

Oracle FLEXCUBE Investor Servicing®
Support By Example

Release 12.0

April 2012

Oracle Part Number E51528-01

Contents

1	Preface.....	3
1.1	Audience	3
1.2	Related documents	3
1.3	Conventions.....	4
2	Introduction.....	4
2.1	How to use this Guide	4
3	Support - Getting started	5
3.1	Support process flow	5
3.2	Debugs in FLEXCUBE.....	6
3.3	How to debug in FLEXCUBE.....	8
3.3.1	<i>Client Side Debugging</i>	8
3.3.2	<i>Front End Debugging</i>	12
3.3.3	<i>Back End Debugging</i>	13
4	Support Examples	14
4.1	FLEXCUBE Host(Online)	14
4.1.1	<i>Issue Raised:</i>	14
4.1.2	<i>Inference:</i>	14
4.1.3	<i>Debugging:</i>	15
4.1.4	<i>Bug:</i>	15
4.1.5	<i>Fix provided:</i>	16
4.2	FLEXCUBE Host (Maintenance).....	16
4.2.1	<i>Issue Raised</i>	16
4.2.2	<i>Inference</i>	16
4.2.3	<i>Debugging:</i>	16
4.2.4	<i>Bug</i>	17
4.2.5	<i>Fix Provided</i>	17
4.3	FLEXCUBE Branch.....	Error! Bookmark not defined.
4.3.1	<i>Issue Raised</i>	Error! Bookmark not defined.
4.3.2	<i>Inference</i>	Error! Bookmark not defined.
4.3.3	<i>Debugging:</i>	Error! Bookmark not defined.
4.3.4	<i>Bug</i>	Error! Bookmark not defined.
4.3.5	<i>Fix Provided</i>	Error! Bookmark not defined.
4.4	FLEXCUBE Gateway	17
4.4.1	<i>Issue Raised</i>	17
4.4.2	<i>Inference:</i>	17
4.4.3	<i>Debugging:</i>	18
4.4.4	<i>Bug:</i>	18
4.4.5	<i>Fix provided:</i>	18
4.4.6	<i>Bug:</i>	19
4.4.7	<i>Fix provided:</i>	19
4.5	EOD in FLEXCUBE.....	19
4.5.1	<i>Issue Raised</i>	19
4.5.2	<i>Inference</i>	19
4.5.3	<i>Debug</i>	19
4.5.4	<i>Bug</i>	20
4.5.5	<i>Fix Provided</i>	20
5	Glossary	20
6	List of Examples	21
7	List of Tables	21
8	List of Figures	21

1 Preface

This document is intended to serve FLEXCUBE development and support teams. This document can be used by the teams to get an idea on how to go about finding an issue in FLEXCUBE and how to fix it.

1.1 Audience

The Support getting started book is intended for FLEXCUBE Application Developers who are authorized to perform the following tasks:

- FLEXCUBE IS Application component development
- FLEXCUBE IS Application implementation
- FLEXCUBE IS Application software support

1.2 Related documents

For more information, see the following documents.

1. FCIS-FD01-01-01-Development Overview Guide
2. RAD
 - a. FCIS-FD02-01-01-RAD Getting Started
 - b. FCIS-FD02-02-01-RAD Function ID Development Volume 1
 - c. FCIS-FD02-02-01-RAD Function ID Development Volume 2
 - d. FCIS-FD02-03-01-RAD Web Service Development
 - e. FCIS-FD02-04-01-RAD BIP Report Integration
 - f. FCIS-FD02-05-01-RAD Notification Development
3. Extensibility
 - a. FCIS-FD03-01-01-Extensibility Getting started
 - b. FCIS-FD03-02-01-Extensibility Reference Guide
 - c. FCIS-FD03-03-01-Extensibility By Example Volume 1
 - d. FCIS-FD03-03-02-Extensibility By Example Volume 2
4. Interface
 - a. FCIS-FD04-01-01-Interface Getting started
 - b. FCIS-FD04-02-01-Generic Interface Configuration Guide
 - c. FCIS-FD04-03-01-Upload Adapter Development Guide
5. Tools
 - a. FCIS-FD05-01-01-Tools-Getting Started
 - b. FCIS-FD05-02-01-RAD-Reference
 - c. FCIS-FD05-02-02-RAD-Installation and Setup
 - d. FCIS-FD05-03-01-DDL-Reference
 - e. FCIS-FD05-04-01-TrAX-Reference
6. Support
 - a. FCIS-FD06-01-01-Support Getting started
 - b. FCIS-FD06-02-01-Support By Example

7. Reports

- a. FCIS-FD07-01-01-Report Getting started
- b. FCIS-FD07-02-01-BIP Report Development Guide
- c. FCIS-FD07-03-01-OBIEE repository Development Guide

1.3 Conventions

The following text conventions are used in this document:

Convention Meaning

boldface	Boldface type indicates graphical user interface elements (for example, menus and menu items, buttons, tabs, dialog controls), including options that you select.
<i>italic</i>	italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates language and syntax elements, directory and file names, URLs, text that appears on the screen, or text that you enter.

2 Introduction

This document is prepared to help users who are new to FLEXCUBE on how to go about debugging in case of issues.

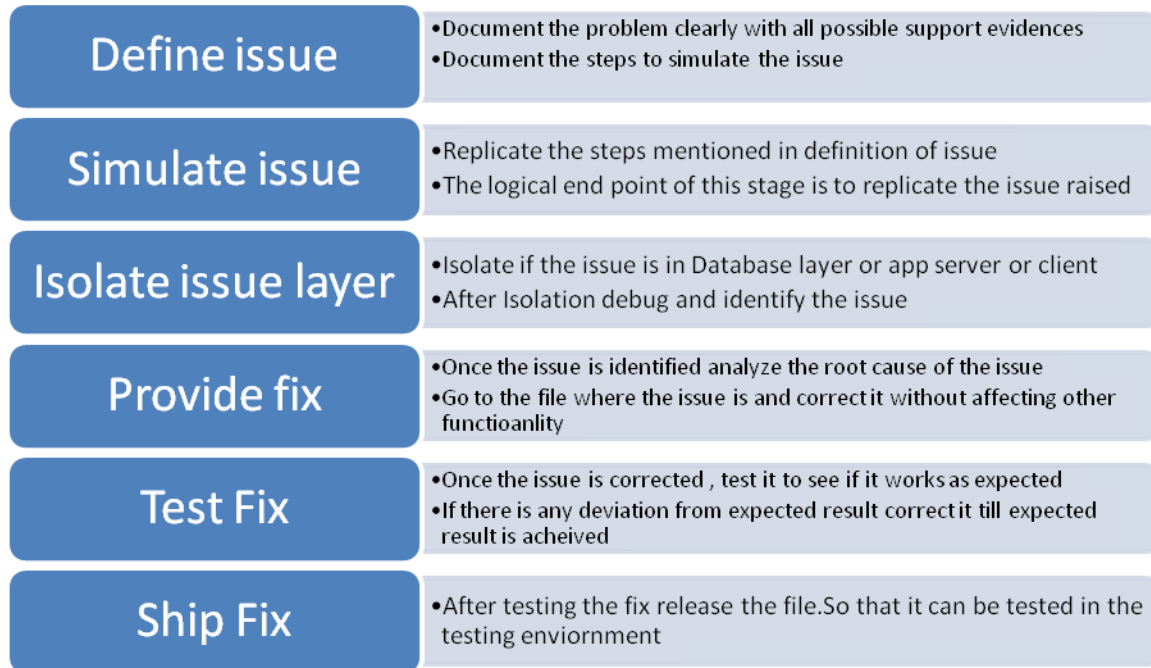
2.1 How to use this Guide

The information in section includes

- Chapter 2, "Introduction"
- Chapter 2, "Support Getting started"
- Chapter 2, "Support - Examples"

3 Support - Getting started

3.1 Support process flow



3.2 Debugs in FLEXCUBE

There are different FLEXCUBE debugs which as a supporter or a developer should be aware of. Listed below are the different debugs in FLEXCUBE.

<i>Type of operations</i>	<i>Front End Debug Applicable</i>	<i>Front debug Area</i>	<i>Backend messaging schema debug Applicable</i>	<i>Backend business schema debug Applicable</i>	<i>Messaging schema debug area</i>	<i>Business schema debug area</i>
Host Online	Y	host debug path mentioned in fcubs.properties	N	Y	NA	WORK_AR EA in CSTB_PAR AM
Host Maintenance	Y	host debug path mentioned in fcubs.properties	N	Y	NA	WORK_AR EA in CSTB_PAR AM
Branch	Y	branch debug path mentioned in fcubs.properties	N	Y	NA	WORK_AR EA in CSTB_PAR AM
EOD	N	NA	N	Y	NA	WORK_AR EA in CSTB_PAR AM
Gateway	Y	debug area mentioned in gateway properties xml	Y	Y	DEBUG_AR EA in GWTB_PAR AMETERS	WORK_AR EA in CSTB_PAR AM

Table 1 : Debugs in FLEXCUBE

<i>Debug</i>	<i>Name Pattern</i>	<i>Server where it is present</i>	<i>Overwrite/Append/New</i>
Front end host	<username>+<branchcode>.log	Appserver	Append
Front end branch	<username>+<branchcode>.log	Appserver	Append
Back end messaging schema	FCUBS_GW_<uniqueid>.txt	Database server	New
Back end business schema	<username>+<branchcode>.txt	Database server	Overwrite
Gateway Debug	As mentioned in logger.xml	Appserver	New

Table 2: Debug Information

- FLEXCUBE back end debug (Business Schema)
 - This area should be a folder in the database server. The area information should be maintained in CSTB_PARAM table for the PARAM_NAME “WORK_AREA”
- FLEXCUBE back end debug(Messaging Schema)
 - This debug is applicable only for FLEXCUBE Gateway transactions. This area should be a folder in the database server. The area information should be maintained in GWTB_PARAMETERS table for the PARAM_NAME “DEBUG_PATH”
- FLEXCUBE front end Host debug
 - This area should be a folder in the Application Server. The area information should be maintained in the properties file of the FLEXCUBE application. This can be maintained while creating the properties file using the Installer. This debug files would contain information about the Host Transactions that is being carried out by a FLEXCUBE user.
- FLEXCUBE front end Branch Debug
 - This area should be a folder in the Application Server. The area information should be maintained in the properties file of the FLEXCUBE application. This can be maintained while creating the properties file using the Installer. This debug files would contain information about the Host Transactions that is being carried out by a FLEXCUBE user.
- FLEXCUBE Gateway Debug
 - This area should be a folder in the application server. The area information should be maintained in the properties file of the FLEXCUBE Gateway application. This can be maintained while creating the properties file using the Installer. This debug files would contain information about the Gateway Transactions that is being carried out by the external system.

3.3 How to debug in FLEXCUBE

3.3.1 Client Side Debugging

Client Side debugging is used if there is any issue which is occurring at Client Side (Browser). These issues could be issues occurred in the code present in the function ID JS files or common JS files. There are different tools using which we can debug based on the browser we are using.

- Browser Java Script debugging (F12 in IE8)
- Visual studio Interdev in < IE8
- Firefox - Firebug
- Other browser debugging

In all the above mentioned tools we would be enabling script debugging and either add a line of code “Debugger;” in JS file where we want to begin debugging or bookmark a point on the JS file in the script editor from which you want to start debugging.

When any operation which has to be debugged is carried out and when the control comes to the point marked we begin our debugging using one of the operations mentioned below on the script editor.

- Step into (Get into the function if the current line of code is a function)
- Step over (Go to next line of code)
- Step out (Go out of the function which you are currently in to the next line of code)

By repeating the above mentioned operations we can reach the place where the issue is. The different features in the script editor like Watch , Intermediate etc will help us in getting the values for a variable in the JS .To add more debugs we can use functions like alert() and msgbox() in JS.

Apart from this we can also use Debug Window option in FLEXCUBE to view Request and Response XML and more.

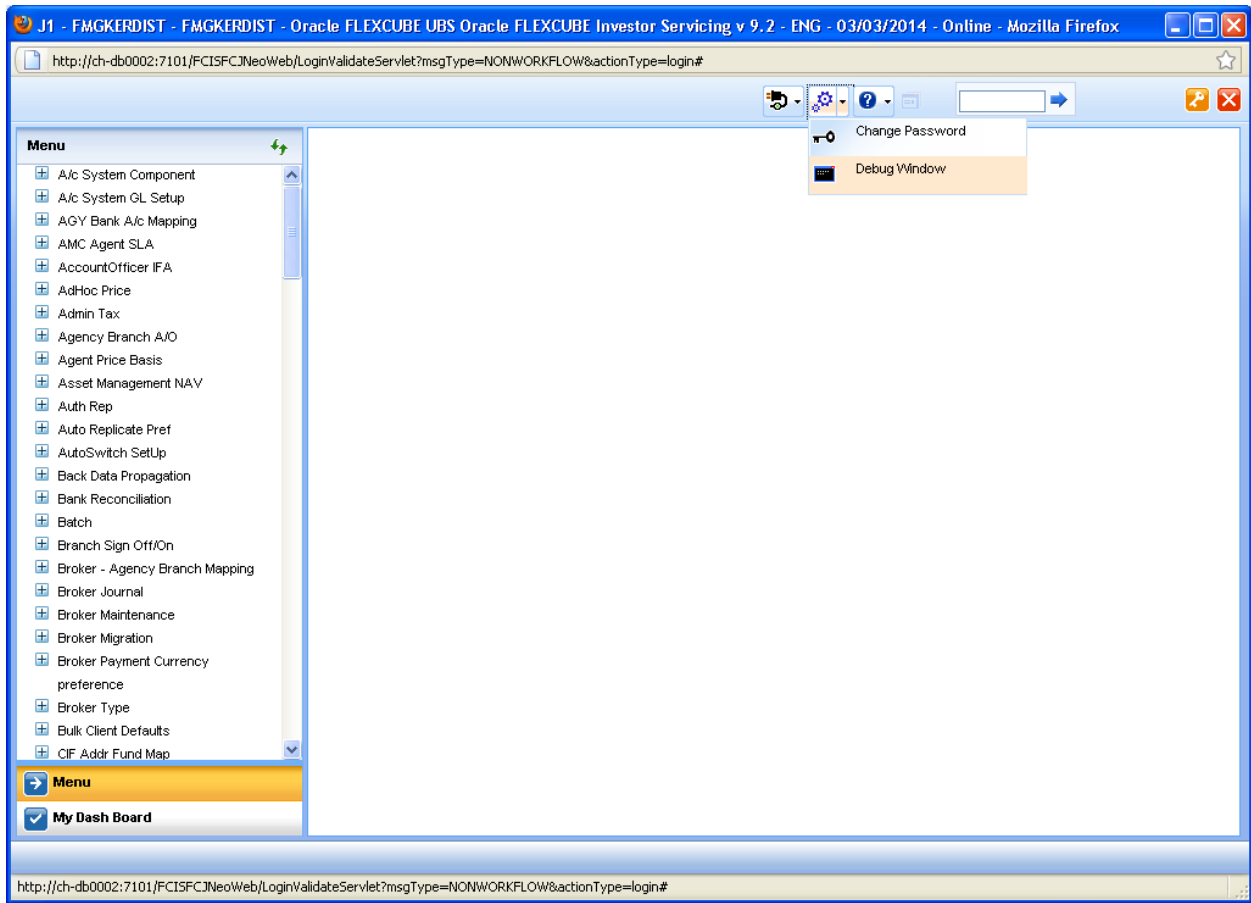


Figure 1 : Debug Window Menu

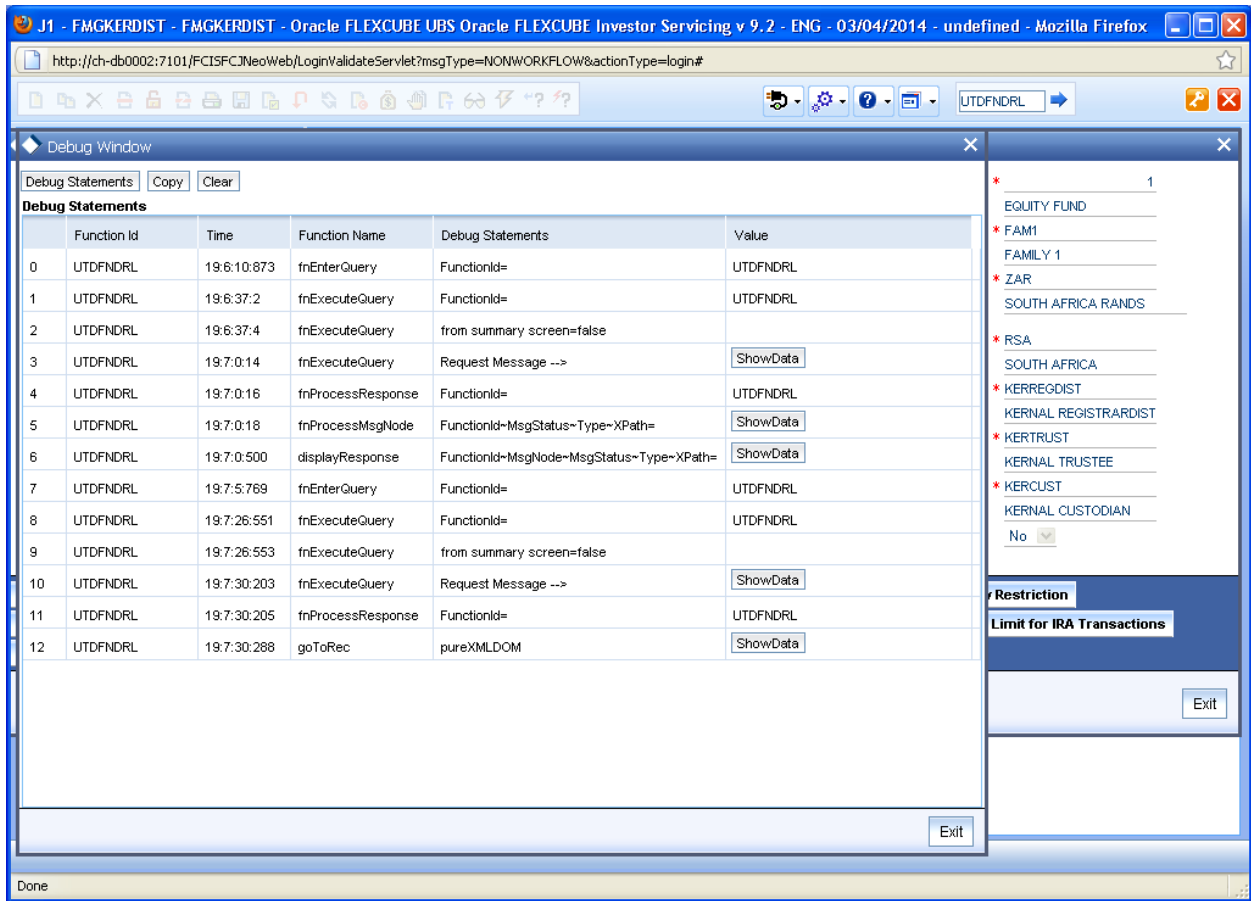


Figure 2 : Debug Window Main

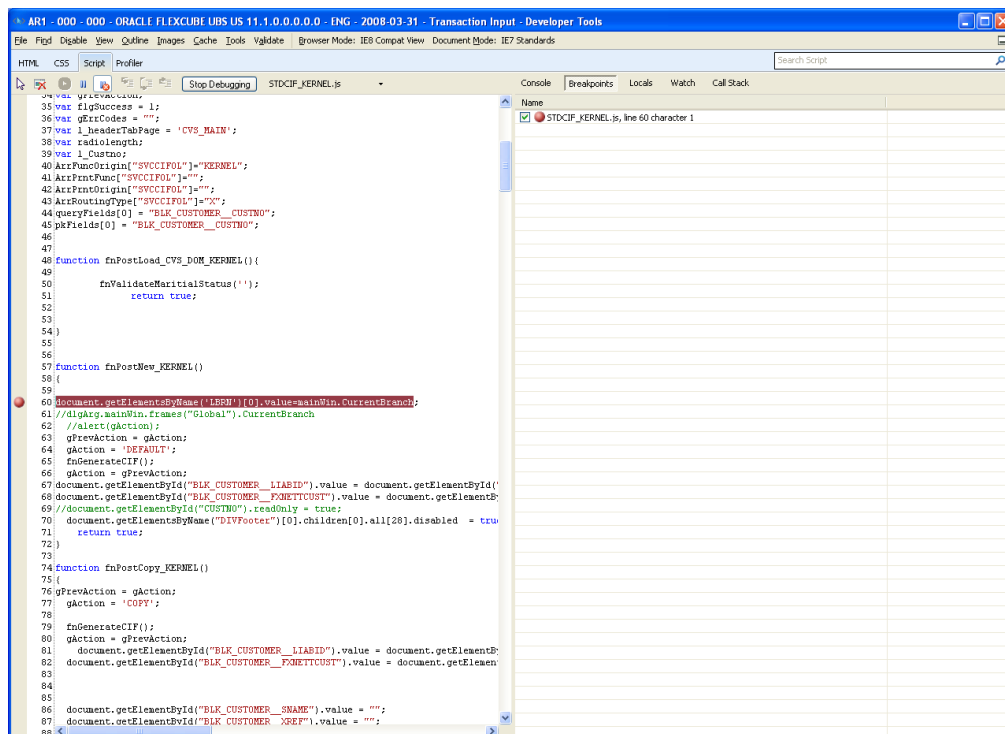


Figure 3: Setting Up Break point

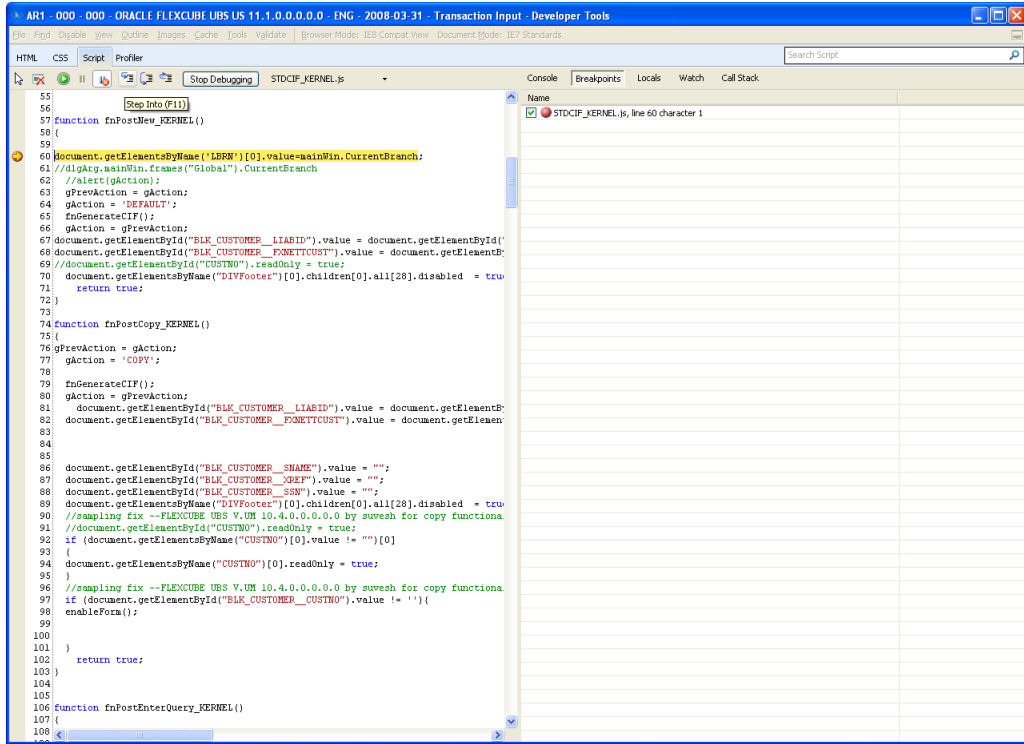


Figure 4: Using Step into option

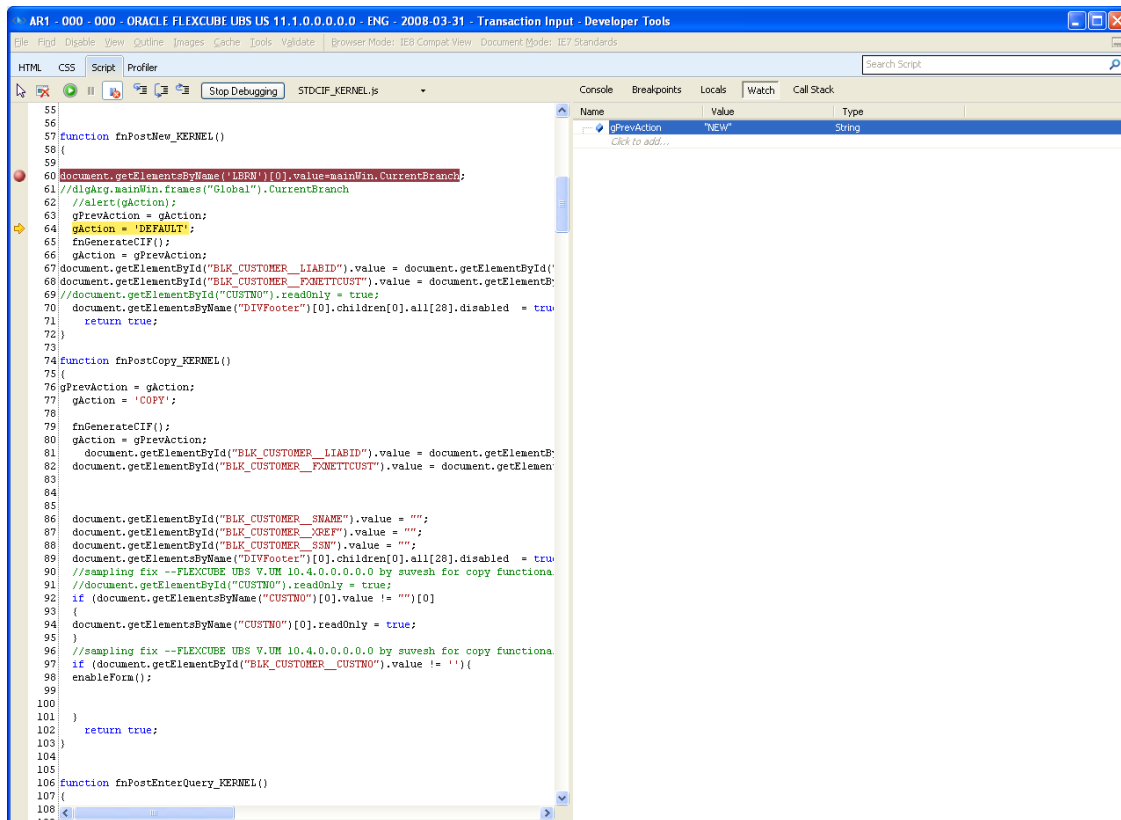


Figure 5: Monitoring values in Watch window

3.3.2 Front End Debugging

Front End debugging is used if there is any issue which is occurring at Front End in FLEXCUBE (Java Layer). These issues could be issues occurring in the code present in the Infra of FLEXCUBE (.java files). There are two ways of debugging this

- Through an IDE
- Log files written in the Server

Through an IDE

IDE is Integrated Development Environment it can be a tool like JDeveloper or ECLIPSE. The FLEXCUBE application is deployed and run in the IDE and as mentioned in the previous section we bookmark the point from which we would like to start debugging. Same process of debugging in script editor applies here.

Log files written in the Server

In this type of debugging we look at different log files written in the server. By going through this log file we would be able to trace an issue and the area of occurrence. Once the issue is figured out the code causing the issue can be corrected, redeployed and tested. If we need to write more debugs we can use `System.out.println()` and `printStackTrace()` to trace any error in a Catch statement block.

There are 3 debugs which will be of interest to us. In this case

- Host Front End Debug(Path is mentioned in Property File)
- Branch Back End Debug(Path is mentioned in Property File)
- Gateway log(EJB,Webservices,MDB,Servlet)(Path mentioned in property xml and logger xml)
- Application Server Debug(Path varies from one Application server to another)

One point that should be noted is the area where the debugs are expected to be written in the server should be given proper permissions. If the application server is in Unix and to access the logs from Windows one can use tools like WinSCP or Putty.

3.3.3 Back End Debugging

Back End debugging is used if there is any issue which is occurring at Back End in FLEXCUBE (PLSQL Layer). These issues could be issues occurring in the code present in FLEXCUBE.

The debugging in Back end is through the going through logs written in server. The debug area where the debugs are expected to be written in the server should be given proper permissions. . If the application server is in Unix and to access the logs from Windows one can use tools like WinSCP or Putty.

The debug is written in FLEXCUBE using `UTL_FILE` feature of PLSQL. For a debug to be written in FLEXCUBE the module for which the debug is written should have an entry in `CSTB_DEBUG` with value Y and similarly the user for which the debug has to be written should have an entry in `CSTB_DEBUG_USERS` with a value Y. The debugs are written using Package level debug statements. There could be three reasons why the debug is not writing

- No proper permissions in Server
- No space in server
- User Debug is locked

- In this case execute these 3 statements twice. These are procedures in FLEXCUBE and this will work only for FLEXCUBE debugs.
 - Exec global.pr_init('BRNCHCODE','USER NAME');
 - Exec debug.pr_debug('MODULE CODE','TEXT');
 - Exec debug.pr_close();

Debugging using log files can be a tedious process. There is no proper defined mechanism. It will vary from person to person.

4 Support Examples

In this document we would be focusing on five types of issues in FLEXCUBE and how to go about debugging and fixing those. The five types are

- FLEXCUBE Host (Online)
- FLEXCUBE Host(Maintenance)
- FLEXCUBE Gateway
- EOD in FLEXCUBE

Each type of issues as mentioned above will be split into 5 parts: Issue Raised, Inference, Debugging, Bug and Fix Provided. We will see one by one in the following sections.

4.1 FLEXCUBE Host(Online)

4.1.1 Issue Raised:

Child subscription Transaction when queried 0220120460000079 (linked), server processing failed is coming.

4.1.2 Inference:

Looking at the issue the first inference was there is some problem with the response xml or response tilde separated list getting build.

4.1.3 Debugging:

There are different debugs which would be of use to this kind of issue.

Request and Response XML's from Debug Window in FLEXCUBE application

Click on Options in menu of FLEXCUBE application and select DEBUG from the sub menu. A new window opens now query for the contract details which has been mentioned in the error. In the debug window you will be able to see two buttons Request Xml and Response Xml on click of this buttons it will open the Request XML which has been send to the backend and Response XML which has come from backend.

Check the request xml whether the request is built properly and all the information is correctly send. Similarly check if the response xml is built properly. Once you make sure the request and response are send properly move to the next step. In this case the request xml seemed fine and response xml was coming wrong. So an error has occurred at the back end or front end. This case needs further investigation.

Front End Host Debugs for any Exceptions Caught

Since this is a Host function id check in the front end host debug for any errors or exceptions which are caught, if there are any errors or exceptions go to the place where the error has occurred and find the reason for the error.

In this case there is no error caught in the Front end host debugs, this case needs further investigation.

FLEXCUBE Backend Debug

This is the most important debugs of all the debugs because this debugs not only has traces of any technical errors which has occurred but also has information of functional errors if any. This debug is also the last resort in identifying any error if the errors are not caught in the previous steps. Checking this debug could be a tedious process. If there is any error code you are looking for you can search the first occurrence of that particular error code in the debug file and you can proceed finding the occurrence of that. In case if you are looking for any miscalculations or a tag getting populated with a wrong value you will have to go through the debug line by line from the point which you think the value could have gone missing.

In this case since we are looking for a Server Processing Failed error we are seeing if there is any error or exception caught because of which the response xml was not built properly. We have found an error and this seems to be the reason why the xml is not built properly we will discuss about the error in the next section in detail.

4.1.4 Bug:

In the back end debug there was an oracle exception which was caught. *Numeric or Value error*. It was clear that there was a problem in tilda separated(TS) list building of response or building of the response xml . After careful inspection it was found that the error was occurring during the response ts build of interest subsystem.

Based on the error found in debug we checked the code where the error was occurring . We found there was no clob handling for response TS build of subsystem. In this case there we close to 100 interests who were accrued and while building TS for it is resulting the values to clob. Even though the main reply builder is capable of handling clob the subsystem reply builders were not. The fix for this issue would be complete revamp of the subsystem reply builder functions so that it can handle this situation.

4.1.5 Fix provided:

As mentioned in the above section the fix for the issue will have to be revamp of the subsystem reply builder functions so that it can handle clob values. But since the subsystem functions are common across different modules changing this would require changes in all modules using this subsystem functions so a different approach was finalized. Since there is clob handling in main reply builder , we decided to send empty temp TS list which were of size Varchar2(32767) and after each subsystem like settlements or interests were built add the TS list to the main TS list and empty the temp TS list . These processes were done before and after calling each subsystem reply build.

The files fixed for this issue are utpks_utdtxn01_main.spc and utpks_utdtxn01_main.sql. Once this fix was given the same issue was retested and it was closed successfully.

4.2 FLEXCUBE Host (Maintenance)

4.2.1 Issue Raised

Querying a record in "Fund Rule Maintenance" screen throws "Unhandled Exception"

4.2.2 Inference

Looking at this issue the first inference was there is an issue in either request reaching the backend or the response getting build because of which unhandled exception was coming.

4.2.3 Debugging:

There are different debugs which would be of use to this kind of issue.

Request and Response XML's from Debug Window in FLEXCUBE application

Click on Options in menu of FLEXCUBE application and select DEBUG from the sub menu. A new window opens now query for the contract details which has been mentioned in the error. In the debug window you will be able to see two buttons Request Xml and Response Xml on click of this buttons it will open the Request XML which has been send to the backend and Response XML which has come from backend.

Check the request xml whether the request is built properly and all the information is correctly send. Similarly check if the response xml is built properly. Once you make sure the request and response are send properly move to the next step. In this case the request xml was getting build without any values. So it was found the issue is in the request xml getting build. For further debugging we enabled script debugging of the browser and started debugging line by line the JS file to find out the reason in a script editor. This led us to the place of error.

4.2.4 Bug

The issue in this case was the element name in the sys js file and ui xml weren't the same. Because of which when the request xml is getting build the field values were missed out. It was also noted a view was being used as a data source for a Subscreen holding limit for investor category maintenance screen which is wrong. A table has to be used.

4.2.5 Fix Provided

Since the data source used was wrong it was decided to correct the screen completely. The sub screen was designed based on the FUNDUHCATREGCAPMAINTTBL table which was created newly and the screen was desgined. The new packages were generated and checked in. Once the new units were deployed the issue was retested and closed.

4.3 FLEXCUBE Gateway

4.3.1 Issue Raised

OT Knock In Knock Out IOPK is failing. Its giving an error cannot process request.

4.3.2 Inference:

Looking at this issue the first inference was there is some maintenance which is missing for the external system , service & operation and user combination.

4.3.3 Debugging:

There are different debugs which would be of use to this kind of issue.

FLEXCUBE Gateway Debug

This debug did not contain information of any error in this case. This debug had the request xml which was sent and it had been sent to the FLEXCUBE and a response has come back with the error. This case needs further investigation.

FLEXCUBE Messaging Schema Backend Debug

In the back end debug there was an oracle exception which was caught. *No Data Found*. So there is some maintenance which is missing because of which the processing is failed. On further investigation it was found that there was no maintenance in GWTM_OPERATIONS_MASTER for the service and operation as mentioned in the request. So it was checked if the service operation which is coming in the request is correct once it was verified we decided to provide an insert script for the service and operation.

4.3.4 Bug:

The issue was there was a maintenance missing for the service and operation which was trying to be invoked. Since the table is a factory shipped table the fix for this issue should be releasing an insert script for the table for that service and operation

4.3.5 Fix provided:

An insert script was released for the service and operation and it was retested and knock in knock out was happening successfully. But there was one error which was noticed that is the knock in rate which was given in the request was not being considered. So the issue was reopened. Now since this is an functional issue this issue needs further analysis and it was decided to first check FLEXCUBE business schema back end debug.

FLEXCUBE Business Schema Backend Debug

Since this is not a technical error there will not be any error or exception that would have been caught. So we started analyzing the debug from starting. It was noted that the knock in rate was

coming in the request xml. It was even send to the place where the request TS was converted to table type. But post that the value was missing. When we checked the code where TS is getting converted to table type we found the bug.

4.3.6 Bug:

The issue was the tag name which was coming in the request was different from the name in TS to table type conversion. Because of which the rate was not getting assigned and the default rate was taken. A fix for this would be either changing the name in request XSD or the code.

4.3.7 Fix provided:

Since the package is RAD tool generated we shouldn't be making change directly in the code. So the fix was to correct the XSD element name in RAD tool and regenerate the XSD and back end files and release them. Once this was corrected and files were released the issue was retested and closed.

4.4 EOD in FLEXCUBE

4.4.1 Issue Raised

During dividend freeze holding process in EOD, system is not considering the transactions allotted on the same day as of freeze holding date though the setup is consider all transactions allotted as of freeze holding date.

4.4.2 Inference

Looking at this issue the first inference was there seems to be some problem in which calculation was not done properly. The issue could be maintenance issue in Fund dividend maintenance itself or in the code.

4.4.3 Debug

The main debug which would be of interest in this case is FLEXCUBE back end debug. In case of EOD front end debugs would be of a very little use.

FLEXCUBE Backend Debug

There will be an entry in EODTBL which will have an EOD Sequence number. There will be a debug file with the name as 'SYSADMIN<DATETIMESTAMP>' for each EOD sequence number. That would be the main file of interest during debugging any issue in EOD. There are two types of issues anyone can face during EOD a) A batch getting aborted b) Batches are successful but there is some behavior which was expected of the EOD but it didn't happen. In the first case it will be easy to trace the error but second case is a bit more difficult.

This issue here is the second one. The issue is one contract which was supposed to be exercised as part of EOD on maturity date because the contract was in the money but it had got expired. The way to go about debugging in a case like this is search for the Fund ID in the debug and go from one occurrence to another. During careful investigation of debug file it was analyzed that the transactions were not picked for freeze holding because of the code issue where date tomorrow was used instead of date today. So the code corresponding to it was checked and bug was found out.

4.4.4 Bug

During the analysis of the issue in the file where transactions to be considered for freeze holdings are getting selected was checked we found the error. The cursor which is selecting the date today was selecting date tomorrow from sysparamtbl. There was an error in the way which the dates were getting passed for the cursor because of which the transactions were not picked. The code for that was corrected in pkgdividendfunctions.sql.

4.4.5 Fix Provided

The code of the cursor which was selecting the system date, was corrected by sending the correct dates. This was corrected. Once the fix was provided a new set of transactions were input and retested during next EOD and closed.

5 Glossary

TS list : Tilda Separated list

XML : Extended Markup Language

XSD : Extended Schema Document

EOD : End Of Day

Messaging Schema : This is the schema which is used for Gateway operations where the message is validated for authenticity

Business Schema : This is the schema which contains the business logic of FLEXCUBE

JS : Java Script

IDE : Integrated Development Environment

6 List of Examples

FLEXCUBE Host Online
FLEXCUBE Host Maintenance
FLEXCUBE Gateway
EOD in FLEXCUBE

7 List of Tables

Table 1 : Debugs in FLEXCUBE
Table 2 : Debug information

8 List of Figures

Figure 1 : Debug Window Menu
Figure 2 : Debug Window Main
Figure 3: Setting up Break Point
Figure 4 : Using Step into option
Figure 5 : Monitoring Values using Watch window



FCIS-FD06-02-01-Support By Example
April 2012
12.0

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/financial_services/

Copyright © 2012 Oracle Financial Services Software Limited. All rights reserved.

No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic recording or otherwise, translated in any language or computer language, without the prior written permission of Oracle Financial Services Software Limited.

Due care has been taken to make this document FCIS-FD06-02-01-Support By Example and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this FCIS-FD06-02-01-Support By Example and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.