

Using Macros and User Tokens

by Ahmed Baraka

Introduction to Oracle Data Guard

In this lecture, we are going to talk about the basic concepts of Oracle Data Guard

Objectives

By the end of this lecture, you should be able to:

- Use macros
- Use User Tokens



Macros Overview

- A set of reusable parameters, commands, and functions
- Used to simplify and automate the work
- Can be used for:
 - Reusable code
 - Eliminating redundant column specifications
 - Creating libraries and share across multiple parameter files
- Using macros: define macros, call macros

Defining Macros

- Can be defined in a parameter file or library
- Macro name must begin with macro character (#)
- Parameter names must begin with macro character



Ahmed Baraka
Oracle Database Administrator

```
Macro #<macro_name>  
Params (#<param1>, #<param2>, ...)  
BEGIN  
<macro_body>  
End;
```

Calling a Marco

- Calling a macro syntax:

```
[target =] macro_name (val[, ...])
```



Ahmed Baraka
Oracle Database Administrator

- Examples:

```
DATECOL1 = #make_date (YR1, MO1, DAY1)
```

```
#option_defaults ()
```

A Macro Example: Returning a Value

```
Macro #make_date
Params (#year, #month, #day)
BEGIN
  @DATE ('YYYY-MM-DD', 'YY', #year, 'MM', #month,
  'DD', #day)
End;
```



```
Map SALES.INVOICE, Target REPORT.INV,
ColMap
( TargetCol1 = SourceCol1,
  Order_Date = #make_date(ORD_YR,ORD_MO,ORD_DAY),
  Inv_Date = #make_date(INV_YR,INV_MO,INV_DAY)
);
```

A Macro Example: Reusing Parameter Sets

```
Macro #option_defaults
Begin
  GetInserts
  GetUpdates
  GetDeletes
  InsertDeletes
End;
```

```
#option_defaults ()
IgnoreUpdates
Map HR.SRCTAB, Target HR.TARGETAB;
#option_defaults ()
Map HR.SRCTAB2, Target HR.TARGETAB2;
```

Creating Macro Library

- File (with mac extension) which has one or macros
- Use `INCLUDE` parameter at the beginning of a parameter file
- `LIST` and `NOLIST` parameters to turn the listing on and off

```
INCLUDE /ggs/dirprm/hrlib.mac
```



Tracing Macro Expansion

- Trace macro expansion with the CMDTRACE parameter

```
CMDTRACE [ON | OFF | DETAIL]
```

- When enabled, the macro processor displays macro expansion steps in the process's report file.

```
. . .  
CmdTrace ON  
Map TEST.TABLE1, Target TEST.TABLE2,  
ColMap (#testmac);  
CmdTrace OFF
```

Using Tokens

- Information saved within the *user token* area of a trail record header.

```
TABLE table_spec,  
  TOKENS (token_name = token_data [, ...]);
```

- Token data is a character string of up to 2000 bytes
- Populate target columns through a `Map` or `ColMap` clauses and `@TOKEN` function

```
COLMAP (target_column = @TOKEN ('token_name'))
```

Using Tokens Example

```
TABLE ora.oratest, TOKENS (  
  TK-OSUSER = @GETENV ('GGENVIRONMENT' , 'OSUSERNAME'),  
  TK-GROUP = @GETENV ('GGENVIRONMENT' , 'GROUPNAME')  
  TK-HOST = @GETENV('GGENVIRONMENT' , 'HOSTNAME'));
```

```
Map ora.oratest, Target ora.oratest,  
ColMap (USEDEFAULTS,  
  OSUSER = @TOKEN('TK-OSUSER'),  
  ENVGROUP = @TOKEN('TK-GROUP'),  
  HOSTNAME = @TOKEN('TK-HOST'));
```

Summary

In this lecture, you should have learnt how to:

- Use macros
- Use User Tokens

