CSF REUSE SPECIFICATIONS

Available Functions

The CSF Reuse library contains several packages and functions for use by the DCII application. These packages include:

- Error Logging
- Error Message Retrieval
- PL/SQL Timing routines
- PL/SQL Debugging routines
- Site Sys interface routines
- Date Count Utilities.

Batch Error Logging

Below you will find information and instructions on how to use the batch mode of the error logging routines.

How to use the packages

Package CSF_KM0100 is used to initialize error session, record errors, and close error session in batch, interface, and non-user interfere applications. Use as follows:

Call CSF_KM0100.init_error_log to open and get the error session id.
Log errors using error session id by calling CSF_KM0100.record_error.
Call CSF_KM0100.close_error_log to close Error Session

CSF_KM0100 calls CSF_KS0200.get_message_text. No external call is needed. Error Session ID is needed in order for Error to be recorded to the Error Log tables.

Function Parameter List

Package: CSF_KM0100
Procedure: Init_Error_log
Return Value: Returns Error Session Id for error logging in P_SESSION_ID
Purpose: Called to initialize Error log session.

Name	Parameter Type	Datatype	Description
P_MSG_PREFIX	IN	VARCHAR2(30)	Application Message Prefix - Used to identify Error Application attempting to initialize an error session.
P_NAME	IN	VARCHAR2(61)	Routine Name - Used to identify Application procedure (routine) recording the errors
P_TYPE	IN	VARCHAR2(10)	Routine Type - Used to Identify the type of routine reporting the errors.
P_SESSION_ID	OUT	NUMBER	Unique ID for this “error session” returned to program
P_PURGE_IND	IN	VARCHAR2	Not used

Usage of Common Service Functions

Package: CSF_KM0100
Procedure: Record_Error
Return Value: None.
Purpose: Called to insert Error Messages into Error Log table
Associated Calls: CSF_KS0200.get_message_text to get DCD error messages.

Name	Parameter Type	Datatype	Description
P_SESSION_ID	IN	NUMBER	Unique ID for this “error session”
P_ERR_CD	IN	VARCHAR2(100)	Concatenated DC_ERROR_MESSAGES identification: ERRAPP_MESSAGE_APP_PREFIX + ':' + MESSAGE_NUMBER
P_ERR_MSG	IN	VARCHAR2(2000)	Defined by DC_ERROR_LOGS.ERROR_MESSAGE, in order to use DEFAULT value for parameter, which allows it to be optional, must be defined explicitly and not be reference.
P_SQL_STMT	IN	NUMBER	Not in use
P_ERR_FROM_UI	IN	BOOLEAN	Indicator flag if RECORD ERROR request originates from an interactive User Interface (Not in use)

Package: CSF_KM0100
Procedure: Close_Error_Log
Return Value: None.
Purpose: Called to close Error Session.

Name	Parameter Type	Datatype	Description
P_ERRAPP_SESSION_ID	IN	NUMBER	Unique ID for this “error session”

Package: CSF_KS0200
Function: Get_Message_Text
Return Value: DCD Error message as VARCHAR2.
Purpose: Returns DCD error message to the application

Name	Parameter Type	Datatype	Description
P_MSG_PREFIX	IN	VARCHAR2(5)	The application system prefix for identifying ownership of a message set.
P_ERR_NBR	IN	NUMBER(8)	Unique number identifying an individual message

Package: CSF_KS0200
Procedure: Get_Message_Text
Return Value: DCD Error message in P_MSG_REC.
Purpose: Returns DCD error message and places on the stack

Name	Parameter Type	Datatype	Description
P_MSG_PREFIX	IN	VARCHAR2(5)	The application system prefix for identifying ownership of a message set.

Usage of Common Service Functions

P_ERR_NBR	IN	NUMBER(8)	Unique number identifying an individual message
P_MSG_REC	MOD (IN/OUT)	CSF_KS0200.V_MSG_REC_T	Variable to return fetched error message records

Returning errors to Applications

Below you will find instructions on how to have errors returned to the application.

CG$FORM_ERRORS.PUSH (client side) and CG$ERRORS.PUSH (server side)

Both of these packages are Oracle Packages that have been modified for DCD use. The modifications are done to allow interception of default Oracle error messages and replace the default error message with defined DCD error messages which have more meaning.

CG$FORM_ERRORS.PUSH

Name	Parameter Type	Datatype	Description
Msg	In	Varchar2	Application prefix concatenated with error number. **Note: Each must be followed by a colon.**
Error	In	Varchar2	Error message severity indicator
Msg_type	In	Varchar2	Error message type (OFG)
Msgid	In	Number	Error message id (0)

CG$FORM_ERRORS.PUSH is located in the DCDLIB01 and is called when a form error occurs. The procedure will retrieve, using csf_ks0200.get_message_text, the client side error message from the DCD error message table and display it in a pop-up alert window. If the error message is not found in the error table, the msg passed into the msg parameter of the procedure will be displayed in the hint bar at the bottom of the window.

Ex - CG$FORM_ERRORS.PUSH(‘CSF:20001:’, ‘I’,’OFG’0)
Note: The two colons in the msg parameter are very important.

CG$ERRORS.PUSH is located in the repository and is called by Table API upon a table error. The error message is retrieved from the server side and returned to the calling application, whether a form or batch process. This one is incorporated into all table APIs for insert, delete, update, and row locks.

If the custom DCD error messages are in the table at the time of compilation, these functions are all automatic, otherwise the application designers will have to manually call the errors.

Debug and Timing Routines

How to use the packages

Package DCD_DEBUG is used to toggle on/off the printing of debug messages within a PL/SQL package / procedure. The PL/SQL module should place calls to DCD_DEBUG.print throughout the program. When it is desired to see the debug messages, place a call to DCD_DEBUG.enable in the module. When it is no longer desired to see the debug messages, place a call to DCD_DEBUG.disable. DCD_DEBUG.enable and DCD_DEBUG.disable do not stack. So calling DCD_DEBUG.disable will immediately cease all debug messages.

Call DCD_DEBUG.enable to enable the printing of debug messages.
Call DCD_DEBUG.print(msg) to print a message using DBMS_OUTPUT.PUT_LINE.
Call DCD_DEBUG.disable to disable the printing of debug messages.

Package DCD_TIMING is used to record the amount of clock time that a PL/SQL module spends on a defined block of code. Call DCD_TIMING.enable to enable the clock timer. Call DCD_TIMING.start_timer to record the current time. Call DCD_TIMING.stop_timer(desc) to stop the timer and insert the Start time, End time, and Description in the DCII_TIMING table. Call DCD_TIMING.disable to disable any future calls to start_timer and stop_timer. Only one timer can be running at a time.

Call DCD_TIMING.enable to enable the start_timer and stop_timer calls.
Call DCD_TIMING.start_timer to record the current time and hold it in a variable.
Call DCD_TIMING.stop_timer to record the current time, and log the start time, end time, and description in the DCII_TIMING table.
Call DCD_TIMING.disable to disable the start_timer and stop_timer calls.

Function Parameter List

Package: DCD_DEBUG
Procedure: Enable
Parameters: None
Return Value: None
Purpose: Enable the output of debug messages

Package: DCD_DEBUG
Procedure: Disable
Parameters: None
Return Value: None
Purpose: Disable the output of debug messages

Package: DCD_DEBUG
Procedure: Print
Return Value: None
Purpose: Print out a message using DBMS_OUTPUT.PUT_LINE

Name	Parameter Type	Datatype	Description
Msg	In	Varchar2	Message to be displayed

Package: DCD_TIMING
Procedure: Enable

Usage of Common Service Functions

Parameters: None
Return Value: None
Purpose: Enable calls to the start_timer and stop_timer procedures.

Package: DCD_TIMING
Procedure: Disable
Parameters: None
Return Value: None
Purpose: Disable calls to the start_timer and stop_timer procedures.

Package: DCD_TIMING
Procedure: Start_Timer
Parameters: None
Return Value: None
Purpose: Record the current time in a variable.

Package: DCD_TIMING
Procedure: Stop_Timer
Parameters: None
Return Value: None
Purpose: Record the current time, and log the Start Time, End Time, and Description in the DCII_TIMING table.

Name	Parameter Type	Datatype	Description
Description	In	Varchar2	Message to be displayed. Record will be truncated at 255 bytes. Calls with description longer than 1000 bytes may result in a run-time error.

CSF Utilities

How to use the packages

Package CSF_KUTIL contains several functions to support calls to the Site Sys tables and to count days, and holidays:

Get_Site_Sys(For_System, For_Site)
Get_Site_Sys(For_Id)
Get_Service(Of_System)
Count_Holidays(Start_date, End_date)
Count_Weekends(Start_date, End_date)
Bus_days(Start_date, End_date, St_Switch)

Function Parameter List

Return value type Site_Sys_Rec:

```
type Site_Sys_Rec is
   record
   (Id                 Number,
    System             Varchar2(16),
    Site               Varchar2(25),
    Service            Varchar2(1),
    Parent_Ssys_Id     Number
    Db_Id              Varchar2(4),
    Active             Varchar2(1));
```

Package: CSF_KUTIL
Function: Get_Site_Sys
Return Value: Site_Sys_Rec
Purpose: Retrieve the designated Site Sys information as outlined above specifying the System and Site codes.

Name	Parameter Type	Datatype	Description
For_System	IN	VARCHAR2(16)	Source System
For_Site	IN	VARCHAR2(25)	Source Site

Package: CSF_KUTIL
Function: Get_Site_Sys
Return Value: Site_Sys_Rec
Purpose: Retrieve the designated Site Sys information as outlined above specifying the Id of the record.

Name	Parameter Type	Datatype	Description
For_Id	IN	NUMBER	Site System ID

Package: CSF_KUTIL
Function: Get_Service
Return Value: Service Code (VARCHAR2)
Purpose: Retrieve the service code specifying the Source System.

Name	Parameter Type	Datatype	Description
Of_System	IN	VARCHAR2	Source System

Usage of Common Service Functions

Package: CSF_KUTIL
Function: Count_Holidays
Return Value: Number of Holidays between the two dates (NUMBER)
Purpose: Count the number of Holidays between two dates.

Name	Parameter Type	Datatype	Description
Start_Date	IN	DATE	Start date of the period
End_Date	IN	DATE	End date of the period

Package: CSF_KUTIL
Function: Count_Weekends
Return Value: Number of Weekend days (SAT, SUN) between the two dates (NUMBER)
Purpose: Count the number of weekend days between two dates.

Name	Parameter Type	Datatype	Description
Start_Date	IN	DATE	Start date of the period
End_Date	IN	DATE	End date of the period

Package: CSF_KUTIL
Function: Bus_Days
Return Value: Number of business days between the two dates (NUMBER)
Purpose: Count the number of business days between two dates.

Name	Parameter Type	Datatype	Description
Start_Date	IN	DATE	Start date of the period
End_Date	IN	DATE	End date of the period
St_Switch	IN	VARCHAR2(1)	Always 1. Not currently implemented.