

TABLE OF CONTENTS

1	Oracle Flat File auto generation tool (FLAG).....	2
2	Configuration file properties.....	2
2.1	Database View.	2
2.2	Comments.....	3
2.3	Print Header	3
2.4	Delimiter	3
2.5	Optional delimiter.....	3
2.6	Record layout	4
2.7	Output directory	4
2.8	Output file name.....	5
2.9	Output file name mask.....	5
2.10	Record Format	5
2.11	Skip identical strings	6
3	PLSQL API Definition	6
4	Register FLAG API as a concurrent program in Oracle EBS.....	7
4.1	Register PL/SQL API as concurrent program executable.	7
5	Examples	8
5.1	Configuration file example:.....	8
A.	DDL for View XX_LOC_EMPLOYEES.....	8
5.1.1	PL/SQL code snippet	8
A.	Configuration File - LOC-EMPS.cfg.....	8
5.1.2	10
B.	Configuration File - LOC-EMPSB.cfg.....	10
6	Contact information:.....	12

1 ORACLE FLAT FILE AUTO GENERATION TOOL (FLAG)

Flat file auto generation tool (FLAG) offers both flexible and powerful approach to flat data file generation process.

Traditional approach is to develop a new piece of code every time new integration point needs to be created. FLAG offers an ability to create data file based on the custom data view and simple configuration file. No custom code is needed. This leads to dramatic saving, simplicity, precision and “do it right the first time” approach.

2 CONFIGURATION FILE PROPERTIES.

Configuration file controls the output layout, record format and defines data source for the output data file.

Configuration file has no naming restrictions. User is free to name configuration file based on the enterprise standards.

Individual properties of the configuration file allow to control appearance of the header record, layout of data fields, format of the individual fields, delimiter, optional delimiter as well as master-detail structure and more.

2.1 Database View.

Database view is the main source of data and starting point of the development. Using views simplifies configuration syntax and offers an approach which allows the same data set to be reused in case similar data extract but with the different layout is needed. View is being defined as “QUERY” property and must have “;” (semicolon) at the end. Multi line query is not allowed!

Property name	Example
QUERY	<p>QUERY=Select * from XX_CUSTOM_VIEW;</p> <p>QUERY=Select * from XX_CUSTOM_VIEW where rownum < 21;</p>

2.2 COMMENTS

Any string starting with # is considered a comment and its content is ignored

2.3 PRINT HEADER

Property name	Example
PRINT_HEADER	PRINT_HEADER=YES PRINT_HEADER=NO

PRINT_HEADER determines if header record is included. YES – included, NO – not included.

2.4 DELIMITER

Fields separator.

Property name	Example
DELIMITER	DELIMITER=, DELIMITER=~

2.5 OPTIONAL DELIMITER

Optional delimiter is applied automatically if data field contains delimiter as part of the data field content.

Property name	Example
OPTIONAL_DELIMITER	OPTIONAL_DELIMITER ="

2.6 RECORD LAYOUT

Property name	Example
OUTPUT_LAYOUT	OUTPUT_LAYOUT=EmployeeID AS Employee Identifier FirstName ^LastName Filler HireDate

OUTPUT_LAYOUT specifies record layout. Fields are separated by “|” (pipe) symbol. Field name must match a corresponding column name in the QUERY. “AS” is required if header display name is different from the view column name.

The following syntax is applied:

AS – to specify display column name alias.

Example:

VIEW_COLUMN_NAME **AS** DISPLAYED_NAME – header record contains DISPLAYED_NAME which matches QUERY column name of “VIEW_COLUMN_NAME”

^ – indication that new line character must be inserted here, which allows to address master-detail structure and convert a single record into multi lines layout.

! – ignore the record if all fields starting with 2nd column following “!” are empty

FILLER – reserved word to indicate blank field

2.7 OUTPUT DIRECTORY

Property name	Example
OUTPUT_DIR	OUTPUT_DIR=TMP

This property specifies DBA DIRECTORY name where output file is created. This must exist in DBA_DIRECTORIES table.

2.8 OUTPUT FILE NAME

Property name	Example
OUTPUT_FILE_NAME	OUTPUT_FILE_NAME=@MASK_VENDORS_LIST.csv

This property determines output file name. @MASK will be replaced with the auto-generated value explained below.

2.9 OUTPUT FILE NAME MASK

Property name	Example
@MASK	@MASK=DDMONYYYYHH24MISS Result: 12OCT2018221755 Filename will be created as 12OCT2018221755_VENDORS_LIST.csv

@MASK is part of the output file name and will be substituted according to the @MASK format string when file name is created. Format matches Oracle format masks.

2.10 RECORD FORMAT

RECORD_FORMAT defines how individual fields are formatted.

Property name	Example
RECORD_FORMAT	RECORD_FORMAT= . . R40 L30~0 .

“.” - default format – as defined in the QUERY (view)

“RNN” – R or L – right or left padded format, NN - field length

“~” - reserved separator character between padding direction + length elements and padded character

For example:

L20~0 with data value of 5 - 00000000000000000005

R20~X with data value of AAA – AAAXXXXXXXXXXXXXXXXXX

Record format line must contain the same number of fields as a header record.

2.11 SKIP IDENTICAL STRINGS

Property name	Example
SKIP_IDENTICAL_STRINGS	SKIP_IDENTICAL_RECORDS=YES/NO

SKIP_IDENTICAL_STRINGS property allows to ignore identical strings from the preceding record.

3 PLSQL API DEFINITION

```

PROCEDURE XXCUST_FILES_BUILDER_PKG.main (
    errbuf OUT VARCHAR2,
    errcode OUT NUMBER,
    filename VARCHAR2,
    filedir VARCHAR2,
    p1 VARCHAR2 DEFAULT NULL,
    d VARCHAR2 DEFAULT 'Y'
);

/* Errbuf Varchar2 – EBS concurrent program error message
   Errcode NUMBER – EBS concurrent program error code
       0 – Success
       1 – Warning
       2 – Error
   Filename Varchar2 - Configuration file name
   Filedir Varchar2 - DBA_DIRIRECTORY name where configuration file is located
   P1 Varchar2 – additional query parameters such as “rownum < 11 or
                                   dept_name='MYDEPARTMENT' ”
   D Varchar2 – debug flag. ‘Y’ - activate debug for concurrent program log file.
*/

```

4 REGISTER FLAG API AS A CONCURRENT PROGRAM IN ORACLE EBS

4.1 REGISTER PL/SQL API AS CONCURRENT PROGRAM EXECUTABLE.

Using System Administrator responsibility create concurrent program executable based on FLAG API – XXCUST_FILES_BUILDER_PKG.Main procedure.

Create individual concurrent program using the same executable for every flat file creation process. Follow the list of parameters as described in Chapter 3.

5 EXAMPLES

5.1 CONFIGURATION FILE EXAMPLE:

A. DDL for View XX_LOC_EMPLOYEES

```
CREATE OR REPLACE FORCE VIEW "APPS"."XX_LOC_EMPLOYEES" ("HEADER", "DETAIL",
"EMPLOYEE_NUMBER", "FULL_NAME", "LOCATION_CODE") AS
  select 'H' header,'D' detail,employee_number,full_name,location_code
from per_all_people_f p,hr_locations_all l,per_all_assignments_f a
where a.person_id=p.person_id
and l.location_id = a.location_id
and trunc(sysdate) between a.effective_start_date and a.effective_end_date
and trunc(sysdate) between p.effective_start_date and p.effective_end_date
order by location_code;
```

5.1.1 PL/SQL code snippet

```
/***** Automation of flat files **** START */
declare
  errbuf Varchar2(100);
  errcode Varchar2(100);
  fn Varchar2(100):='LOC-EMPS.cfg';
  fd Varchar2(100):='TMP';
Begin
  xxcust_files_builder_pkg.main (errbuf,errcode,fn,fd);
End;
/***** Automation of flat files **** END */
```

A. Configuration File - LOC-EMPS.cfg

```
PRINT_HEADER=NO
QUERY=select * from xx_loc_employees where rownum < 1001;
RECORD_LAYOUT=LOCATION_CODE|^EMPLOYEE_NUMBER|FULL_NAME
```



```

RECORD_FORMAT=|.|.
@MASK=DDMONYYYY_HH24MISS
OUT_DIR=TMP
OUT_FILE_NAME=VENDOR_@MASK.csv
DELIMITER=,
OPTIONAL_DELIMITER="
SKIP_IDENTICAL_STRINGS=YES

```

Location Code string is shown only once as determined by SKIP_IDENTICAL_STRINGS=YES
 Employee number starts on a new line as determined by ^ character.

Flat File output:

```
LOCATION_CODE|^EMPLOYEE_NUMBER|FULL_NAME
```

```

AD1-Vision A&D Final Assy Plant 1,
1274,"Stern, Mr. Frank"
1727,"Eldridge, Ms. Nickie"
1731,"Williams, Mr. Charlie"
1733,"Colton, Mr. Carl"
1736,"Alfonso, Ms. Hillarie"
1735,"Beckett, Mr. Henry"
1734,"Dely, Mr. Samuel"
1732,"Erdos, Mr. Connor"
1737,"Fort, Mr. Saul"
1746,"Brinton, Mr. Charles"
1745,"Colson, Ms. Taylor"
1744,"Billow, Mr. Leonard"
1743,"Delaware, Mr. Donald"
1742,"Cleasue, Mr. George"
1741,"Benjamin, Mr. Karl"
1740,"Fudio, Ms. Anatole"
1739,"Fosdi, Mr. Tony"
1738,"Jason, Mr. Robert"
1750,"Moore, Ms. Anna"
1749,"Stone, Mr. Michael"
1748,"Conton, Mr. Daniel"
1747,"France, Mr. Charles"
AD2-Vision A&D Final Assembly Plant 2,
1728,"Becker, Mr. John"
1751,"Brod, Mr. Johnny"
1752,"Fowler, Mr. Samuel"

```

1754,"Cogen, Mr. Richard"
 1769,"Allen, Ms. Barbara"
 1768,"Wilmont, Mr. Stephen"
 1767,"Worthy, Mr. Harrison"

5.1.2

B. Configuration File - LOC-EMPSB.cfg

```
PRINT_HEADER=YES
QUERY=select * from xx_loc_employees where rownum < 1001;
RECORD_LAYOUT=LOCATION_CODE|FILLER|EMPLOYEE_NUMBER|FULL_NAME
RECORD_FORMAT=|.|.
@MASK=DDMONYYYY_HH24MISS
OUT_DIR=TMP
OUT_FILE_NAME=VENDOR_@MASK.csv
DELIMITER=,
OPTIONAL_DELIMITER="
SKIP_IDENTICAL_STRINGS=NO
```

Resulting flat file -

Location Code is written on every record as determined by SKIP_IDENTICAL_STRINGS=NO
 Header record is visible based on the property PRINT_HEADER=YES

```
LOCATION_CODE,NA,EMPLOYEE_NUMBER,FULL_NAME
AD1-Vision A&D Final Assy Plant 1,,1274,"Stern, Mr. Frank"
AD1-Vision A&D Final Assy Plant 1,,1727,"Eldridge, Ms. Nickie"
AD1-Vision A&D Final Assy Plant 1,,1731,"Williams, Mr. Charlie"
AD1-Vision A&D Final Assy Plant 1,,1733,"Colton, Mr. Carl"
AD1-Vision A&D Final Assy Plant 1,,1736,"Alfonso, Ms. Hillarie"
AD1-Vision A&D Final Assy Plant 1,,1735,"Beckett, Mr. Henry"
AD1-Vision A&D Final Assy Plant 1,,1734,"Dely, Mr. Samuel"
AD1-Vision A&D Final Assy Plant 1,,1732,"Erdos, Mr. Connor"
AD1-Vision A&D Final Assy Plant 1,,1737,"Fort, Mr. Saul"
AD1-Vision A&D Final Assy Plant 1,,1746,"Brinton, Mr. Charles"
AD1-Vision A&D Final Assy Plant 1,,1745,"Colson, Ms. Taylor"
AD1-Vision A&D Final Assy Plant 1,,1744,"Billow, Mr. Leonard"
AD1-Vision A&D Final Assy Plant 1,,1743,"Delaware, Mr. Donald"
```

AD1-Vision A&D Final Assy Plant 1,,1742,"Cleasue, Mr. George"
AD1-Vision A&D Final Assy Plant 1,,1741,"Benjamin, Mr. Karl"
AD1-Vision A&D Final Assy Plant 1,,1740,"Fudio, Ms. Anatole"
AD1-Vision A&D Final Assy Plant 1,,1739,"Fosdi, Mr. Tony"
AD1-Vision A&D Final Assy Plant 1,,1738,"Jason, Mr. Robert"
AD1-Vision A&D Final Assy Plant 1,,1750,"Moore, Ms. Anna"
AD1-Vision A&D Final Assy Plant 1,,1749,"Stone, Mr. Michael"
AD1-Vision A&D Final Assy Plant 1,,1748,"Conton, Mr. Daniel"
AD1-Vision A&D Final Assy Plant 1,,1747,"France, Mr. Charles"
AD2-Vision A&D Final Assembly Plant 2,,1728,"Becker, Mr. John"
AD2-Vision A&D Final Assembly Plant 2,,1751,"Brod, Mr. Johnny"
AD2-Vision A&D Final Assembly Plant 2,,1752,"Fowler, Mr. Samuel"
AD2-Vision A&D Final Assembly Plant 2,,1754,"Cogen, Mr. Richard"
AD2-Vision A&D Final Assembly Plant 2,,1769,"Allen, Ms. Barbara"
AD2-Vision A&D Final Assembly Plant 2,,1768,"Wilmont, Mr. Stephen"
AD2-Vision A&D Final Assembly Plant 2,,1767,"Worthy, Mr. Harrison""

6 CONTACT INFORMATION:

iTerra Consulting Inc. Denver CO, USA

Email: support@migrationlogik.com

Phone: +1-303-324-3328