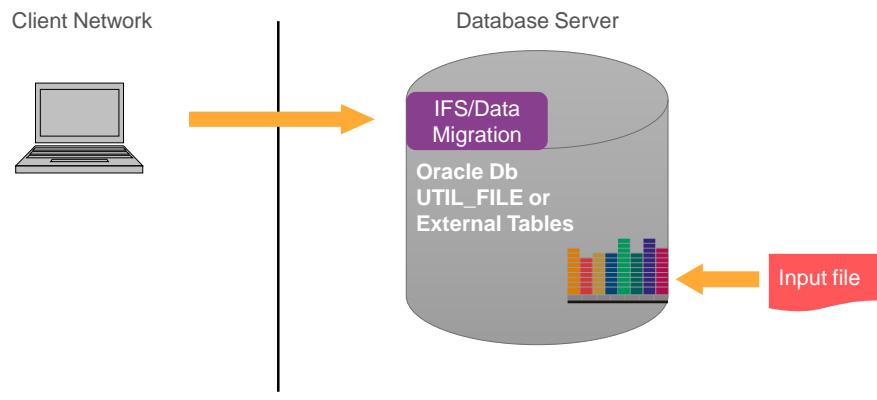
# DATA MIGRATION OVERVIEW

## FILE LOCATION ON CLIENT



IFS

## FILE LOCATION ON SERVER





## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



## FILE MIGRATION

### LESSON OBJECTIVES

By the end of this lesson, students should know:

- An overview of inserting or updating data from a file to a logical unit in IFS Applications
- How to configure the migration job for inserting or updating data from a file.
- The different migration parameters for inserting or updating data from a file.

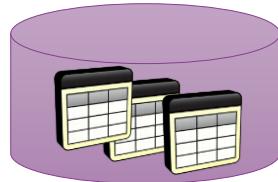
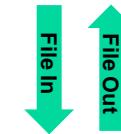


# WHAT IS FILE MIGRATION?

## OVERVIEW

- Transfers data between the database and files.
- Data can be imported to
  - a single database logical unit or
  - A data migration container table (IC Table)
- Export the data from multiple logical units to a single file.

| PART. NO. | DESCRIPTION                       | UNIT | CO. |
|-----------|-----------------------------------|------|-----|
| 1799      | Aobby lock higk1                  | 1    | pcs |
| 1819      | Garage door handle Belucci design |      |     |
| 1749      | EasyFix bent rail 4,5 m           | 3    | pcs |
| 1609      | Garage door base engine 200W      |      |     |
| 4.pcs     |                                   |      |     |
| 1679      | Power supply 240V                 | 5    | pcs |
| 1700      | Connector kit United Kingdom      |      |     |
| 6.pcs     |                                   |      |     |
| 1659      | Remote control Radio              | 7    | pcs |
| 1609      | Garage door base engine 200W      |      |     |
| 4.pcs     |                                   |      |     |
| 1679      | Power supply                      |      |     |
| 1770      | Connector kit United Kingdom      |      |     |
| 6.pcs     |                                   |      |     |
| 1659      | Remote control Radio              | 7    | pcs |
| 1609      | Garage door base engine 200W      |      |     |
| 4.pcs     |                                   |      |     |



IFS

## CREATE A FILE MIGRATION JOB

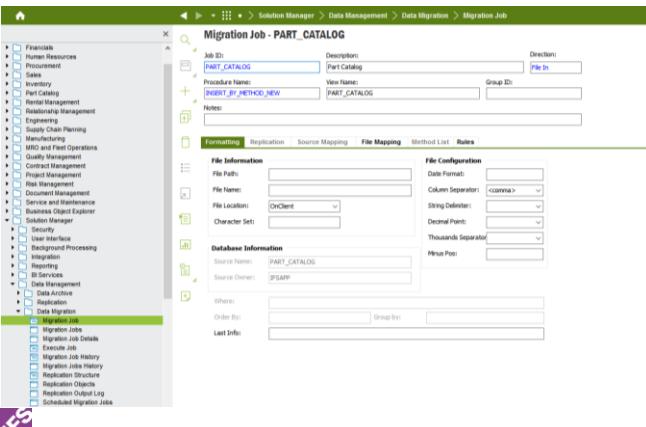
### FILE MIGRATION

- Create Data Migration Job "header"
- Point at text file containing data to be migrated
- Map source columns to destination columns
- Run Data migration job



# CREATE DATA MIGRATION JOB "HEADER"

## SET FILE LOCATION AND DELIMITER



1. Create a Migration Job by (Navigator: Data Migration | Migration Job Maintenance)
  - Enter description
  - Enter Procedure Name "INSERT\_BY\_METHOD\_NEW"
  - Enter View Name of the LU that you want to insert data into (tips use LOV) Method <view\_name>\_API.New\_\_ will be executed later on
  - Enter a descriptive Note Text
2. Point at textfile containing data to be migrated
  - Select File Location "On Client"
  - Select column separator <comma>

# MAP FILE COLUMNS WITH POSITION

## FILE MIGRATION

The screenshot shows the SAP Solution Manager interface for a Data Migration job named 'Migration Job - PART\_CATALOG'. The 'File Mapping' tab is selected. The 'File Mapping' table lists columns from a source file and their corresponding destination fields in the database:

| Pos | Column Name   | Description         | File | Data Type | Length | Decimal Length | Default Value | Change Defaults | Amount Factor | Default Where | Pad Char Right | Pad Char Left | Alt Seq. | Notes |
|-----|---------------|---------------------|------|-----------|--------|----------------|---------------|-----------------|---------------|---------------|----------------|---------------|----------|-------|
| 10  | INPUT_ID      | Part No             | K    | VARCHAR2  | 25     |                |               | Not Allowed     |               |               |                |               | 10       |       |
| 20  | DESCRIPTION   | Description         | W    | VARCHAR2  | 200    |                |               | Not Allowed     |               |               |                |               | 20       |       |
| 30  | INFO_CODE     | Code                | W    | VARCHAR2  | 30     |                |               | Not Allowed     |               |               |                |               | 50       |       |
| 40  | STD_NAME_ID   | Std Name ID         | W    | NUMBER    | 10     | 0              | 0             | Not Allowed     |               |               |                |               | 40       |       |
| 50  | INFO_TEXT     | Info Text           | S    | VARCHAR2  | 2000   |                |               | Not Allowed     |               |               |                |               | 30       |       |
| 60  | LOT_TRACKN    | LotBatch Tracking   | W    | VARCHAR2  | 4000   |                |               | Not Allowed     |               |               |                |               | 60       |       |
| 70  | LOT_TRACKN    | LotBatch Tracking   | W    | VARCHAR2  | 30     |                |               | Not Allowed     |               |               |                |               | 70       |       |
| 80  | SERIAL_RULE   | Serial Rule         | W    | VARCHAR2  | 4000   |                |               | Not Allowed     |               |               |                |               | 80       |       |
| 90  | SERIAL_RULE_  | Serial Rule         | W    | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |               | 90       |       |
| 100 | SERIAL_TRACK  | Serial Tracking     | W    | VARCHAR2  | 4000   |                |               | Not Allowed     |               |               |                |               | 100      |       |
| 110 | SERIAL_TRACK  | Serial Tracking     | W    | VARCHAR2  | 30     |                |               | Not Allowed     |               |               |                |               | 110      |       |
| 120 | END_SERIAL_T  | After Delivery Ser. | W    | VARCHAR2  | 4000   |                |               | Not Allowed     |               |               |                |               | 120      |       |
| 130 | END_SERIAL_T_ | After Delivery Ser. | W    | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |               | 130      |       |
| 140 | INPUT_MAN_ID  | Part Info Group     | S    | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |               | 140      |       |
| 150 | CONFIRMAPBLE  | Confirmapble        | W    | VARCHAR2  | 4000   |                |               | Not Allowed     |               |               |                |               | 150      |       |
| 160 | CONFIRMAPBLE  | Confirmapble        | W    | VARCHAR2  | 70     |                |               | Not Allowed     |               |               |                |               | 160      |       |

### 1. Map source columns into destination columns

- Select tab File Mapping.
- Declare file layout by entering values in Pos for each column. (As we are here using separators the POS column indicates the sequence of the columns from the text file)
- (In addition to column values from the file, you may add default-value to columns with Pos=0. These columns will also be added to the attr-string.)
- It is here possible to add default values and data conversion rules, but we'll come back to this
- TIP1: The Std\_Name\_ID requires a value, but no value exists in the source data file. It is possible to add a default value e.g. "0" or the id of a proper standard name
- TIP2: DataMigration will only reads column values up to specified length, and ignores data exceeding this length.

# RUN DATA MIGRATION JOB

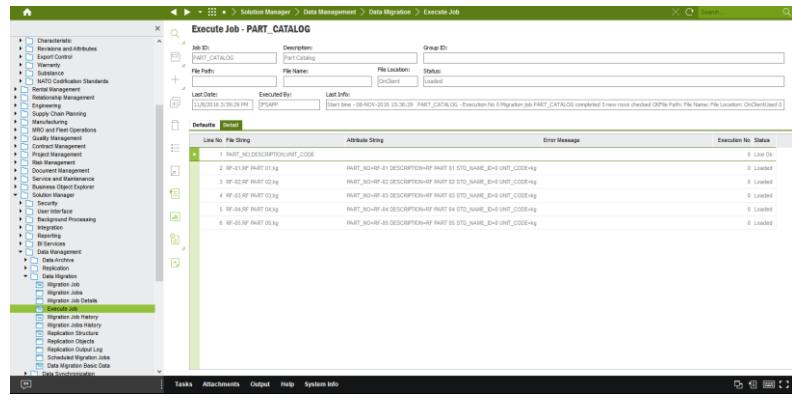
## FILE MIGRATION



1. Go to the navigator Item "Execute Job" and query for the Data Migration Job you just created
2. Then go to the tab 'Detail and run the Load File... menu selection in order to browse for the csv file on your computer

# RUN DATA MIGRATION JOB

## FILE MIGRATION



- When the file is loaded the column File String will be filled in with data. Remember to save the result.
- In order check the layout and default values you specified under File Mapping, you may now run the menu selection called "Unpack File" (In the header of the form) This function will build an Attribute string and produce error-messages if there are format conflicts (date format, characters in number columns etc). This is for debug purpose only.
- When you have verified that the file-layout is according to your specifications ,you run the "Start Online" menu selection to really start the migration.
- Your data will now be migrated into the selected LU, and an message box will display number of successful or error rows. If no errors occur, then all data will disappear in the Detail form. Otherwise you will have a number of error-records remaining. These records have not made it all the way to the LU, and you have to investigate the error message in order to find out what to do.

## EXERCISE 1 – SINGLE LU MIGRATION

### FILE MIGRATION

- You are going to create a new Data Migration Job in order to migrate data from the file "Part.csv" into the PartCatalog LU.
- When successful you will be able to view the result of this exercise by querying the Overview Part Catalog form.
- HINT: Use the Procedure Name: INSERT\_BY\_METHOD\_NEW



In this exercise we are going to run a very simple example. We have a csv file containing rudimentary parts data. The purpose is to use IFS Data Migration to insert these data into the Part Catalog LU.

Before you start migrating it might be a good idea to let everyone in the class prefix the part\_nos on the file in order to make the parts unique. (ex. U1xxxx,U2xxxx,U3xxxx for User1, User2, User3 and so on)

Tip: The first line in the file contains the name of the columns and we would like to skip this line. Go to the Rules tab and add the rule "SKIPLINES" with the value = 1

# HOW DOES IT WORK?

## FILE MIGRATION

- Procedure Insert\_By\_Method\_New
- Builds attribute string from file and default-values
- Completes attribute-string with values from New\_\_/PREPARE
- Will fail if duplicates occurs

Data from file + defaults

New/PREPARE

Attr-string

New\_\_

IFS

## EXERCISE 2 – RE-RUN THE SINGLE LU MIGRATION

### FILE MIGRATION

- Take your Data Migration Job that was created in Exercise 1 and try and re-run it. See what happens.
- Modify your Data Migration job to use the Procedure Name INSERT\_OR\_UPDATE. Run the job again and see what happens.
- Modify some of the data in the Part.csv file then re-run the jobs. Confirm that the Part Catalog records have been updated.



In this exercise we are going to run a very simple example. We have a csv file containing rudimentary parts data. The purpose is to use IFS Data Migration to insert these data into the Part Catalog LU.

Before you start migrating it might be a good idea to let everyone in the class prefix the part\_nos on the file in order to make the parts unique. (ex. U1xxxx,U2xxxx,U3xxxx for User1, User2, User3 and so on)

Tip: The first line in the file contains the name of the columns and we would like to skip this line. Go to the Rules tab and add the rule "SKIPLINES" with the value = 1

# FILE LOCATION ON SERVER

- Files read using Oracle UTL-FILE utility.
- Create the Oracle directory and give grants
  - `CREATE OR REPLACE DIRECTORY FNDMIG AS '<Path_in_DB_Server>'`
  - `GRANT READ,WRITE ON DIRECTORY FNDMIG TO '<APPLICATION_OWNER>'`
- File name is required.
- Character set can be converted to database character set

File Information

|                |                 |
|----------------|-----------------|
| File Path:     | FNDMID          |
| File Name:     | PartCatalog.txt |
| File Location: | OnServer        |
| Character Set: | [empty]         |



- The file *Character Set* of the file must be specified unless it is the same as the database character set.
- The *Character Set* column has a List-Of-Value with all character sets supported by the Oracle database.
- Some multi bytes character sets will not work properly
- UTF-8 files should also be correctly encoded

## EXERCISE 3 – FILE LOCATION – ON SERVER

### FILE MIGRATION

- Create a new oracle directory called “FNDMIG”
- Grant read and write permission to IFSAPP
- Copy the same data file used in the exercise 02 to Oracle directory “FNDMIG”
- Change the “File Location” property of the data migration job created in exercise 02 to “OnServer”
- Run the job again and see what happens



## CONVERSION LISTS

- A conversion list allows you to create a lookup table.
- More than one conversion list can be created.
- Conversion list set against a column in your Migration job.
- Setup and maintained in the Data Migration Basic Data screen.



## ADDING A CONVERSION LIST

- Define conversion lists and attach to columns
- Navigate to Application Base Data>Data Migration>Data Migration Basic Data

The screenshot shows a software interface titled "Data Migration Basic Data - Conversion Lists". At the top, there are tabs: Procedures, Default Rules, Group, User, Pr User, and Conversion Lists (which is highlighted). Below the tabs, there is a search bar and a table for defining conversion rules. The table has columns for "Old Value" and "New Value". Three entries are listed:

| Old Value | New Value |
|-----------|-----------|
| PCS       | pcs       |
| METER     | m         |
| LTR       | l         |



It is possible to create a number of conversion lists.

The list can be connected to a column. In the above example the unit measure PCS will be translated into pcs, LTR to l and METRE to m.

One conversion List may be used from many migration jobs.

Note that wildcards may be used. Eg by putting "42%" in the Old Value column and "HEY PRESTO" in the New Value column all values starting at 42 will be replaced by "HEY PRESTO"

# USING A CONVERSION LIST

- In your Data Migration Job, set the Conversion List ID column to the conversion list you want to use.

Migration Job - PART\_CATALOG\_INSERT\_UPDATE

Job ID: PART\_CATALOG\_INSERT\_UPDATE Description: Part Catalog Insert or Update Direction: File In

Procedure Name: INSERT\_OR\_UPDATE View Name: PART\_CATALOG Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Pos | Column Name       | Description        | Flags | Data Type | Length | Decimal Length | Default Value | Conversion List ID | Change Defaults |
|-----|-------------------|--------------------|-------|-----------|--------|----------------|---------------|--------------------|-----------------|
| 10  | PART_NO           | Part No            | K     | VARCHAR2  | 25     |                |               | Not Allowed        |                 |
| 20  | DESCRIPTION       | Description        | MU    | VARCHAR2  | 200    |                |               | Not Allowed        |                 |
| 30  | UNIT_CODE         | UoM                | MU    | VARCHAR2  | 30     |                | UNIT_MEASURES | Not Allowed        |                 |
| 0   | STD_NAME_ID       | Std Name ID        | MU    | NUMBER    | 10     | 0              | 0             | Not Allowed        |                 |
| 0   | INFO_TEXT         | Info Text          | IU    | VARCHAR2  | 2000   |                |               | Not Allowed        |                 |
| 0   | LOT_TRACKING_CODE | Lot/Batch Tracking | MU    | VARCHAR2  | 4000   |                |               | Not Allowed        |                 |



## EXERCISE 4

### CONVERSION LIST

- Use a conversion list in the parts migration job created earlier
- Open Parts2.csv file
- Check the data values given in the “Unit of Measure” column
- Create a new conversion list to use as a look up table for the “Unit of Measure” column
- Modify your parts migration job to use the new conversion list



# CREATING AN OUTPUT FILE

## OVERVIEW

- Set up in same manner as other file jobs
- Enhanced functionality to specify multiple data sources and conditions
- JOINS
- WHERE, ORDER BY and GROUP BY clauses
- Select list may contain SQL-expressions
- Server file and backup options
- Same scheduling options as other data migration jobs



# CREATING AN OUTPUT FILE

## FORMATTING

Migration Job - EXPORT\_BRANCH\_DETAILS

Job ID: REPORT\_BRANCH\_DETAILS | Description: Export Branch Details | Direction: File Out

Procedure Name: CREATE\_OUTPUT\_FILE | View Name: TRN\_BRANCH | Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

**File Information**

File Path: FNDBMSG  
File Name: BranchDetails.csv  
File Location: OrderServer  
Character Set: AL32UTF8

**File Configuration**

Date Format:   
Column Separator: <comma>  
String Delimiter:   
Decimal Point:   
Thousands Separator:   
Minus Pos:

**Database Information**

Source Name: TRN\_BRANCH\_B,TRN\_AREA\_A  
Source Owner: IFSAAPP

Where: B.COUNTRY\_ID = A.COUNTRY\_ID AND B.REGION\_ID = A.REGION\_ID  
Order By:  Group by:   
Last Info:

1. Create a new Data Migration job setting the Procedure Name to CREATE\_OUTPUT\_FILE.
2. Set the View Name to the source of your data.
3. Set the File Location and column separator.

# CREATING AN OUTPUT FILE

## FILE MAPPING

Migration Job - EXPORT\_BRANCH\_DETAILS

Job ID: EXPORT\_BRANCH\_DETAILS Description: Export Branch Details Direction: File Out

Procedure Name: CREATE\_OUTPUT\_FILE View Name: TRN\_BRANCH Group ID:

Notes:

Formatting Replication Source Mapping **File Mapping** Method List Rules

| Pos | Column Name  | Description | Flags | Data Type | Length | Decimal Length | Default Value | Change Defaults | Amount Factor | Default Where | Pad Char Right |
|-----|--------------|-------------|-------|-----------|--------|----------------|---------------|-----------------|---------------|---------------|----------------|
| 10  | COMPANY_ID   | Company ID  | P     | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |
| 20  | BRANCH_ID    | Branch ID   | K     | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |
| 30  | TIME_OFFSET  | Time Offset | MU    | NUMBER    | 20     |                |               | Not Allowed     |               |               |                |
| 40  | A_COUNTRY_ID | Country ID  | IU    | VARCHAR2  | 10     |                |               | Not Allowed     |               |               |                |
| 50  | A_AREA_ID    | Area ID     | IU    | VARCHAR2  | 20     |                |               | Not Allowed     |               |               |                |
| 60  | AREA         | Area        |       | VARCHAR2  | 100    |                |               | Not Allowed     |               |               |                |

1. Set the columns and the order they are output to the file by setting the value in the Pos column.
2. Not all columns in the TRN\_BRANCH view need to be output.
3. To exclude columns from being written to the file, set Pos value as 0

# CREATING AN OUTPUT FILE

## RULES

Migration Job - EXPORT\_BRANCH\_DETAILS

Job ID: EXPORT\_BRANCH\_DETAILS Description: Export Branch Details Direction: File Out

Procedure Name: CREATE\_OUTPUT\_FILE View Name: TRN\_BRANCH Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Rule ID     | Rule Description                                    | Rule Value | Rule Flag |
|-------------|---|------------|-----------|
| CALLEXTPROC | Call external procedure after each row              |            | Inactive  |
| FILEHEAD    | Create fileheader with column names. CSV-files only |            | Active    |
| FILEBKP     | Backup file after successful completion             |            | Active    |

IFS

Activate any rules against the output job,

e.g.

FILEHEAD to output header information

FILEBKP to backup any previously created output(data) file before creating the new data file

# CREATING AN OUTPUT FILE

## EXECUTING JOB

The screenshot shows the 'Execute Job - EXPORT\_BRANCH\_DETAILS' screen. At the top, there are fields for 'Job ID' (EXPORT\_BRANCH\_DETAILS), 'Description' (Export Branch Details), and 'Group ID'. Below these are fields for 'File Path' (D:\), 'File Name' (BranchDetails.csv), 'File Location' (OnClient), and 'Status' (Line Ok). Under 'Last Date' (11/8/2016 5:51:44 PM) and 'Executed By' (IFSAPP), the 'Last Info' shows 'Start time - 08-NOV-2016 17:51:44' and 'Job EXPORT\_BRANCH\_DETAILS compl'. A table below lists six records with columns: Line No, File String, Attribute String, Error Message, Execution No, and Status. The first record is highlighted in green. A context menu is open over the table, showing options like 'Start Online', 'Restart Online', 'Schedule Job', 'Unpack File', 'Clean Up', 'Migration Job...', 'Populate', 'Search in Context...', 'Edit', 'Subscriptions', and 'Conditional Fields...'. The 'Start Online' option is highlighted.

1. To execute the job, RMB and select the Start Online option.

If the data file location = OnServer

- It creates the file directly inside the Oracle directory specified in the formatting window

If the data file location = OnClient

- The Detail tab is populated with the output data.
- RMB and select the Export to File option to write this information to a file.

## EXERCISE 5

### OUTPUT FILE

- Create an output file data migration job to export the data of the TrnBranch LU to a comma delimited file with field headers as show in the below picture

|    | A          | B                        | C         | D          | E       | F          | G           | H |
|----|------------|--------------------------|-----------|------------|---------|------------|-------------|---|
| 1  | COMPANY_ID | COMPANY_NAME             | BRANCH_ID | COUNTRY_ID | AREA_ID | AREA_NAME  | TIME_OFFSET |   |
| 2  | TCS        | Teknoids Computer Spares | BAN-01    | IN         | BAN     | Bangalore  | 6           |   |
| 3  | TCS        | Teknoids Computer Spares | CHE-01    | IN         | CHE     | Chennai    | 6           |   |
| 4  | TCS        | Teknoids Computer Spares | COL-01    | LK         | COL     | Colombo    | 6           |   |
| 5  | TCS        | Teknoids Computer Spares | DEL-01    | IN         | DEL     | New Delhi  | 6           |   |
| 6  | TCS        | Teknoids Computer Spares | GAL-01    | LK         | GAL     | Galle      | 6           |   |
| 7  | TCS        | Teknoids Computer Spares | GBG-01    | SE         | GBG     | Gothenburg | 2           |   |
| 8  | TCS        | Teknoids Computer Spares | KDY-01    | LK         | KDY     | Kandy      | 6           |   |
| 9  | TCS        | Teknoids Computer Spares | LKP-01    | SE         | LKP     | Linköping  | 2           |   |
| 10 | TCS        | Teknoids Computer Spares | S         | IN         | BAN     | Bangalore  | 3           |   |
| 11 | TCS        | Teknoids Computer Spares | we        | SE         | GTB     | Gäteborg   | 4           |   |
| 12 | TCS        | Teknoids Computer Spares | zaz       | SE         | GBG     | Gothenburg | 5           |   |
| 13 |            |                          |           |            |         |            |             |   |

IFS

# HISTORY LOGGING

- For migration Jobs with SAVEJOBDAYS=Active

**Migration Jobs**

| Job ID                     | Data Migration Job Description | Group ID | Group Description | File Path | File Name | Date Format | Minus Pos | Where |
|----------------------------|--------------------------------|----------|-------------------|-----------|-----------|-------------|-----------|-------|
| PART_CATALOG_INSERT_UPDATE | Part Catalog Insert or Update  |          |                   |           |           |             |           |       |

Job Details...  
Start Job...  
**Job History Overview...**  
Print Documentation...

**Migration Jobs History**

| Job ID                     | Description                   | Execution No           | Last Info   | Executed By | Executed At |
|----------------------------|-------------------------------|------------------------|---|-------------|-------------|
| PART_CATALOG_INSERT_UPDATE | Part Catalog Insert or Update | 2                      | Start time - 08-NOV-2016 18:23:11 PART_CA... IFSAPP   | IFSAPP      | 11/8/2016   |
| PART_CATALOG_INSERT_UPDATE |                               | Job History Details... | 1 Start time - 08-NOV-2016 18:22:49 PART_CA... IFSAPP | IFSAPP      | 11/8/2016   |
|                            |                               | Populate               | 0 Start time - 08-NOV-2016 16:28:17 PART_CA... IFSAPP | IFSAPP      | 11/8/2016   |



## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



### 3. SOURCE MIGRATION

# SOURCE MIGRATION

## LESSON OBJECTIVES

By the end of this lesson, students should know:

- Migrate data from one data source to another.
- The source can be a TABLE, VIEW, JOIN or DB LINK, while destination can be a single or multiple logical units in the database.



## TWO STAGE MIGRATION JOB

### SOURCE MIGRATION

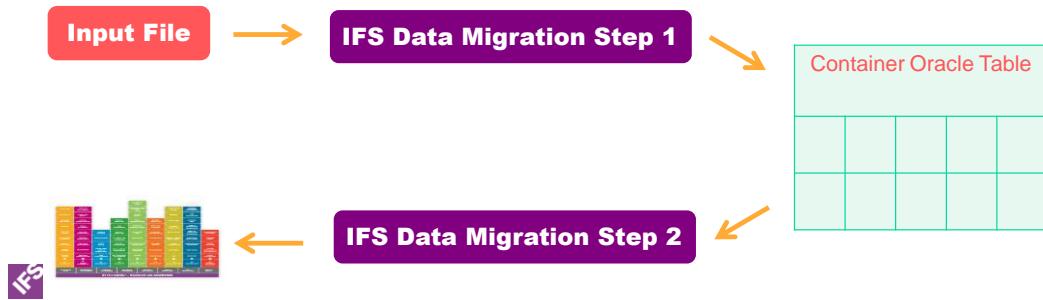
- Job run in 2 stages:
  - Create 1st job to load data from CSV file into a temporary database table.
  - Create 2nd job to move data from temporary table into IFS tables.
- Why do a 2 stage migration?
  - Greater control on how you select the information from the source table.
  - Allows you to load data in multiple Logical Units.



## MIGRATING DATA INTO MULTIPLE LU'S

### SOURCE MIGRATION

- Data is transferred into a container table in the database
- This table acts as the source for a second migration, which updates IFS Applications



# CREATE 1<sup>ST</sup> MIGRATION JOB - FORMATTING

## SOURCE MIGRATION

Migration Job - LOAD\_USER\_DATA\_TO\_TEMP

Job ID: LOAD\_USER\_DATA\_TO\_TEMP Description: Load FndUser data to temp table Direction: File In

Procedure Name: CREATE\_TABLE\_FROM\_FILE View Name: Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

**File Information**

|                |                 |
|----------------|-----------------|
| File Path:     | D:\             |
| File Name:     | FndUserData.csv |
| File Location: | On Client       |
| Character Set: |                 |

**File Configuration**

|                      |         |
|----------------------|---------|
| Date Format:         |         |
| Column Separator:    | <comma> |
| String Delimiter:    |         |
| Decimal Point:       |         |
| Thousands Separator: |         |
| Minus Sign:          |         |

**Database Information**

|               |  |
|---------------|--|
| Source Name:  |  |
| Source Owner: |  |

**Where:**

Order By: [group by] Last Info:



# 1<sup>ST</sup> MIGRATION JOB – SET FILE MAPPINGS

## SOURCE MIGRATION

Migration Job - LOAD\_USER\_DATA\_TO\_TEMP

Job ID: LOAD\_USER\_DATA\_TO\_TEMP Description: Load FndUser data to temp table Direction: File In

Procedure Name: CREATE\_TABLE\_FROM\_FILE View Name: Group ID:

Notes:

Formatting Replication Source Mapping **File Mapping** Method List Rules

| Pos | Column Name | Description  | Flags | Data Type | Length | Decimal Length | Default Value | Conversion List ID | Change Defaults | Amount Factor |
|-----|-------------|--------------|-------|-----------|--------|----------------|---------------|--------------------|-----------------|---------------|
| 10  | IDENTITY    | ID           | K     | VARCHAR2  | 30     |                |               |                    | NotAllowed      |               |
| 20  | DESCRIPTION | Description  | MU    | VARCHAR2  | 2000   |                |               |                    | NotAllowed      |               |
| 30  | ORACLE_USER | Oracle User  | MU    | VARCHAR2  | 30     |                |               |                    | NotAllowed      |               |
| 40  | WEB_USER    | Directory ID | IU    | VARCHAR2  | 2000   |                |               |                    | NotAllowed      |               |
| 50  | ACTIVE      | Active       | MU    | VARCHAR2  | 5      |                |               |                    | NotAllowed      |               |
| 60  | VALID_FROM  | Valid From   | IU    | DATE      | 21     |                |               |                    | NotAllowed      |               |
| 70  | VALID_TO    | Valid To     | IU    | DATE      | 21     |                |               |                    | NotAllowed      |               |
| 80  | NAME        | Name         | K     | VARCHAR2  | 50     |                |               |                    | NotAllowed      |               |
| 90  | VALUE       | Value        | MU    | VARCHAR2  | 250    |                |               |                    | NotAllowed      |               |

The View Name can be used to fill in some of the Column Names as a starting point and can be deleted afterwards

# 1<sup>ST</sup> MIGRATION JOB – SET RULES

## SOURCE MIGRATION

Migration Job - LOAD\_USER\_DATA\_TO\_TEMP

Job ID: LOAD\_USER\_DATA\_TO\_TEMP Description: Load FndUser data to temp table Direction: File In

Procedure Name: CREATE\_TABLE\_FROM\_FILE View Name: Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Rule ID       | Rule Description                                       | Rule Value | Rule Flag |
|---------------|--|------------|-----------|
| IGNOREXEERROR | Continue procedure even if errors occur                | Active     |           |
| REMLOG        | Remove log file if ExternalTable was created OK        | Active     |           |
| SKPLINES      | Start reading file after this line number              | 1          | Active    |
| CREINDEX      | Create index for key columns                           |            | Inactive  |
| CRETABCONF    | Options for container table                            |            | Inactive  |
| EXTTABLE      | Create ExternalTable and insert directly into IC-table |            | Inactive  |
| SYNCDETAILS   | Synchronize job details with columns in file-header    |            | Inactive  |

# 1<sup>ST</sup> MIGRATION JOB – EXECUTE JOB

## SOURCE MIGRATION

Execute Job - LOAD\_USER\_DATA\_TO\_TEMP

Description: Load Prod User data to temp table

File Path: File Name: ProdUserDetails.csv

Last Date: Executed By: Last Info:

Status: Pending

Group ID: Procedure Execution - IFS Applications

Information Message

Start time - 08-NOV-2016 19:20:19 LOAD\_USER\_DATA\_TO\_TEMP - Execution No 0

Created table IC\_LOAD\_USER\_DATA\_TO\_TEMP\_TAB (

IDENTITY VARCHAR(20),

DESCRIPTION VARCHAR(200),

ORACLE\_USER VARCHAR(30),

WEB\_USER VARCHAR(200),

ACTRIS VARCHAR(20),

VALID\_FROM\_DATE DATE,

VALID\_TO\_DATE DATE,

NAME VARCHAR(250),

VALUE VARCHAR(250),

IC\_ROW\_NO NUMBER)

TABLESPACE IFAPP\_DATA

47 rows selected

0 rows inserted

0 rows updated

0 rows failed

Used 0 hours and 0 minutes

Average of 2820 rows per minute

File Path: D:\ProdUserDetails.csv

File Name: ProdUserDetails.csv

File Location: OnClient

Used 0 hours and 0 minutes

End time - 08-NOV-2016 19:20:19 LOAD\_USER\_DATA\_TO\_TEMP

OK

Tasks Attachments Output Help System Info



# 1<sup>ST</sup> MIGRATION JOB – QUERY DATA

## SOURCE MIGRATION

SQL Query Tool

File select:

View: Columns:

SQL Statement:

```
SELECT * FROM IC_LOAD_USER_DATA_TO_TEMP_TAB
```

Help Execute Clear

| Identity | Description    | Oracle user | Web User | Active | Valid From | Valid To | Name                  | Value           | It Row No |
|----------|----------------|-------------|----------|--------|------------|----------|-----------------------|-----------------|-----------|
| ALCO_1   | Alice Cooper   | ALCO_1      | ALCO_1   | FALSE  |            |          | PRES_OBJECT_SEC_SETUP | SYSTEM-DEFINED  | 1         |
| ALCO_1   | Alice Cooper   | ALCO_1      | ALCO_1   | FALSE  |            |          | DEFAULT_PAPER_FORMAT  | SYSTEM-DEFINED  | 2         |
| ALCO_1   | Alice Cooper   | ALCO_1      | ALCO_1   | FALSE  |            |          | SMTP_MAIL_ADDRESS     | ALCO_1@venevane | 3         |
| ALCO_1   | Alice Cooper   | ALCO_1      | ALCO_1   | FALSE  |            |          | DEFAULTPLANT          | 1               | 4         |
| ARFL_1   | Artha Franklin | ARFL_1      | ARFL_1   | FALSE  |            |          | PRES_OBJECT_SEC_SETUP | SYSTEM-DEFINED  | 5         |
| ARFL_1   | Artha Franklin | ARFL_1      | ARFL_1   | FALSE  |            |          | DEFAULT_PAPER_FORMAT  | SYSTEM-DEFINED  | 6         |
| ARFL_1   | Artha Franklin | ARFL_1      | ARFL_1   | FALSE  |            |          | CUSTOMER_NO           | 6200            | 7         |
| ARFL_1   | Artha Franklin | ARFL_1      | ARFL_1   | FALSE  |            |          | CUSTOMER_ID           | 6200            | 8         |
| ARFL_1   | Artha Franklin | ARFL_1      | ARFL_1   | FALSE  |            |          | DEFAULTPLANT          | 1               | 9         |
| ALAR_1   | ALAR_1 Pres    | ALAR_1      | ALAR_1   | TRUE   |            |          | DEFAULT_PAPER_FORMAT  | SYSTEM-DEFINED  | 10        |
| ALAR_1   | ALAR_1 Pres    | ALAR_1      | ALAR_1   | TRUE   |            |          | CUSTOMER_ID           | 1000            | 11        |

IFS

## CREATE 2<sup>ND</sup> MIGRATION JOB - FORMATTING

### SOURCE MIGRATION

Migration Job - MIGRATE\_USER\_DATA - 2 (2)

Job ID: MIGRATE\_USER\_DATA Description: Migrate FndUser and FndUserProperty data Direction: Source Migration

Procedure Name: MIGRATE\_SOURCE\_DATA View Name: Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

**File Information**

|                |                      |
|----------------|----------------------|
| File Path:     | Date Format:         |
| File Name:     | Column Separator:    |
| File Location: | String Delimiter:    |
| Character Set: | Decimal Point:       |
|                | Thousands Separator: |
|                | Minus Pos:           |

**Database Information**

|               |                               |
|---------------|-------------------------------|
| Source Name:  | IC_LOAD_USER_DATA_TO_TEMP_TAB |
| Source Owner: | IFSAAPP                       |
| Where:        |                               |
| Order By:     | Group by:                     |
| Last Info:    |                               |

1. Create a Migration Job by (Navigator: Data Migration | Migration Job Maintenance)
  - Enter description
  - Enter Procedure Name "MIGRATE\_SOURCE\_DATA"
  - Enter a descriptive Note Text
2. In the Database Information section:
  - Enter the name of your source database table (HINT: it will be that name of your CREATE\_TABLE\_FROM\_FILE job prefix with IC\_ and a postfix of \_TAB)

## 2<sup>ND</sup> MIGRATION JOB – METHOD LIST

### SOURCE MIGRATION

Migration Job - MIGRATE\_USER\_DATA ▾ 2 (2)

Job ID: MIGRATE\_USER\_DATA Description: Migrate FndUser and FndUserProperty data Direction: Source Migration

Procedure Name: MIGRATE\_SOURCE\_DATA View Name: Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Execute Seq | View Name         | Action | Prefix Option | Method Name           | New                                 | Modify                              | Column Name | Column Value | On New Master            | On First Row             | Notes | Source Name |
|-------------|-------------------|--------|---------------|-----------------------|-------------------------------------|-------------------------------------|-------------|--------------|--------------------------|--------------------------|-------|-------------|
| 10          | FND_USER          | InLoop | ViewName      | FND_USER_API          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |              | <input type="checkbox"/> | <input type="checkbox"/> |       |             |
| 20          | FND_USER_PROPERTY | InLoop | ViewName      | FND_USER_PROPERTY_API | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |              | <input type="checkbox"/> | <input type="checkbox"/> |       |             |

IFS

1. Next setup your Method List. We skip Source Mapping for the moment because your method list will populate the Source Mapping information for you.

2. Identified IFS views we want to add data to:

FND\_USER

FND\_USER\_PROPERTY

1. NOTE: Need to set the Execute Seq. column to the order you want the data added, e.g. header information first followed by detail.

2. NOTE: If API names are consistent with the view names when the records are saved the API Method is filled in automatically.

## 2<sup>ND</sup> MIGRATION JOB – SOURCE MAPPING

### SOURCE MIGRATION

Migration Job - MIGRATE\_USER\_DATA 2 (2)

Job ID: MIGRATE\_USER\_DATA Description: Migrate FndUser and FndUserProperty data Direction: Source Migration

Procedure Name: MIGRATE\_SOURCE\_DATA View Name: Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

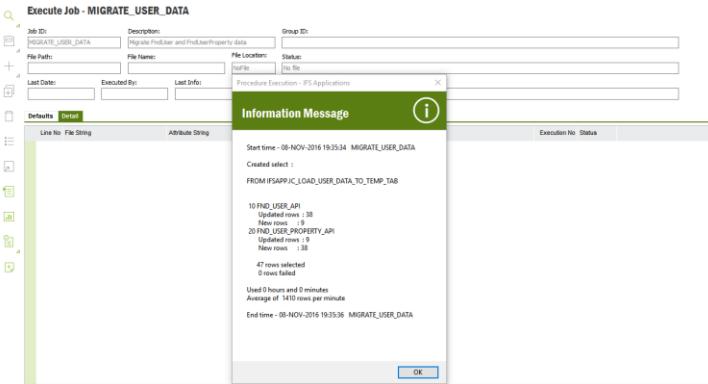
| Pos | Column Name             | Description | Flag | Data Type | Length | Source Column | Default Value | ID Values | Conversion List | Notes | Change Defaults                               |
|-----|-------------------------|-------------|------|-----------|--------|---------------|---------------|-----------|-----------------|-------|---|
| 10  | IDENTITY                |             | I    | VARCHAR2  | 30     | IDENTITY      |               |           |                 |       | Not Allowed                                   |
| 20  | FND_USER.DESCRIPTION    | Description | MU   | VARCHAR2  | 2000   | DESCRIPTION   |               |           |                 |       | Not Allowed                                   |
| 30  | FND_USER.ORACLE_USER    | Oracle User | MU   | VARCHAR2  | 30     | ORACLE_USER   |               |           |                 |       | Not Allowed                                   |
| 40  | FND_USER.WEB_USER       | Directory   | I    | VARCHAR2  | 2000   | WEB_USER      |               |           |                 |       | Not Allowed                                   |
| 50  | FND_USER.ACTIVE         | Active      | MU   | VARCHAR2  | 5      | ACTIVE        | 'TRUE'        |           |                 |       | Default value "TRUE" from PREPARE Not Allowed |
| 60  | FND_USER.VALID_FROM     | Valid From  | U    | DATE      | 21     | VALID_FROM    |               |           |                 |       | Not Allowed                                   |
| 70  | FND_USER.VALID_TO       | Valid To    | U    | DATE      | 21     | VALID_TO      |               |           |                 |       | Not Allowed                                   |
| 90  | NAME                    | Name        | K    | VARCHAR2  | 50     | NAME          |               |           |                 |       | Not Allowed                                   |
| 100 | FND_USER_PROPERTY.VALUE | Value       | U    | VARCHAR2  | 250    | VALUE         |               |           |                 |       | Not Allowed                                   |

IFS

1. Now you can see that Source Mapping has been populated automatically based upon the Method List selected.
2. It has tried to automatically map some of the columns in the source table with those in the target views if they are named the same. (OBJID can also be removed here)
3. Where possible it has also generated default values for mandatory columns.
4. Now you should be able to run the second migration job and transfer the data from the source file into IFS.

## 2<sup>ND</sup> MIGRATION JOB – EXECUTE JOB

### SOURCE MIGRATION



## EXERCISE 6

### TWO STAGE MIGRATION

- Create your first migration job that will load the contents of 'FndUserData.csv' into a temporary table.
- Create a second migration job to transfer the contents of the temporary table into IFS.
- Query back your FndUser records to confirm the data has loaded.





## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



## MIGRATE DATA WITH CUSTOM FIELDS

### LESSON OBJECTIVES

By the end of this lesson, students should know:

- How to migrate data when entities contain custom fields



## CUSTOM FIELDS

- Custom Fields provides the ability to add fields without doing any customization.
- This gives IFS customers an option to stay closer to the core product and follow IFS' upgrade path more easily.
- Use Custom Fields to add fields that are important to the customers business and is not part of IFS Applications core product.



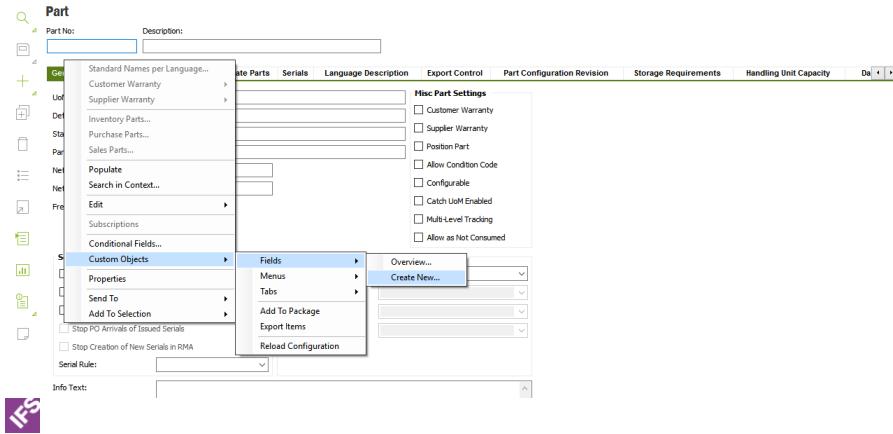
## BENEFITS OF CUSTOM FIELDS

- Add attributes to a Logical Unit without any coding
- Simple to demo and prototype
  - Easy to add Custom Fields during a demo
  - Easy to add Custom Fields if you are not really sure what you want
- Almost like a customization without coding
  - Can be used together with Custom Menus and Custom Defined Events to create more complex interactions
- No limitation of the number of Custom Fields



# CONFIGURE CUSTOM FIELDS

## ADDING A NEW CUSTOM FIELD



# CONFIGURE CUSTOM FIELDS

## ADDING A NEW CUSTOM FIELD

Custom Fields - PartCatalog

| Logical Unit:   | Prompt:   | Module:   |                                     |                                     |                  |   |         |                |        |           |
|---|---|-----------|-------------------------------------|-------------------------------------|------------------|---|---------|----------------|--------|-----------|
| PartCatalog   | Part Catalog  | PARTCA    |                                     |                                     |                  |   |         |                |        |           |
| <input checked="" type="checkbox"/> RowKey Activated  | Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs. |           |                                     |                                     |                  |   |         |                |        |           |
| <input checked="" type="checkbox"/> Approved  | Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.   |           |                                     |                                     |                  |   |         |                |        |           |
| <input checked="" type="checkbox"/> Published   | Published will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.                       |           |                                     |                                     |                  |   |         |                |        |           |
|   | Published Date: 11/8/2016 at 9:57:40 PM   |           |                                     |                                     |                  |   |         |                |        |           |
|   | Presentation Object: cPartCatalog   |           |                                     |                                     |                  |   |         |                |        |           |
| <input checked="" type="checkbox"/> Synchronized  | Synchronized means that no changes has been made to the configuration since it was last Published.  |           |                                     |                                     |                  |   |         |                |        |           |
| <a href="#">Attribute Name</a> <a href="#">Prompt</a> <a href="#">Approved</a> <a href="#">Published</a> <a href="#">Field Type</a> <a href="#">Comment</a> <a href="#">Translations</a> <a href="#">UI Object</a> <a href="#">LU Reference</a> <a href="#">LOV View</a> <a href="#">Custom Field Impl Type</a> <a href="#">Expression</a> <a href="#">Meta Data</a> <a href="#">Data Type</a> <a href="#">Format</a> |   |           |                                     |                                     |                  |   |         |                |        |           |
|   | GOVT_CODE   | Govt Code | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Persistent Field | 0 | Default | Not Applicable | String | Uppercase |



# CONFIGURE CUSTOM FIELDS

## CONFIGURING THE LAYOUT

Part

Part No:  Description:  Government Code:

General GTIN Part Manufacturer Alternate Parts Serials Language Description Export Control Part Configuration Revision Storage Re

UoM:   Default Input UoM Group:

Standard Name ID:   Part Main Group:

Net Weight:  Net Volume:  Freight Factor:

Misc Part Settings

- Customer Warranty
- Supplier Warranty
- Position Part
- Allow Condition Code
- Configurable
- Catch UoM Enabled
- Multi-Level Trading
- Allow as Not Consumed

Serial Tracking

- At Receipt and Issue
- In Inventory
- After Delivery Serial Tracking
- Stop PO Arrivals of Issued Serials
- Stop Creation of New Serials in RMA

Lot/Batch Tracking

Serial Rule:

Lot/Batch Trading:   
Lot Quantity Rule:   
Sub Lot Rule:   
Component Lot Rule:

IES

## MIGRATION SETPS

### CUSTOM FIELDS

- PartCatalog LU has a custom field GOVT\_CODE
- This GOVT\_CODE belongs to
  - PART\_CATALOG\_CFP – Custom Package
  - PART\_CATALOG\_CFT – Custom Table
- Migrating data for PART\_CATALOG\_TAB and PART\_CATALOG\_CFT simultaneously from a file
- Relationship between PART\_CATALOG\_TAB & PART\_CATALOG\_CFT is maintained by ROWKEY column
- Use PART\_CATALOG\_API and PART\_CATALOG\_CFP.Cf\_New\_



# MIGRATION SETPS

## CREATE TABLE FROM DATA FILE

### Migration Job - MIGRATE\_PARTS\_CUSTOM

Job ID:  Description:  Direction:

Procedure Name:  View Name:  Group ID:

Notes:

**File Mapping**

| Pos | Column Name | Description     | Flags | Data Type | Length | Decimal Length | Default Value | Conversion List ID | Change Defaults | Amount Factor | Default Where | Pa |
|-----|-------------|-----------------|-------|-----------|--------|----------------|---------------|--------------------|-----------------|---------------|---------------|----|
| 10  | PART_NO     | Part No         | K     | VARCHAR2  | 25     |                |               |                    | Not Allowed     |               |               |    |
| 20  | DESCRIPTION | Description     | MIU   | VARCHAR2  | 200    |                |               |                    | Not Allowed     |               |               |    |
| 30  | UNIT_CODE   | UoM             | MIU   | VARCHAR2  | 30     |                |               |                    | Not Allowed     |               |               |    |
| 40  | GOVT_CODE   | Government Code | IU    | VARCHAR2  | 5      |                |               |                    | Not Allowed     |               |               |    |



## MIGRATION SETPS

### MIGRATE SOURCE DATA

- PART\_CATALOG view for the normal data migration of the base table
- PART\_CATALOG\_CFP and PART\_CATALOG\_CFP.Cf\_New\_\_ are for migration of the custom field data
- The PART\_CATALOG\_CFP.Cf\_New\_\_ method needs to be invoked only if “DUMMY” column is not null [Condition]

The screenshot shows the 'Migration Job - MIGRATE\_PARTS\_IC' interface with two steps listed:

- Step 1: MIGRATE\_SOURCE\_DATA**
  - Job ID: MIGRATE\_PARTS\_IC
  - Description: Migrate Part Catalog from IC with Custom Fields
  - Direction: Source Migration
  - Procedure Name: MIGRATE\_SOURCE\_DATA
  - View Name:
  - Group ID:
  - Notes:
- Step 2: PART\_CATALOG**
  - Execute Seq: 10
  - Action: InLoop
  - ViewName: PART\_CATALOG\_API
  - Method Name: PART\_CATALOG\_CFP
  - New:
  - Modify:
  - Column Name: Column Value
  - On New Master: On

## MIGRATION SETPS

### METHOD LIST ATTRIBUTES

- Configure method list attributes of the method PART\_CATALOG\_CFP.Cf\_New\_\_
- Map the OBJID column (parameter of the Cf\_New\_\_ method ) with OBJID column of the previous execution sequence
- Custom fields data values are packed into a special attribute string called ATTR\_CF\_
  - Specify ATTR\_ as the parameter value here

The screenshot shows the 'Method List Attribute' interface for the method 'PART\_CATALOG\_CFP.Cf\_New\_\_'. The table lists the attributes with their descriptions, data types, and flags. The first attribute, '510 CF\$\_GOVT\_CODE', is highlighted in green.

| Column Sequence | Column Name    | Description     | Fixed Value | Flags                    | On New                   | On Modify                | LU Reference | ID Values |
|-----------------|----------------|-----------------|-------------|--------------------------|--------------------------|--------------------------|--------------|-----------|
| 510             | CF\$_GOVT_CODE | Government Code | IU          | <input type="checkbox"/> | <input type="checkbox"/> |                          |              |           |
| 520             | INFO_          | VARCHAR2 OUT    | ARG         | <input type="checkbox"/> | <input type="checkbox"/> |                          |              |           |
| 530             | OBJID_         | VARCHAR2 IN     | OBJID_@10   | ARG                      | <input type="checkbox"/> | <input type="checkbox"/> |              |           |
| 540             | ATTR_CF_       | VARCHAR2 IN OUT | ATTR_       | ARG                      | <input type="checkbox"/> | <input type="checkbox"/> |              |           |
| 550             | ATTR_          | VARCHAR2 IN     | ARG         | <input type="checkbox"/> | <input type="checkbox"/> |                          |              |           |
| 560             | ACTION_        | VARCHAR2 IN     | ARG         | <input type="checkbox"/> | <input type="checkbox"/> |                          |              |           |

# MIGRATION SETPS

## SOURCE MAPPING

- Map the custom fields
- PART\_CATALOG\_CVF.CF\$\_GOVT\_CODE => GOVT\_CODE**
- The concatenated value of the custom fields columns are mapped to the DUMMY column

Migration Job - MIGRATE\_PARTS\_IC - 2 (2)

Job Id: MIGRATE\_PARTS\_IC Description: Migrate Part Catalog from IC with Custom Fields

Source System: SOURCE\_SYSTEM Target System: TARGET\_SYSTEM

Source Name: SOURCE\_SOURCE\_DATA Group ID: SOURCE\_GROUP\_ID

Notes:

Formatting: Resolution: Associate Registry: File Mapping: Method List: Rules

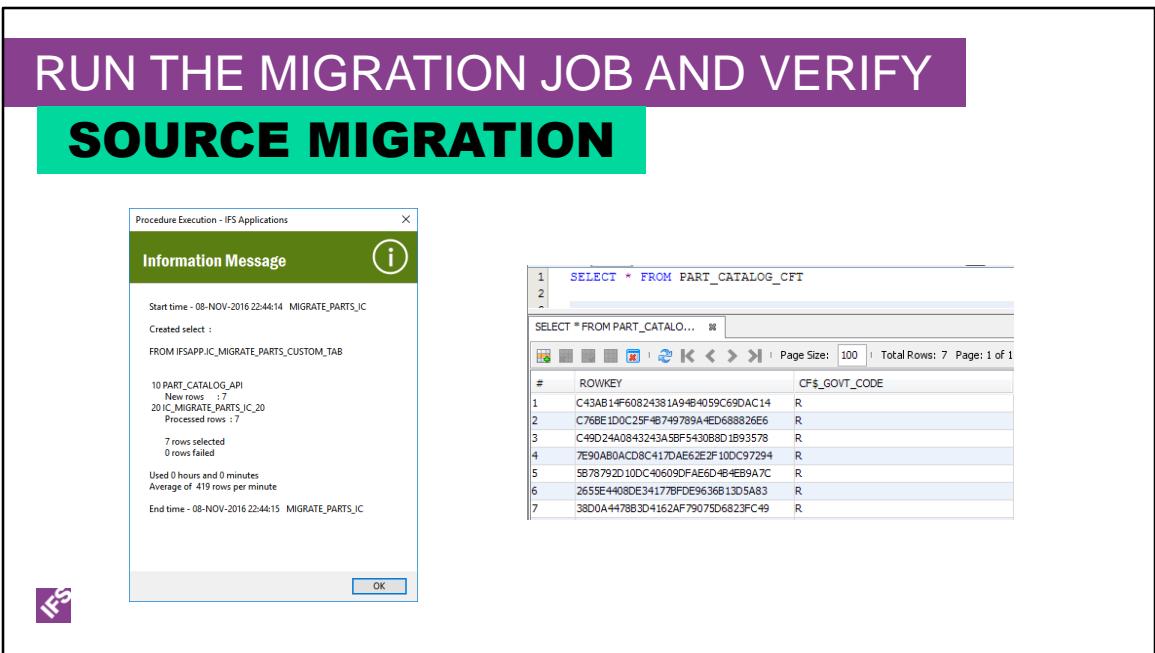
| Row  | From Column Name                         | Description                         | To Column Name | Type     | Data Type   | Length  | Source Column | Default Value | ID Values | Conversion List ID | Notes                     |
|------|--|-------------------------------------|----------------|----------|-------------|---------|---------------|---------------|-----------|--------------------|---------------------------|
| 390  | PART_CATALOG.DIM_FK_PU_VOLUME_NET        | Volume ID#                          | 1              | VARCHAR2 | 38          |         |               |               |           |                    |                           |
| 400  | PART_CATALOG.PRICING_FACTOR              | Freight Factor                      | 2              | NAMES    | 28          | 'T'     |               |               |           |                    | Default value "T" for     |
| 410  | PART_CATALOG.DIM_ALIQUOT_AS_NFT_COMBINED | Allow as Net Combined               | 3              | VARCHAR2 | 4000        |         |               |               |           |                    | Default value "False" for |
| 420  | PART_CATALOG.DIM_ALIQUOT_AS_NFT          | Allow as Net                        | 4              | VARCHAR2 | 4000        |         |               |               |           |                    | Default value "True" for  |
| 430  | PART_CATALOG.DIM_ALIQUOT_AS_NFT_D        | Allow as Net Dimension              | 5              | VARCHAR2 | 4000        |         |               |               |           |                    | Default value "True" for  |
| 440  | PART_CATALOG.RECEIVE_INVOICE_DETAIL_LN   | Serial Tracking & Record art Invoic | 6              | VARCHAR2 | 38          | 'FALSE' |               |               |           |                    | Default value "False" for |
| 450  | PART_CATALOG.DIM_ALIQUOT_AS_NFT_D        | Step Create of New Serial Invoic    | 7              | VARCHAR2 | 4000        |         |               |               |           |                    | Default value "False" for |
| 460  | PART_CATALOG.DIM_ALIQUOT_AS_NFT_R        | Step Create of New Serial Invoic    | 8              | VARCHAR2 | 28          | 'TRUE'  |               |               |           |                    | Default value "True" for  |
| 970  | PART_CATALOG.DIM_VCF_GOV_CODE            | Government Code                     | 9              | VARCHAR2 | 5 GOVT_CODE |         |               |               |           |                    |                           |
| 980  | METHODCODE.QUOTE                         | R                                   | 10             | VARCHAR2 | 2000        |         |               |               |           |                    |                           |
| 1000 | METHODCODE.ATTR_DC                       | In OUT                              | 11             | VARCHAR2 | 2000        |         |               |               |           |                    |                           |
| 1010 | METHODCODE.ATTR_DC                       | R                                   | 12             | VARCHAR2 | 2000        |         |               |               |           |                    |                           |
| 1020 | METHODCODE.ATTR_DC                       | R                                   | 13             | VARCHAR2 | 2000        |         |               |               |           |                    |                           |
| 1030 | DUMMY                                    | Dummy                               | 14             | VARCHAR2 | 5 GOVT_CODE |         |               |               |           |                    |                           |
| 9999 | QSD                                      | QSD                                 | 15             | VARCHAR2 | 28 ROWNID   |         |               |               |           |                    |                           |

Tasks: Attachments: Output: Help: System Info



# RUN THE MIGRATION JOB AND VERIFY

## SOURCE MIGRATION



## EXERCISE 7

### MIGRATION WITH CUSTOM FIELDS

- Add a new custom field called “Government Code” (GOVT\_CODE) to frmPartCatalog form. Position it in the header of the form.
- Create the first migration job that will load the contents of ‘CustomizedPartCatalog.txt’ into a temporary table.
- Create a second migration job to transfer the contents of the temporary table into IFS. Use both PART\_CATALOG\_API and PART\_CATALOG\_CFP in the method list.
- Configure method list attributes of the method PART\_CATALOG\_CFP.Cf\_New\_
- Map DUMMY column to the value of the custom field in Source Mapping tab.
- Run the 2 migration jobs and query back your PART\_CATALOG\_TAB and PART\_CATALOG\_CFT records to confirm the data has loaded.





## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



## 4. REPLICATION

# REPLICATION

## LESSON OBJECTIVES

By the end of this lesson, students should know:

- An overview of how IFS Data Migration can be used in replication of data.
- How to configure the migration job for data replication.
- Different replication mechanisms and configuring
- Different migration parameters related to data replication



# OVERVIEW

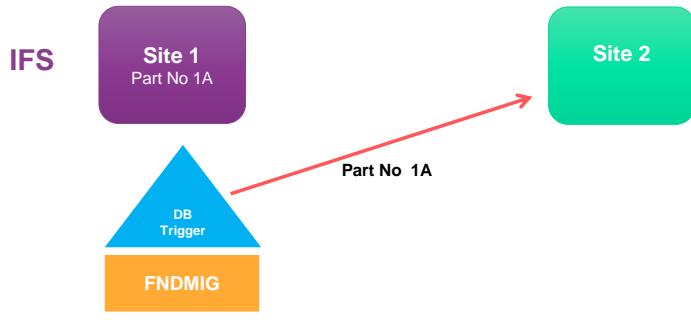
## REPLICATION

- Replicate IFS internal data between different sites and companies.
  - Also known as [Internal Replication](#)
- Normally database Trigger driven, replicate data according to the operation
- Can be used for synchronizations
- NOT a substitute for IFS Replication



# OVERVIEW

## REPLICATION



# JOB CONFIGURATION

## HEADER

### Migration Job

|                    |                           |                  |
|--------------------|---------------------------|------------------|
| Job ID:            | Description:              | Direction:       |
| REPLICATE_INV_PART | Replicate Inventory Parts | Source Migration |
| Procedure Name:    | View Name:                | Group ID:        |
| REPLICATION        | INVENTORY_PART            |                  |
| Notes:             |                           |                  |

IFS

# JOB CONFIGURATION

## MANUAL REPLICATION

- Used to Sync data initially
- Does not create any triggers

Formatting Replication Source Mapping File Mapping Method List Rules

Mode: Manual

Criteria: Repl Criteria: Contract

From value:

To Value List: TOK

Database Information

Source Name: INVENTORY\_PART

Source Owner: IFSAPP

Where: CONTRACT = 'NYC'

Order By:

Last Info:

Trigger

Table Name: INVENTORY\_PART\_TAB

Trigger When:

Enabled Insert Update



# JOB CONFIGURATION

## MANUAL REPLICATION

- Replication Criteria: Company or Contract
- To Value List: Semicolon separated value list

| Criteria       |                      |
|----------------|----------------------|
| Repl Criteria: | Contract             |
| From value:    | <input type="text"/> |
| To Value List: | TOK                  |

- Where clause:

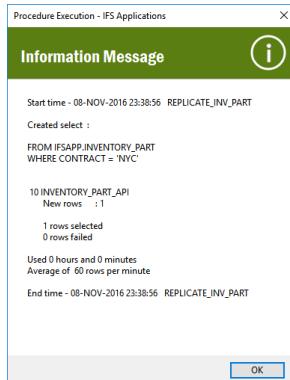
Where:

- Run the Job and verify



# JOB EXECUTE

## MANUAL REPLICATION



IFS

## EXERCISE 8

### MANUAL REPLICATION

- Create a manual replication job named 'REP\_INVENTORY\_PART' for the view 'INVENTORY\_PART'
- Set the replication criteria to 'Contract'. Data should be migrated from site 'CMB' to 'TOK'.
- Add a new inventory part to site 'CMB'.
- Run the migration job and verify whether it is migrated to site 'TOK'.



# JOB CONFIGURATION

## AUTOMATIC REPLICATION

- Trigger based:
  - Fired when inserting/updating source data
  - No need to run the migration job manually
- Need to specify the source (From value) & the destination (To Value List)

| Criteria   |
|--|
| Repl Criteria: <input type="text" value="Contract"/> |
| From value: <input type="text" value="NYC"/>         |
| To Value List: <input type="text" value="TOK"/>      |

- You may enter a Where Clause to specify a condition for the replication



# JOB CONFIGURATION

## AUTOMATIC REPLICATION

- It's better to keep the trigger disabled until the job is configured
- Column "From Value" generates a default "Trigger-When" clause
- We can also choose which events on the table (Insert/Update) should start the replication

The screenshot shows the Oracle Database Job Configuration interface. It is divided into two main sections: 'Criteria' on the left and 'Trigger' on the right.

**Criteria:**

- Repl Criteria: Contract
- From value: NYC
- To Value List: TOK

**Database Information:**

- Source Name: INVENTORY\_PART
- Source Owner: IFSAPP

**Trigger:**

- Table Name: INVENTORY\_PART\_TAB
- Trigger When: new.contract = 'NYC'
- Enabled:
- Insert:
- Update:

IFS

# AUTOMATIC REPLICATION

## CODE BEHIND

### Replication Objects

| Job ID             | Object name            | Object name | Status  | Table Name         | Error Text | Created               | Last DDL time         |
|--------------------|------------------------|-------------|---------|--------------------|------------|-----------------------|-----------------------|
| REPLICATE_INV_PART | RPL_REPLICATE_INV_PART | TRIGGER     | ENABLED | INVENTORY_PART_TAB |            | 11/8/2016 11:44:36 PM | 11/8/2016 11:46:39 PM |

IFS Applications - Editor [HiddenSql]

```
PRAGMA RPL_REPLICATE_INV_PART
AFTER INSERT OR UPDATE ON INVENTORY_PART_TAB FOR EACH ROW
WHEN (new.contract = 'NYC')
DECLARE
    old_id_number;
    event_short VARCHAR2(2000);
    info_ VARCHAR2(2000);
BEGIN
    SELECT Rep_Inv_Seq.NEXTVAL INTO log_id_ FROM DUAL;
    IF INSERTING THEN
        event_short := 'T';
        BEGIN
            event_short := U;
        END;
        IF ((old.decline_date IS NULL AND new.decline_date IS NULL)
            OR ((old.decline_date IS NOT NULL AND new.decline_date IS NOT NULL)
                AND (old.decline_date < new.decline_date)))
            AND ((old.expired_date IS NULL AND new.expired_date IS NULL)
                OR ((old.expired_date IS NOT NULL AND new.expired_date IS NOT NULL)
                    AND (old.expired_date < new.expired_date)))
            AND ((old.decline_issue_counter IS NULL AND new.decline_issue_counter IS NULL)
                OR ((old.decline_issue_counter IS NOT NULL AND new.decline_issue_counter IS NOT NULL)
                    AND (old.decline_issue_counter < new.decline_issue_counter)))
        )
```

IFS

## AUTOMATIC REPLICATION

### CODE BEHIND

- When the trigger is triggered :
- Inserts a new record to Intface\_Replication\_Log\_Tab
- The rowid of the new record is copied to "info\_" variable
- Invokes the "Start\_job" method of the "Intface\_Header\_API" with
  - Info\_ (rowid of the new record)
  - 'NOW' (execution plan)
  - 'REP\_INVENTORY\_PART' (name of the data migration job) as parameters

```
INSERT INTO Intface_Replication_Log_tab (Log_Seq,Intface_Name,Trigger_Type,Repl_Rowid,Log_Date,Rowversion)
VALUES (log_id_,'REP_INVENTORY_PART',event_short_,:new.rowid,SYSDATE,SYSDATE);
info_ := 'ROWID='||:new.rowid;
Intface_Header_API.Start_job(info_,'NOW','REP_INVENTORY_PART');
```



## EXERCISE 9

### AUTOMATIC REPLICATION

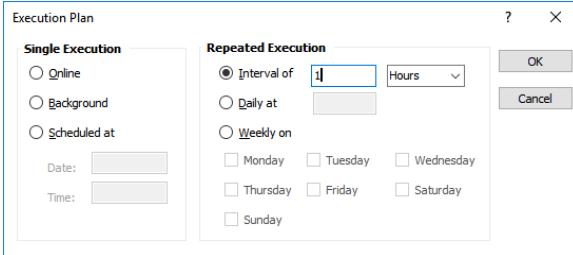
- Create an automatic replication job named 'REP\_INVENTORY\_PART\_AUTO' for the view 'INVENTORY\_PART'
- Set the replication criteria to 'Contract'. Data should be migrated from site 'CMB' to 'DEL'.
- Both inserts and updates of Inventory Parts should start this replication job.
- Add a new inventory part to site 'CMB'.
- Verify whether it is automatically migrated to site 'DEL'.



# JOB CONFIGURATION

## BATCH REPLICATION

- Same as Automatic Replication except for the execution plan



IFS

## RULES

| Rule Name     | Description  |
|---------------|--|
| BUILDTIGGERIF | Build IF-statement for updateable columns  |
| ALTREPLUSER   | Impersonating current user with the "Data Migration alternative user for replication jobs" defined in the system parameters window |



### **BUILDTIGGERIF :**

To avoid that Replication is executed on all Updateable columns, you may activate the rule BUILDTIGGERIF to limit when the replication is executed. Combine this rule with the columns under "Method List Attributes" that are checked On Modify



## REPLICATION OUT

- Simple way of sending XML-messages, also known as Replication Out
- Reads data from IFS and sends a message to IFS Connect using procedure  
PLSQLAP\_Server\_API.Post\_Outbound\_BizAPI
- Message types
  - Single level
  - Structured
- The message may be addressed to a specific BIZAPI in Connect



# JOB CONFIGURATION

## JOB HEADER

- Procedure name: REPLIC\_DATA\_OUT
- Mode: Automatic or Batch (Manual mode - testing purpose)
- Message Type: BIZAPI (DisplayTrace – debugging purpose)
- Message Sender and Message Receiver is used to identify your message in IFS Connect

### Replication Structure

|                        |                                |                   |
|------------------------|--------------------------------|-------------------|
| Job ID:                | Description:                   | Mode:             |
| REP_OUT_INVENTORY_PROD | Send Inventory Product Details | Automatic         |
| Procedure Name:        | Group ID:                      | Message Name:     |
| REPLIC_DATA_OUT        |                                |                   |
| Message Sender:        | Message Receiver:              | Date Format:      |
| IFS                    | RUFELK                         |                   |
| Notes:                 |                                | Server Processes: |
|                        |                                | 0                 |



# JOB CONFIGURATION

## MESSAGE STRUCTURE

- First you specify the views that are part of the structure
- Then you specify the relations between them. One has to be the master, i.e. no value added here, the rest must point to their parent view (by Sequence).
- If the replication is of type Batch or Automatic, you may specify which table(s) should trigger the event

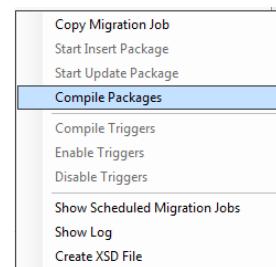
| Seq | View Name             | Parent Sequence | Table Name                | On Insert                           | On Update                           | Record Name | Element Name | Element Type |
|-----|-----------------------|-----------------|---------------------------|-------------------------------------|-------------------------------------|-------------|--------------|--------------|
| 10  | TRN_INVENTORY         |                 |                           | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |              | Array        |
| 20  | TRN_INVENTORY_PRODUCT | 10              | TRN_INVENTORY_PRODUCT_TAB | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |              | Array        |

IFS

## JOB CONFIGURATION

### CREATE AND COMPILE DATABASE OBJECTS

- Use the context menu option to create and compile packages
  - RPL\_<JOBName>\_I
  - RPL\_<JOBName>\_U
- Use the context menu option to create and enable database trigger
- Once the database trigger is enabled, it is not possible to change the migration job
- Create XSD if necessary



## JOB CONFIGURATION

### CREATE AND COMPILE DATABASE OBJECTS

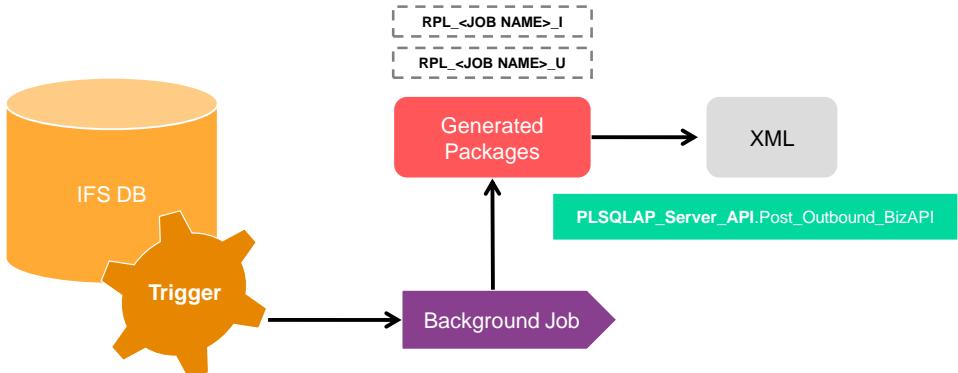
DB Objects

| Object name                    | Object Type  | Status  | Table Name                | Error Text | Date Created          | Last DDL time         |
|--------------------------------|--------------|---------|---------------------------|------------|-----------------------|-----------------------|
| RPL REP OUT INVENTORY PROD I   | PACKAGE BODY | VALID   |                           |            | 11/9/2016 12:29:30 AM | 11/9/2016 12:29:30 AM |
| RPL REP OUT INVENTORY PROD U   | PACKAGE BODY | VALID   |                           |            | 11/9/2016 12:29:30 AM | 11/9/2016 12:29:30 AM |
| RPL REP OUT INVENTORY PROD T20 | TRIGGER      | ENABLED | TRN_INVENTORY_PRODUCT_TAB |            | 11/9/2016 12:29:52 AM | 11/9/2016 12:30:07 AM |



# REPLICATION OUT

## BIG PICTURE



## GENERATED XML

```
<?xml version="1.0"?>
- <TRN_INVENTORY>
  <COMPANY_ID>TCS</COMPANY_ID>
  <BRANCH_ID>BAN-01</BRANCH_ID>
  <INVENTORY_ID>1234</INVENTORY_ID>
  <NAME>Banglore Branch Inventory</NAME>
  <PICKING LEAD TIME>32</PICKING LEAD TIME>
- <TRN_INVENTORY_PRODUCT>
  - <TRN_INVENTORY_PRODUCT>
    <COMPANY_ID>TCS</COMPANY_ID>
    <BRANCH_ID>BAN-01</BRANCH_ID>
    <INVENTORY_ID>1234</INVENTORY_ID>
    <PRODUCT_ID>2122</PRODUCT_ID>
    <QUANTITY>37</QUANTITY>
    <REORDER_LEVEL>15</REORDER_LEVEL>
    <RESERVED_QUANTITY>0</RESERVED_QUANTITY>
  </TRN_INVENTORY_PRODUCT>
  - <TRN_INVENTORY_PRODUCT>
    <COMPANY_ID>TCS</COMPANY_ID>
    <BRANCH_ID>BAN-01</BRANCH_ID>
    <INVENTORY_ID>1234</INVENTORY_ID>
    <PRODUCT_ID>2191</PRODUCT_ID>
    <QUANTITY>3</QUANTITY>
    <REORDER_LEVEL>5</REORDER_LEVEL>
    <RESERVED_QUANTITY>5</RESERVED_QUANTITY>
  </TRN_INVENTORY_PRODUCT>
  - <TRN_INVENTORY_PRODUCT>
    <COMPANY_ID>TCS</COMPANY_ID>
    <BRANCH_ID>BAN-01</BRANCH_ID>
    <INVENTORY_ID>1234</INVENTORY_ID>
    <PRODUCT_ID>2204</PRODUCT_ID>
```



## OUT QUEUE

- Your XML message gets posted in OUT1 queue
- Next the IFS Batch Processor will pickup the message from OUT1 queue and process the message according to the instructions given in the routing rule

Application Messages

| Application Message ID:     |  | Created From: |                          |
|-----------------------------|--|---------------|--------------------------|
| 45                          |  |               |                          |
| Subject:                    | Message from Pref Access Provider          |               |                          |
| Message Type:               | Sender:                                    | Initiated By: | Initiated:               |
| INIF_JBZAPI                 | IFS  | IFSCONNECT    | 11/9/2016 12:33 AM       |
| Message Function:           | Receiver:                                  | Execute As:   | Locale:                  |
| REP_OUT_INVENTORY_PR00      | RUFRLK                                     | Initiator     | en-US                    |
| Status:                     | Queue:                                     | Inbound....   | <input type="checkbox"/> |
| Failure                     | External Message ID:                       |               |                          |
| External Message ID:        | 1249307939PDH4888A0AF5381000213            |               |                          |
| Status Date:                | Options:                                   |               |                          |
| 11/9/2016 12:33 AM          |  |               |                          |
| Error Text:                 | Message Input Data:                        |               |                          |
| No valid routing rule found | <a href="#">REP_OUT_INVENTORY_PR00.xml</a> |               |                          |

IFS

# ROUTING ADDRESS

## Routing Address - Inventory Product Details to File

Destination Type: **File**

Description: **Inventory Product Details to File**

Format:

Envelope:

Encoding:

Transformers:

Response Transformers:

Compress

[Advanced](#)

**File Address**

File Sender: **FILE\_SENDER1**

Output File: **%FILE\_OUT%\%MESSAGE\_FUNCTION%\_%GUID%.xml**



# ROUTING RULE

## Routing Rule - Route Inventory Products

Route From: APPLICATION\_MESSAGE  Enabled  Customized

Description: Route Inventory Products

Queue: QUT1

### Destination Address

| Chain Link No | Description                       | Connector | Address                         |
|---------------|-----------------------------------|-----------|---------------------------------|
| 1             | Inventory Product Details to File | File      | %FILE_OUT%/%MESSAGE_FUNCTION... |

### Content based conditions

| Search   | Op     | Match  |
|----------|--------|--------|
| SENDER   | equals | IFS    |
| RECEIVER | equals | RUFELK |



# FINISHED MESSAGE

## Application Messages

|                         |  |
|-------------------------|--|
| Application Message ID: | 41   |
| Subject:                | Message from Psql Access Provider                      |
| Message Type:           | Sender:<br>INTE_BIZAPI                                 |
| Message Function:       | Receiver:<br>REP_OUT_INVENTORY_PROD                    |
| Status:                 | External Message ID:<br><b>Finished</b>                |
| Status Date:            | 23473D7589FD440B88A0AF53810DEE13<br>11/9/2016 12:42 AM |
| Options:                |  |
| Error Text:             |  |

|                     |                            |
|---------------------|----------------------------|
| Created From:       |                            |
| Initiated By:       | IFSCONNECT                 |
| Initiated:          | 11/9/2016 12:33 AM         |
| Execute As:         | Initiator                  |
| Locale:             | en-US                      |
| Queue:              | OUT1                       |
| Inbound....         | <input type="checkbox"/>   |
| Routing Rule used:  | Route Inventory Products   |
| Message Input Data: | REP_OUT_INVENTORY_PROD.xml |

## Destination Addresses Response Addresses

| Address                                    | Connector | Status      | Text | Message Data         | Envelope | Transformer | Transformer | Response Tr |
|--|-----------|-------------|------|----------------------|----------|-------------|-------------|-------------|
| D %FILE_OUT%/%MESSAGE_FUNCTION%_%GUID%.xml | File      | Transferred |      | MESSAGE_RESPONSE.txt |          |             |             |             |



## EXERCISE 10

### REPLICATION OUT

- Create a replication out job named 'REPOUT\_INVENTORY\_PROD'. Set Mode to 'Automatic' and Message Type to 'BizApi'.
- Configure message structure to include views TRN\_INVENTORY and TRN\_INVENTORY\_PRODUCT. Set TRN\_INVENTORY\_PRODUCT\_TAB to trigger the event on inserts and updates.
- Create and compile the database objects (packages and triggers). Enable the triggers.
- Insert a record into TRN\_INVENTORY\_PRODUCT\_TAB.
- Verify the message with the generated XML file in OUT1 queue.





## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



## 6. EXCEL MIGRATION

# EXCEL MIGRATION

## LESSON OBJECTIVES

By the end of this lesson, students should know:

- About IFS Data Migration Excel Add-In
- How Excel Migration Jobs work
- How to configure/execute Excel migration jobs
- Grant Excel migration jobs for end users



## BACKGROUND

# INTERFACING WITH EXCEL

- Excel sheets heavily used in business operations
- Very useful to have something to move data between IFS Applications and Excel sheets
- IFS Spreadsheet Data Manager (SDMAN) from IFS Supply Chain Management can manipulate IFS data within an Excel sheet but lacks configurability and is outdated
- Goal with IFS Data Migration Excel Add-in is to provide a generic and configurable framework solution to serve this purpose
  - Use the Add-in directly in Microsoft Excel to read IFS Applications data
  - Modify the data using all available Excel features
  - Send the data back to IFS Applications to be updated/inserted using standard Logical Unit methods



# SELLING IFS DATA MIGRATION EXCEL ADD-IN

## EXCEL MIGRATION

- Separate Sales part 9001012ATU – IFS Data Migration Excel Add-in ATU



| Module          | Name                                    | Track    | Market | Vertical | Configuration Type | Sales Part Type            | Status | Sales Part Created | Owner Company Id | Owner Empl No | Owner Name     | Description  |
|-----------------|---|----------|--------|----------|--------------------|----------------------------|--------|--------------------|------------------|---------------|----------------|--|
| 1 □ 9001000     | IFS Foundation - LAYE <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Foundation is the technology platform for IFS Applications. Based on open standards <sup>2,2</sup> . |
| 2 □ 9001001     | IFS Cloud Uplink                        | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL, Restricted | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Cloud Uplink establishes a bi-directional, secure, communications link between any <sup>2,2</sup> .  |
| 3 □ 9001002     | IFS Touch Apps Subsol <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL, Restricted | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Touch Apps is a series of Smartphone apps for iOS (iPhone) and Android platforms <sup>2,2</sup> .    |
| 4 □ 9001003     | Touch Apps Server Conn <sup>2,2</sup>   | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL, Restricted | Final  | ☒                  | 57               | 840           | Roger Laudon   | A device license will be required for each device running Touch Apps and connecting <sup>2,2</sup> .     |
| 5 □ 9001004     | IFS Application Extens <sup>2,2</sup>   | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Application Extensibility includes solutions for tailoring IFS Applications using <sup>2,2</sup> .   |
| 6 □ 9001005     | Crystal Reports Integrat <sup>2,2</sup> | IFSAPP99 | Core   | None     | Sales              | Third Party License GPL    | Final  | ☒                  | 57               | 840           | Roger Laudon   | This sales part consists of OEM software required for the Crystal Reports Integration <sup>2,2</sup> .   |
| 7 □ 9001009     | IFS Middleware Server                   | IFSAPP99 | Core   | None     | Sales              | Third Party License GPL    | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Middleware Server is the application server for IFS Applications 9 and a mandatory <sup>2,2</sup> .  |
| 8 □ 9001010     | IFS Middleware Server <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Third Party License GPL    | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Middleware Server is the application server for IFS Applications 9 and a mandatory <sup>2,2</sup> .  |
| 9 □ 9001011     | IFS Middleware Server <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Third Party License GPL    | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Middleware Server is the application server for IFS Applications 9 and a mandatory <sup>2,2</sup> .  |
| 10 □ 9001012ATU | IFS Data Migration Excel <sup>2,2</sup> | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | ATU Sales Part for IFS Data Migration Excel Add-in. IFS Data Migration Excel Add-in <sup>2,2</sup> .     |
| 11 □ 9001013    | IFS Lobby Configuration                 | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | A Lobby provides an at-a-glance view of information in IFS Applications based on your <sup>2,2</sup> .   |
| 12 □ 9001014    | SQL Server Reporting <sup>2,2</sup>     | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | With the SSRS Reporting Integration you will be able to use MS SQL Server Reporting <sup>2,2</sup> .     |
| 13 □ 9001015    | IFS Mobile Framework                    | IFSAPP99 | Core   | None     | Sales              | Sales Part Non-GPL         | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Mobile Framework (formerly known as IFS Mobile Client) provides an enhanced mobile <sup>2,2</sup> .  |
| 14 □ 9001016    | IFS Mobile Work Order                   | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | The Mobile Work Order solution supports separate and route work order processes and <sup>2,2</sup> .     |
| 15 □ 9001017    | Mobile Work Order - W <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Sales Part Non-GPL         | Final  | ☒                  | 57               | 840           | Roger Laudon   | This sales part applies to Windows OS (Desktop/Laptop/Tablet) Windows 8.1 Client and <sup>2,2</sup> .    |
| 16 □ 9001018    | Mobile Work Order - A <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Sales Part Non-GPL         | Final  | ☒                  | 57               | 840           | Roger Laudon   | This sales part applies to Android Clients and it's purpose is to keep track of the <sup>2,2</sup> .     |
| 17 □ 9001019    | IFS Mobile Work Order <sup>2,2</sup>    | IFSAPP99 | Core   | None     | Sales              | Sales Part Non-GPL         | Final  | ☒                  | 57               | 840           | Roger Laudon   | This sales part applies to iOS Clients and it's purpose is to keep track of the number <sup>2,2</sup> .  |
| 18 □ 9001020    | IFS Lobby Web Access                    | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Lobby Web Access extends IFS Lobby concept making it possible to view Lobby pages <sup>2,2</sup> .   |
| 19 □ 9001021    | Application Visualizer                  | IFSAPP99 | Core   | None     | Sales              | Supplementary Non-GPL      | Final  | ☒                  | 57               | 840           | Roger Laudon   | This Supplementary Sales Part support the move of the component APPVIZ (Application V <sup>2,2</sup> .   |
| 20 □ 9001022    | IFS Applications In-Mem <sup>2,2</sup>  | IFSAPP99 | Core   | None     | Sales              | Sales Part Non-GPL         | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Applications In-memory includes the IFS tools and pre-defined Acceleration Packag <sup>2,2</sup> .   |
| 21 □ 9001023    | IFS Data Synchronization                | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | IFS Data Synchronization provides the framework runtime, configuration and monitoring <sup>2,2</sup> .   |
| 22 □ 9001044CTU | Offshore Worker LTU                     | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 840           | Roger Laudon   | The Offshore Worker LTU (CTU license) is specifically aimed and defined for the offsh <sup>2,2</sup> .   |
| 23 □ 9001100    | Consolidated Accounts                   | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 892           | Tomas Hellgren | IFS Consolidated Accounts puts you in full control of your enterprise performance by <sup>2,2</sup> .    |
| 24 □ 9001101    | Accounts Receivable                     | IFSAPP99 | Core   | None     | Sales              | Sales Part GPL             | Final  | ☒                  | 57               | 892           | Tomas Hellgren | Handles all types of customer invoices, integrated with IFS Customer Order. Contains <sup>2,2</sup> .    |

# SELLING IFS DATA MIGRATION EXCEL ADD-IN

## EXCEL MIGRATION

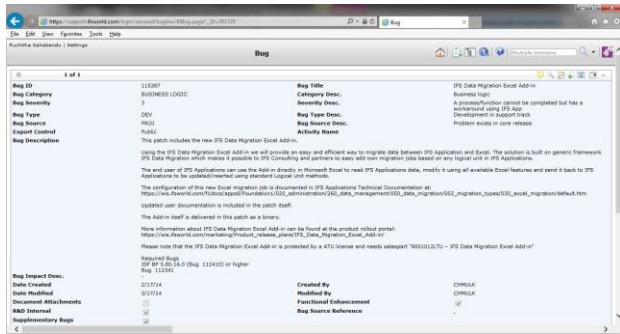
- Separate Sales part 9001012ATU – IFS Data Migration Excel Add-in ATU
- License type ATU (Additional Task User License )
- Connected to permission set FNDMIG\_EXCEL\_ADDIN
- Excel Add-in and used server interface are protected
- The Excel Add-in will be available in FNDMIG component



# INSTALLING IFS DATA MIGRATION EXCEL ADD-IN

## BUSINESS LOGIC AND STORAGE TIERS

- Runs on Foundation 18 SP2 or higher (including IFS Applications 8 SP1 after 25 October 2015)



- Firstly install the following bug fixes
    - JSF BP 5.80.16.0 (Bug 112410) or higher
    - Bug 112341
  - Next install the bug fix 115387 to your existing IFS Home to install IFS Data Migration Add-in
  - Supplementary Bug: [117343](#)

# INSTALLING IFS DATA MIGRATION EXCEL ADD-IN

## CLIENT INSTALLATION

← → 🔍 [https://cmblmdev02.corpelt.ifsworld.com:58080/index\\_admin.htm](https://cmblmdev02.corpelt.ifsworld.com:58080/index_admin.htm)

### Welcome to IFS Applications 10

#### Application Access



**IFS Arena**  
Main user interface for employees, providing browser based access on any device.



**IFS Enterprise Explorer - Admin**  
User interface for employees, connecting from Windows devices, Admin Gateway.



**IFS Enterprise Explorer**  
User interface for employees, accessing from Windows devices.



**IFS Arena B2B**  
Entry point for business-to-business solutions.

#### Other Links



**IFS Middleware Server Admin Console**  
Used to manage a Middleware Server installation.



**IFS System Monitoring**  
IFS System Monitoring console monitors the performance and health of IFS Applications.



**IFS Interface Browser**  
Provides technical interface specifications of service APIs exposed in IFS Applications.



**Add-ons**  
Install add-ons or pre-requisites needed to use some specific parts of IFS applications.



**Documentation**  
Online documentation in English and technical documentation.

#### Add-ons:

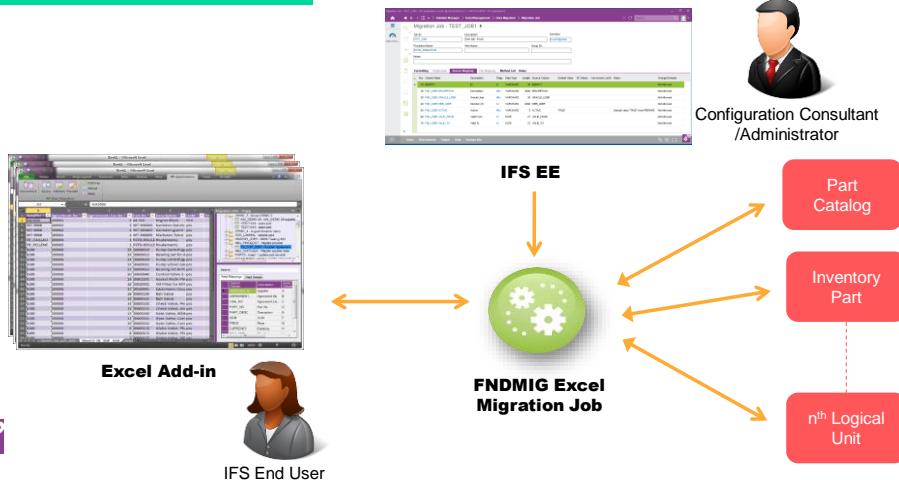
- If running Chrome as default browser, you need to install an extension to get the IFS Enterprise Explorer links to work properly. The extension consists of two parts, an [msi file](#) that needs to be installed on the client machine and a [Chrome extension](#) that needs to be installed from Chrome Web Store.
- To create and update data in IFS Applications using Microsoft Excel you need to install the [IFS Data Migration Excel Add-in](#). As a prerequisite you need to install [Visual Studio 2010 Tools for Office Runtime](#).
- To run and develop .NET based add-ins for IFS you need to install the [IFS .NET Access Provider](#).
- To enable Skype for Business / Lync integration for Streams you need to install the [Lync 2010 SDK Redistributable](#).

[End user access](#)



# IFS DATA MIGRATION EXCEL ADD-IN

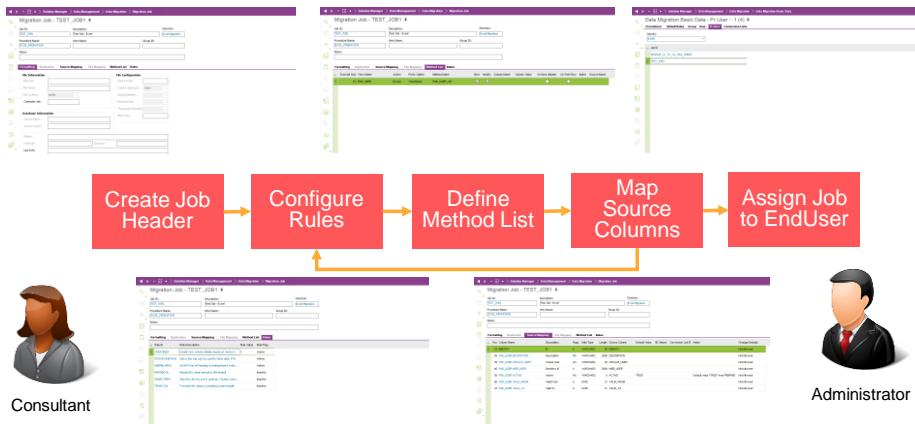
## OVERVIEW



1. A Migration Specialist logs into IFS Application and creates a new Migration Job, and verify's that it is working
2. Admin then grants the created job to end-users.
3. End-users then can use MS Excel to retrieve and modify data in the application.

# IFS DATA MIGRATION EXCEL ADD-IN

## WORKFLOW



1. A Migration Specialist logs into IFS Application and creates a new Migration Job, and verify's that it is working
2. Admin then grants the created job to end-users.
3. End-users then can use MS Excel to retrieve and modify data in the application.

# CONFIGURING THE JOB

## EXCEL MIGRATION

Migration Job - EXCEL\_MIGRATE\_COMPANY\_DATA 21 (21)

|  |                            |                |                            |                      |                   |
|--|----------------------------|----------------|----------------------------|----------------------|-------------------|
| Job ID:  | EXCEL_MIGRATE_COMPANY_DATA | Description:   | Excel Migrate Company Data | Direction:           | Excel Migration   |
| Procedure Name:  | EXCEL_MIGRATION            | View Name:     | TRN_COMPANY                | Group ID:            |                   |
| Notes:   |                            |                |                            |                      |                   |
| <a href="#">Formatting</a> <a href="#">Replication</a> <a href="#">Source Mapping</a> <a href="#">File Mapping</a> <a href="#">Method List</a> <a href="#">Rules</a> |                            |                |                            |                      |                   |
| <b>File Information</b>  |                            |                | <b>File Configuration</b>  |                      |                   |
| File Path:   | File Name:                 | Character Set: | Date Format:               | Column Separator:    | String Delimiter: |
|  |                            |                |                            | <tab>                |                   |
| File Locations:  | File Location:             | Character Set: | Decimal Point:             | Thousands Separator: | Minus Pos:        |
|  | NoFile                     |                |                            |                      |                   |
| <b>Database Information</b>  |                            |                |                            |                      |                   |
| Source Name:   | Source Owner:              | Where:         | Order By:                  | Group By:            | Last Info:        |
| TRN_COMPANY  | IPSAPP                     |                |                            |                      |                   |
| Last Info:   |                            |                |                            |                      |                   |



# CONFIGURING THE JOB

## METHOD LIST

Migration Job - EXCEL\_MIGRATE\_COMPANY\_DATA 21 (21)

Job ID: EXCEL\_MIGRATE\_COMPANY\_DATA Description: Excel Migrate Company Data Direction: Excel Migration

Procedure Name: EXCEL\_MIGRATION View Name: TRN\_COMPANY Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Execute Seq | View Name   | Action | Prefix Option | Method Name     | New                                 | Modify                              | Column Name | Column Value | On New Master            | On First Row             | Notes | Source Name |
|-------------|-------------|--------|---------------|-----------------|-------------------------------------|-------------------------------------|-------------|--------------|--------------------------|--------------------------|-------|-------------|
| 10          | TRN_COMPANY | InLoop | ViewName      | TRN_COMPANY_API | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |             |              | <input type="checkbox"/> | <input type="checkbox"/> |       |             |

IFS

1. Specify the necessary Method List
2. Also specify the main View Name
3. Save the header to get automatic suggestions for the Source Mappings tab

# CONFIGURING THE JOB

## SOURCE MAPPING

Migration Job - EXCEL\_MIGRATE\_COMPANY\_DATA ✓ 21 (21)

Job ID: EXCEL\_MIGRATE\_COMPANY\_DATA Description: Excel Migrate Company Data Direction:

Procedure Name: EXCEL\_MIGRATION View Name: TRN\_COMPANY Group ID:

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Pos | Column Name               | Description   | Flags | Data Type | Length | Source Column    | Default Value | ID Values | Conversion List ID | Notes | Change Defaults |
|-----|---------------------------|---------------|-------|-----------|--------|------------------|---------------|-----------|--------------------|-------|-----------------|
| 10  | COMPANY_ID                | Company ID    | K     | VARCHAR2  | 20     | COMPANY_ID       |               |           |                    |       | Not Allowed     |
| 20  | TRN_COMPANY_NAME          | Name          | MU    | VARCHAR2  | 1000   | NAME             |               |           |                    |       | Not Allowed     |
| 30  | TRN_COMPANY_COMMENTS      | Comments      | IU    | VARCHAR2  | 2000   | COMMENTS         |               |           |                    |       | Not Allowed     |
| 40  | TRN_COMPANY_CREATION_DATE | Creation Date | MU    | DATE      |        | 21 CREATION_DATE |               |           |                    |       | Not Allowed     |

Only columns which have a “Source Column” will be visible to the Excel Add-In

# GRANT EXCEL MIGRATION JOB FOR ENDUSERS

## EXCEL MIGRATION

**Data Migration Basic Data - User**

Procedures Default Rules Group User Pr User Conversion Lists

+ Identity Privilege  
IFSAPP UpdateAllowed  
+ RUFEKL UpdateAllowed

**Data Migration Basic Data - Pr User ▾ 2 (2)**

Procedures Default Rules Group User Pr User Conversion Lists

Identity:  
RUFEKL

+ Job ID  
+ EXCEL\_MIGRATE\_COMPANY\_DATA

Only columns which have a “Source Column” will be visible to the Excel Add-In

# GRANT EXCEL MIGRATION JOB FOR ENDUSERS

## FNDMIG\_EXCEL\_ADDIN

The screenshot shows the IFS Cloud user profile for 'User - Ruwan Fernando'. The profile details include:

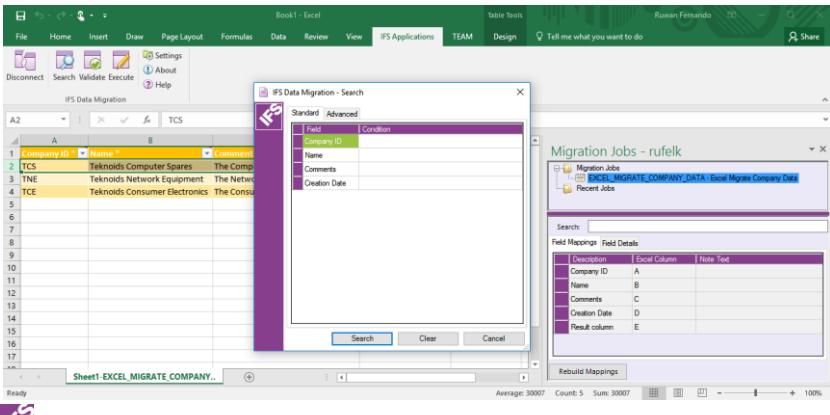
- Identity: RUFEKL
- User Name: Ruwan Fernando
- Directory ID: RUFEKL
- Language: en - English
- E-mail: ruwan.fernando@fsworld.com
- Database ID: RUFEKL

The 'License Usage' is listed as 'Full'. A small user icon is shown with the status 'Active User'. Below the profile, there are two tabs: 'User Settings' (selected) and 'Summary'. Under 'User Settings', the 'Security Settings' section is expanded, showing the 'Permission Sets' tree. The 'APPSRV\_ATTACHMENTS' set is selected, and under it, the 'FNDMIG\_EXCEL\_ADDIN' permission is checked. Other permissions listed include BA\_BARS\_ADMIN, BA\_REPORT\_ADMIN, BA\_REPORT\_DESIGNER, BA\_USER, FND\_ADMIN, and FND\_ANONYMOUS. To the right of the tree, there is a 'Security Settings' summary box with a lock icon, stating: "Users can be assigned to a group of Permission Sets. Permissions Sets grants access to the system based on the assigned roles." A link "View Security Summary for user Ruwan Fernando" is provided.

Only columns which have a “Source Column” will be visible to the Excel Add-In

## PROCESS DATA USING EXCEL ADD-IN

### EXCEL MIGRATION



Only columns which have a “Source Column” will be visible to the Excel Add-In

# INSERT DATA USING EXCEL ADD-IN

## EXCEL MIGRATION

The screenshot shows a Microsoft Excel window titled "Book1 - Excel". The ribbon tabs include File, Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, IFS Applications, TEAM, Design, and Share. A search bar says "Tell me what you want to do". The main area displays a table with data from "Sheet1-EXCEL\_MIGRATE\_COMPANY...". The table has columns: Company ID, Name, Comments, Creation Date, and Result column. The first row is a header. The second row contains data: WGG, Wagga Airways, The Wagga Airways Flight Management, 2016-11-16-00:00:00.00, and a yellow cell. The third row contains TCS, Teknoids Computer Spares, The Computer Warehouse, 1982-02-25-00:00:00.00, and a yellow cell. The fourth row contains TNE, Teknoids Network Equipment, The Network Warehouse, 1986-05-01-00:00:00.00, and a yellow cell. The fifth row contains TCE, Teknoids Consumer Electronics, The Consumer Electronic Warehouse, 1991-08-25-00:00:00.00, and a yellow cell. Rows 6 through 17 are empty. On the right side, a "Migration Jobs - rufelk" pane is open, showing a tree view with "Migration Jobs" expanded, "EXCEL\_MIGRATE\_COMPANY\_DATA: Excel Migrates Company Data" selected, and "Recent Jobs" listed. Below the tree view is a "Search" input field and a "Field Mappings" table. The table has four rows: Description (Company ID), Excel Column (A), Note Text (Name), and Description (Comments), Excel Column (B), Note Text (Comments), and so on. The "Field Mappings" table has a header row with "Description", "Excel Column", and "Note Text". The data rows are: Company ID, A; Name, B; Comments, C; Creation Date, D; and Result column, E. At the bottom of the pane, there is a "Rebuild Mappings" button and a "Count: 4" indicator.

Only columns which have a “Source Column” will be visible to the Excel Add-In

## INSERT DATA USING EXCEL ADD-IN

### EXCEL MIGRATION

The screenshot shows the IFS Data Migration interface. At the top, there are several icons: Disconnect, Search, Validate (which is highlighted with a red box), Execute, Settings, About, and Help. Below this is a toolbar with standard Excel-like buttons: a dropdown arrow, a search icon, a validate icon (highlighted), a pencil icon, and a green checkmark icon. The main area is titled "IFS Applications" with the sub-tittle "What do you want to process?". It contains three buttons: "Selected rows only" (highlighted with a red box), "All rows", and "Cancel". Below this is a table with data. The first row has headers: "Company ID \*", "Name \*", "Comments", "Creation Date \*", and "Result column". The second row, which is highlighted with a red box, contains the values: "WGG", "Wagga Airways", "The Wagga Airways Flight Management", "2016-11-16-00.00.00", and "Validate OK". The third row contains "TCS", "Teknoids Computer Spares", "The Computer Warehouse", "1982-02-25-00.00.00", and "OK". The fourth row contains "TNE", "Teknoids Network Equipment", "The Network Warehouse", "1986-05-01-00.00.00", and "OK". The fifth row contains "TCE", "Teknoids Consumer Electronics", "The Consumer Electronic Warehouse", "1991-08-25-00.00.00", and "OK". The bottom left corner of the interface has the IFS logo.

| Company ID * | Name *                        | Comments                            | Creation Date *     | Result column |
|--------------|-------------------------------|-------------------------------------|---------------------|---------------|
| 2 WGG        | Wagga Airways                 | The Wagga Airways Flight Management | 2016-11-16-00.00.00 | Validate OK   |
| 3 TCS        | Teknoids Computer Spares      | The Computer Warehouse              | 1982-02-25-00.00.00 | OK            |
| 4 TNE        | Teknoids Network Equipment    | The Network Warehouse               | 1986-05-01-00.00.00 | OK            |
| 5 TCE        | Teknoids Consumer Electronics | The Consumer Electronic Warehouse   | 1991-08-25-00.00.00 | OK            |

Only columns which have a “Source Column” will be visible to the Excel Add-In

## INSERT DATA USING EXCEL ADD-IN

### EXCEL MIGRATION



| A              | B                             | C                                   | D                   | E             |
|----------------|-------------------------------|-------------------------------------|---------------------|---------------|
| 1 Company ID * | Name *                        | Comments                            | Creation Date *     | Result column |
| 2 WGG          | Wagga Airways                 | The Wagga Airways Flight Management | 2016-11-16-00.00.00 | Inserted      |
| 3 TCS          | Teknoids Computer Spares      | The Computer Warehouse              | 1982-02-25-00.00.00 |               |
| 4 TNE          | Teknoids Network Equipment    | The Network Warehouse               | 1986-05-01-00.00.00 |               |
| 5 TCE          | Teknoids Consumer Electronics | The Consumer Electronic Warehouse   | 1991-08-25-00.00.00 |               |



Only columns which have a “Source Column” will be visible to the Excel Add-In

## INSERT DATA USING EXCEL ADD-IN

### EXCEL MIGRATION



| A | B                                 | C                                   | D                   | E             |
|---|-----------------------------------|-------------------------------------|---------------------|---------------|
| 1 | Company ID * Name *               | Comments                            | Creation Date *     | Result column |
| 2 | WGG Wagga Airways                 | The Wagga Airways Flight Management | 2016-11-16 00:00:00 | Inserted      |
| 3 | TCS New Teknoids Computer Spares  | The New Computer Warehouse          | 1982-02-25 00:00:00 | Updated       |
| 4 | TNE Teknoids Network Equipment    | The Network Warehouse               | 1986-05-01 00:00:00 |               |
| 5 | TCE Teknoids Consumer Electronics | The Consumer Electronic Warehouse   | 1991-08-25 00:00:00 |               |

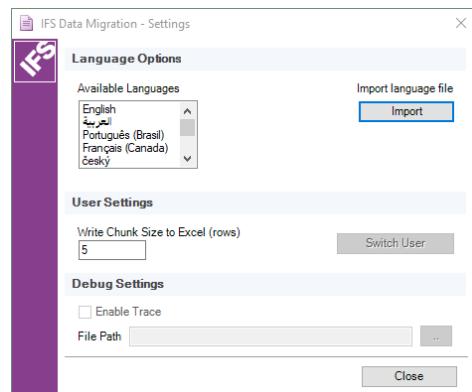
IFS

Only columns which have a “Source Column” will be visible to the Excel Add-In

# SETTINGS

## EXCEL MIGRATION

- Translations
- Switch User
- Debug Settings



## EXERCISE 11

### EXCEL MIGRATION

- Create an Excel Migration job and name it as 'MIGRATE\_COMPANY\_DATA'. Set the view to TRN\_COMPANY.
- Update the list of required columns in 'Source Mapping' tab and set their positions.
- Add TRN\_COMPANY view to Method List. Allow both inserts and updates.
- Grant privileges of the new migration task to ifsapp user.
- Log into Excel Migration Add-on using ifsapp user.
- Insert new records and update existing ones using Excel Add-on.
- Query TRN\_COMPANY and check whether the changes are reflected there.





## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



## 7. EXPORT & IMPORT MIGRATION JOBS

## EXPORT AND IMPORT MIGRATION JOBS

### LESSON OBJECTIVES

By the end of this lesson, students should know:

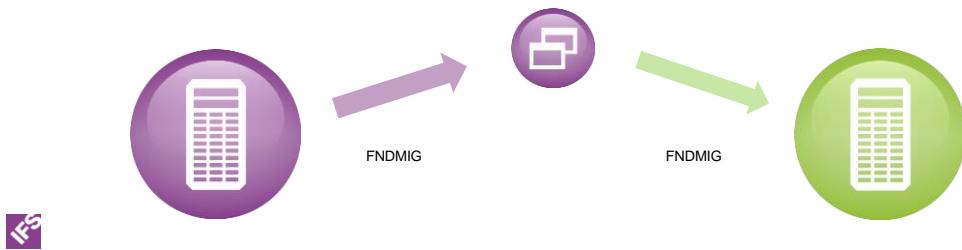
- How to import and export migration jobs
- Different export and import migration job parameters



# EXPORT & IMPORT MIGRATION JOBS

## OVERVIEW

- Provide mechanism to import or export migration metadata (Data Migration Job definitions) from one IFS Environment to another.
- Possible to do bulk import and export
- If a Migration Group is associated that group is attached with metadata



# EXPORT MIGRATION JOB

The screenshot shows the IFS Applications interface with the following details:

- Migration Jobs Screen:** A list of migration jobs. The job `PART_CATALOG_INSERT_UPDATE` is selected, highlighted in green.
- Job Context Menu:** A context menu is open for the selected job. The "Export job..." option is highlighted with a red box and a green arrow pointing to it.
- Information Message Dialog:** A modal dialog titled "Information Message" provides details about the export process.
  - Start time: 09-NOV-2016 02:29:22 FNDMIG\_EXPORT - Execution No 0
  - Exported migration jobs:  
PART\_CATALOG\_INSERT\_UPDATE
  - Exported conversion lists:  
UNIT\_MEASURES
  - File Path: OnClient
  - File Name: UNIT\_MEASURES
  - Used 0 hours and 0 minutes
  - End time: 09-NOV-2016 02:29:22 FNDMIG\_EXPORT

# EXPORT MIGRATION JOB

## Execute Job - FNDMIG\_EXPORT

| Job ID:   | Description:                           | Group ID:   |
|---|--|---|
| FNDMIG_EXPORT                                   | Export of Migration definitions        | MIG_UTIL  |
| File Path:                                      | File Name:                             | File Location:  |
|   |  | Status:   |
| Last Date:                                      | Executed By:                           | Last Info:  |
| 11/9/2016 2:29:22 AM                            | IFPSAPP                                | Start time - 09-NOV-2016 02:29:22 FNDMIG_EXPORT - Execution No 0 Exported migration jobs : PART_CATALOG_INSERT_UPDATE |
| <a href="#">Defaults</a> <a href="#">Detail</a> |  |   |
| Column Name                                     | Description                            | Default Value   |
| INTFACE_NAME                                    | Migration name                         | = 'PART_CATALOG_INSERT_UPDATE'  |
| EXP_LIST_FLAG                                   | Export existing conversion lists (Y/N) | Y   |

| Parameter Name | Description  |
|----------------|--|
| INTFACE_NAME   | Name of the Job(s) to be exported                                |
| EXP_LIST_FLAG  | Indicate the conversion list is imported or not (default is 'Y') |



# EXPORT MIGRATION JOB

**Execute Job - FNNDIG\_EXPORT**

Job ID: FNNDIG\_EXPORT Description: Export of Migration definitions Group ID: MTG\_UTIL

File Path: File Name: File Location: Status: Line Ok

Last Date: 11/9/2016 2:29:22 AM Executed By: DBSAPP Last Info: Start time - 09-NOV-2016 02:29:22 FNNDIG\_EXPORT - Execution No 0 Exported migration jobs : PART\_CATALOG\_INSERT\_UPDATEExported conversio

**Details**

| Line | String  | Attribute    | Value | String | Error Message | Execution No | Status  |
|------|---|--------------|-------|--------|---------------|--------------|---------|
| 1    | --INTFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#ALLOW_AS_N...   | Load File... |       |        |               | 0            | Line Ok |
| 2    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#ALLOW_AS_N...                               |              |       |        |               | 0            | Line Ok |
| 3    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CATCH_UNIT_...                              |              |       |        |               | 0            | Line Ok |
| 4    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#COMPONENT_...                               |              |       |        |               | 0            | Line Ok |
| 5    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CONDITION_C...                              |              |       |        |               | 0            | Line Ok |
| 6    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CONDITION_C...                              |              |       |        |               | 0            | Line Ok |
| 7    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CONFIGURAB...                               |              |       |        |               | 0            | Line Ok |
| 8    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CONFIGURAB...                               |              |       |        |               | 0            | Line Ok |
| 9    | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#DESCRIPTION#...                             |              |       |        |               | 0            | Line Ok |
| 10   | --LOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#DESCRIPTION#...                             |              |       |        |               | 0            | Line Ok |
| 11   | --INTFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#CUST_WARR...    |              |       |        |               | 0            | Line Ok |
| 12   | --INTFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#DESCRIPTION#... |              |       |        |               | 0            | Line Ok |

**Actions**

- 1 Export to File...
- 2 Search in Context...
- 3 Edit
- 4 Output
- 5 Quick Chart...
- 6 Subscriptions
- 7 Conditional Fields...
- 8 Custom Objects
- 9 Properties



# EXPORT MIGRATION JOB

## EXPORT FILE FORMAT

- Data Migration Internal format
  - Can be imported back by using Data Migration client

```
--INTFDINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#UOM_FOR_VOLUME_NET#CHR30#DESCRIPTION#CHR31#Volume_UoM#CHR30#DATA_TYPE#CHR  
--INTFDINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#UOM_FOR_WEIGHT_NET#CHR30#DESCRIPTION#CHR31#Weight_UoM#CHR30#DATA_TYPE#CHR  
--INTFDINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#VOLUME_NET#CHR30#DESCRIPTION#CHR31#Net_Volume#CHR30#DATA_TYPE#CHR31#NUMBE  
--INTFDINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#COLUMN_NAME#CHR31#WEIGHT_NET#CHR30#DESCRIPTION#CHR31#Net_Weight#CHR30#DATA_TYPE#CHR31#NUMBE  
--INTFRINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#RULE_ID#CHR31#IGNOREEXERROR#CHR30#RULE_VALUE#CHR31##CHR30#RULE_FLAG#CHR31#2#CHR30#  
--INTFRINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#RULE_ID#CHR31#SAVE30DAY#CHR30#RULE_VALUE#CHR31#10#CHR30#RULE_FLAG#CHR31#1#CHR30#  
--INTFRINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#RULE_ID#CHR31#SKIPLINES#CHR30#RULE_VALUE#CHR31#1#CHR30#RULE_FLAG#CHR31#1#CHR30#  
--INTFHINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#SYNCDETAILS#CHR30#RULE_VALUE#CHR31##CHR30#RULE_FLAG#CHR31#2#CHR30#  
--INTFHINTERFACE_NAME#CHR31#PART_CATALOG_INSERT_UPDATE#CHR30#DESCRIPTION#CHR31#Part_Catalog_Insert or Update#CHR30#INTERFACE_PATH#CHR31##CHR30#INTERFACE_FIL  
--R_SET#CHR31##CHR30#  
--INTFCCONV_LIST_ID#CHR31#UNIT_MEASURES#CHR30#OLD_VALUE#CHR31#LTR#CHR30#NEW_VALUE#CHR31#1#CHR30#  
--INTFCCONV_LIST_ID#CHR31#UNIT_MEASURES#CHR30#OLD_VALUE#CHR31#METER#CHR30#NEW_VALUE#CHR31##CHR30#  
--INTFCCONV_LIST_ID#CHR31#UNIT_MEASURES#CHR30#OLD_VALUE#CHR31#PCS#CHR30#NEW_VALUE#CHR31#1#PCS#CHR30#  
--INTFLCONV_LIST_ID#CHR31#UNIT_MEASURES#CHR30#DESCRIPTION#CHR31#Unit of Measures#CHR30#  
--INTFENDIMPORT
```



# EXPORT MIGRATION JOB

## EXPORT FILE FORMAT

- Deployable file format using PL/SQL block
- Useful when Data Migration jobs should be installed during installation or delivery of IFS Applications.

```
FNDMIG_EXPORT2PL
```

```
1  --
2  -- Export of Data Migration jobs in deploy file format.
3  --
4  DECLARE
5      job_      VARCHAR2(30) := 'FNDMIG_IMPORT';
6      line_no_   NUMBER      := Intface_Job_Detail_API.Get_Max_Line_No(job_);
7      --
8      del_list_flag_  Intface_Detail.Default_Value#TYPE := 'N';
9      del_job_flag_   Intface_Detail.Default_Value#TYPE := 'N';
10     --
11     file_location_db_ Intface_Header.File_Location_DB#TYPE;
12     intface_path_     Intface_Header.Intface_Path#TYPE;
13     intface_file_    Intface_Header.Intface_File#TYPE;
14     char_set_        Intface_Header.Char_Set#TYPE;
15     do_reset_       BOOLEAN := FALSE;
16     --
```



# EXPORT MIGRATION JOB

## EXPORT FILE FORMAT

- DEPLOYFILE rule controls the file format.
  - DEPLOYFILE = **Inactive** => Internal Data Migration format is created
  - DEPLOYFILE = **Active** => PL/SQL Block format is created.

Migration Job - FNDMIG\_EXPORT

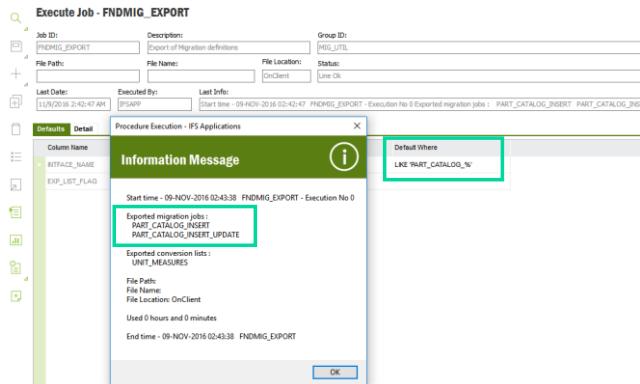
|                 |                           |              |                                 |            |          |
|-----------------|---------------------------|--------------|---------------------------------|------------|----------|
| Job ID:         | FNDMIG_EXPORT             | Description: | Export of Migration definitions | Direction: | File Out |
| Procedure Name: | FNDMIG_EXPORT             | View Name:   |                                 | Group ID:  | MIG_UTIL |
| Notes:          | =====Used to export all c |              |                                 |            |          |

Formatting   Replication   Source Mapping   File Mapping   Method List   Rules

| Rule ID    | Rule Description                              | Rule Value | Rule Flag |
|------------|---|------------|-----------|
| DEPLOYFILE | Make export-file for deployment with SQL tool | Active     | ✓         |

# EXPORT MIGRATION JOB

**DIRECTLY FROM “EXECUTE JOB” WINDOW**



IFS

## EXERCISE 12

### EXPORT MIGRATION JOB

- Go to Migration Jobs window.
- RMB on a specific job and select Export Job.
- Execute the FND\_EXPORT task.
- Export the generated content to a file.
- Activate DEPLOYFILE rule.
- Execute the task again and export the content to another file.



## EXPORT MIGRATION JOB

### EXPORT MULTIPLE MIGRATION JOBS

- Export all jobs belonging to a group

Defaults Detail

| Column Name    | Description                            | Default Value | Default Where  |
|----------------|--|---------------|--|
| ► INTFACE_NAME | Migration name                         |               | IN (SELECT interface_name FROM INTERFACE_HEADER WHERE NVL(GROUP_ID, 'X') = 'MIG_FND_USER') |
| EXP_LIST_FLAG  | Export existing conversion lists (Y/N) | Y             |  |

- Export all jobs

Defaults Detail

| Column Name    | Description                            | Default Value | Default Where |
|----------------|--|---------------|---------------|
| ► INTFACE_NAME | Migration name                         |               | IS NOT NULL   |
| EXP_LIST_FLAG  | Export existing conversion lists (Y/N) | Y             |               |



# EXPORT MIGRATION JOB

## FILE LOCATION = ON SERVER

Migration Job - FNDMIG\_EXPORT

Job ID: FNDMIG\_EXPORT Description: Export of Migration definitions Direction: File Out

Procedure Name: FNDMIG\_EXPORT View Name: Group ID: MIG\_UTIL

Notes: ----- Used to export all

Formatting Replication Source Mapping File Mapping Method List Rules

**File Information**

|                |          |
|----------------|----------|
| File Path:     | FNDMIG   |
| File Name:     | *.ins    |
| File Location: | OnServer |
| Character Set: |          |

**Database Information**

|               |           |
|---------------|-----------|
| Source Name:  |           |
| Source Owner: |           |
| Where:        |           |
| Order By:     | Group by: |

**File Configuration**

|                      |  |
|----------------------|--|
| Date Format:         |  |
| Column Separator:    |  |
| String Delimiter:    |  |
| Decimal Point:       |  |
| Thousands Separator: |  |
| Minus Pos:           |  |

Last Info: Start time - 09-NOV-2016 02:51:00 FNDMIG\_EXPORT - Execution No 0 Exported migration jobs : 0



# EXPORT MIGRATION JOB

## FILE LOCATION = ON SERVER

Execute Job - FNDMIG\_EXPORT

Job ID: FNDMIG\_EXPORT Description: Export of Migration definitions Group ID: MIG\_UTIL

File Path: FNDMIG File Name: \*.ins File Location: OnServer Status: No file

Last Date: 09/11/2016 Procedure Execution - IFS Applications

MSG\_EXPORT - Execution No 0 Created files : PART\_CATALOG\_INSERT.ins

Information Message

Start time - 09-NOV-2016 02:54:08 FNDMIG\_EXPORT - Execution No 0

Created files :  
PART\_CATALOG\_INSERT.ins  
PART\_CATALOG\_INSERT\_UPDATE.ins

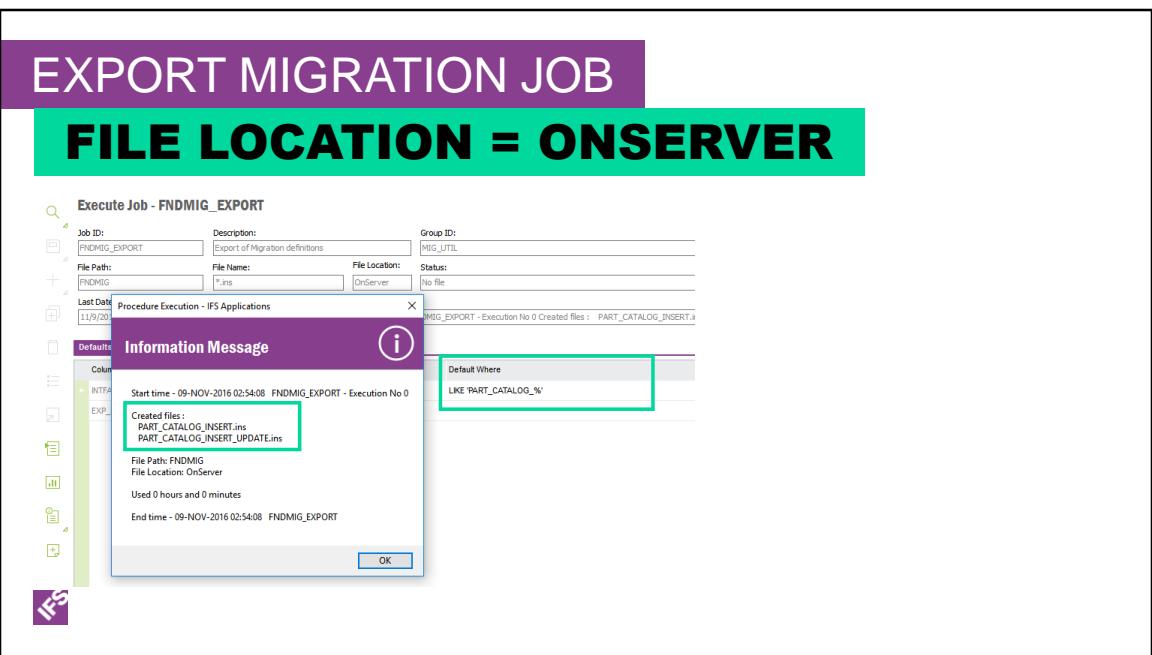
File Path: FNDMIG File Location: OnServer

Used 0 hours and 0 minutes

End time - 09-NOV-2016 02:54:08 FNDMIG\_EXPORT

OK

Default Where  
LIKE 'PART\_CATALOG\_%'



IFS

# EXPORT MIGRATION JOB

## DEPLOYFILE = F

Migration Job - FNDMIG\_EXPORT

Job ID: FNDMIG\_EXPORT Description: Export of Migration definitions

Procedure Name: FNDMIG\_EXPORT View Name: MIG\_UTIL

Notes:

Formatting Replication Source Mapping File Mapping Method List Rules

| Rule ID    | Rule Description                              | Rule Value | Rule Flag |
|------------|---|------------|-----------|
| DEPLOYFILE | Make export-file for deployment with SQL tool | F          | Active    |

Procedure Execution - IFS Applications

Information Message

Start time - 09-NOV-2016 02:57:09 FNDMIG\_EXPORT - Execution No 0

Created files :  
PartCatalogInsert.ins  
PartCatalogInsertUpdate.ins

File Path: FNDMIG  
File Location: OnServer

Used 0 hours and 0 minutes

End time - 09-NOV-2016 02:57:09 FNDMIG\_EXPORT

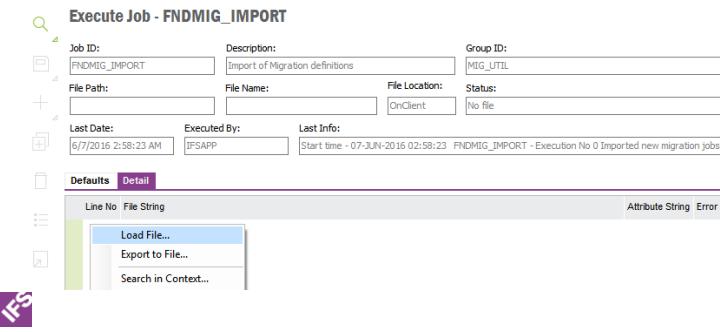
OK

Adding the letter F (for formatting) on the rule DEPLOYFILE, the <JOB\_ID> will be formatted using InitCap + underscores will be removed.

EX: job **CUSTOMER\_INFO\_ADDR** will be written to file **CustomerInfoAddr.ins**

# IMPORT MIGRATION JOB ON CLIENT

- Locate the pre-packed FNFMIG\_IMPORT migration job in the “Execute Job” window
- Load the file and execute the job (FNFMIG\_IMPORT) to complete the import



# IMPORT MIGRATION JOB ON SERVER

- Specify the name of the ins file as shown below
- Importing multiple migrations jobs
  - Copy related “ins” files to “FNDMIG”
  - File Name: \*.ins

**File Information**

|                |                             |
|----------------|-----------------------------|
| File Path:     | FNDMIG                      |
| File Name:     | PartCatalogInsertUpdate.ins |
| File Location: | OnServer                    |
| Character Set: |                             |



# IMPORT MIGRATION JOB

## IMPORT PARAMETERS

| Parameter Name | Description                               |
|----------------|---|
| DEL_JOB_FLAG   | Override existing migration job (Y/N)     |
| DEL_LIST_FLAG  | Overwrite existing conversion lists (Y/N) |

Execute Job - FNDMIG\_IMPORT

Job ID: FNDMIG\_IMPORT Description: Import of Migration definitions Group ID: MIG\_UTIL

File Path: File Name: File Location: Status:

Last Date: Executed By: Last Info: 6/7/2016 2:58:23 AM IFSAAPP Start time - 07-JUN-2016 02:58:23 FNDMIG\_IMPORT - Execution No 0 Imported new migration jobs.

Defaults Detail

| Column Name   | Description                               | Default Value | Default Where |
|---------------|---|---------------|---------------|
| DEL_JOB_FLAG  | Overwrite data (Y/N)                      | N             |               |
| DEL_LIST_FLAG | Overwrite existing conversion lists (Y/N) | N             |               |







## FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



# WRAP-UP

161

The background of the slide features a photograph of a person's hand raised, palm facing forward, against a backdrop of warm, glowing string lights. The overall atmosphere is intimate and focused. A large, solid purple rectangular overlay covers the bottom half of the image, containing the text.

# QUESTIONS?



**PLEASE PROVIDE US FEEDBACK**



# THANK YOU





**IFSworld.com**

© COPYRIGHT© 2020 BY INDUSTRIAL AND FINANCIAL SYSTEMS, IFS AB (PUBL). ALL RIGHTS RESERVED. THIS MATERIAL AND ITS CONTENT IS PRODUCED BY THE IFS ACADEMY FOR AUTHORIZED TRAINING PURPOSES ONLY AND REMAINS THE INTELLECTUAL PROPERTY OF IFS. NEITHER THE MATERIAL OR ITS CONTENT MAY BE COPIED, REPRODUCED, OR DISTRIBUTED WITHOUT IFS' EXPRESS WRITTEN PERMISSION.

IFS DOES NOT WARRANT, EITHER EXPRESSLY OR IMPLIED, THE ACCURACY, TIMELINESS, OR APPROPRIATENESS OF THE INFORMATION CONTAINED IN THIS TRAINING MATERIAL AND DISCLAIMS ANY RESPONSIBILITY FOR CONTENT ERRORS, OMISSIONS, OR INFRINGING MATERIAL. IFS ALSO DISCLAIMS ANY RESPONSIBILITY ASSOCIATED WITH RELYING ON THE INFORMATION PROVIDED IN THIS DOCUMENT AND ANY AND ALL LIABILITY FOR ANY MATERIAL CONTAINED ON OTHER CHANNELS THAT MAY BE LINKED TO THE IFS TRAINING MATERIAL.

