Almost Everything about Transaction Launcher – Part I

Applies to:
SAP CRM WebClient UI. For more information, visit the Customer Relationship Management homepage.

Summary
While there are many documents that explain how to take advantage of Transaction Launcher functionality in CRM, I personally have not come across one which explains how to troubleshoot it. This article is a humble attempt. It includes configuration steps, Http Watch traces and debugging. Due to the sheer size of this document, I have only covered three most commonly used scenarios. I will be writing Part II soon, so stay tuned.

The three different scenarios used are:
1. Using Transaction Launcher to launch http://www.google.com in an external window
2. Using Transaction Launcher to launch MiddleWare transaction SMW01 within the same window on the same system
3. And finally launch ECC transaction VA01 in the same window

Author: Hasan Zubairi
Company: SAP
Created on: 12 June, 2011

Author Bio
I have been with SAP for a little over 5 years working as a Senior Support Consultant I in Primary Support, AGS Business Suite and Technology. I was working with SAP Retail POS and moved to CRM in 2009. I mainly focus on FRW, IC and Sales components.
Table of Contents

Scenario I - Using Transaction Launcher to launch http://www.google.com in an external window ........................................3
  Define URL........................................................................................................................................................................3
  Configure Transaction Launcher........................................................................................................................................3
  Define Logical Link for the URL (Navigation Bar Customizing) ..........................................................................................6
  Test in WebUI ........................................................................................................................................................................10
  HTTP Watch.........................................................................................................................................................................12
  Debugging...........................................................................................................................................................................13

Scenario II - Using Transaction Launcher to launch MiddleWare transaction SMW01 within the same window on the same system ..............................................................17
  Create BOR Object Type.....................................................................................................................................................17
  Test the newly created BOR Object..................................................................................................................................19
  Check Logical Systems and ITS URL................................................................................................................................22
  Configure Transaction Launcher........................................................................................................................................23
  Define Logical Link for the URL (Navigation Bar Customizing) .........................................................................................30
  Test in WebUI ........................................................................................................................................................................34
  HTTP Watch.........................................................................................................................................................................36
  Check Configuration for New Window................................................................................................................................44
  Test # 2 in WebUI ..................................................................................................................................................................46
  Test in SAPGUI......................................................................................................................................................................49
  Debugging...........................................................................................................................................................................51
  Test using .htm File............................................................................................................................................................62
  Edit Configuration..............................................................................................................................................................64
  WebUI Final Test Result......................................................................................................................................................68

Scenario III - Launch ECC transaction VA01 in the same window.........................................................................................68
  Find the RFC-Destination (ECC System – Does not apply to multiple backend scenarios) ..............................................68
  Connect to ECC ....................................................................................................................................................................70
  Create BOR Object Type in ECC.......................................................................................................................................72
  Test the newly created BOR Object..................................................................................................................................74
  Check Logical Systems and ITS URL....................................................................................................................................78
  Configure Transaction Launcher........................................................................................................................................80
  Define Logical Link for the URL (Navigation Bar Customizing) .........................................................................................87
  Test in WebClient UI ............................................................................................................................................................91
  HTTP Watch.........................................................................................................................................................................93
  Check Configuration..........................................................................................................................................................102
  Debugging........................................................................................................................................................................105
  Debugging in ECC ..............................................................................................................................................................111
  Back to Debugging in CRM................................................................................................................................................114
  Changes in Configuration....................................................................................................................................................121
  Test in WebClient UI............................................................................................................................................................122
  Related Content.................................................................................................................................................................131
  Copyright............................................................................................................................................................................132
Scenario 1 - Using Transaction Launcher to launch http://www.google.com in an external window

In this scenario we will configure Launch Transaction to launch URL http://www.google.com in a separate window. We will then examine it using HTTP Watch and then finally debug.

Define URL

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework Definition -> Technical Role Definition -> Transaction Launcher -> Define URLs and Parameters

Configure Transaction Launcher

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework Definition -> Technical Role Definition -> Transaction Launcher -> Configure Transaction Launcher
Almost Everything about Transaction Launcher – Part I

1. Enter anything as Transaction Launcher ID (Z_HZ_GOOGLE in our example)

2. Provide a meaningful description

3. Choose a class name, starting with X, Y or Z (it will be created automatically)

4. Activate checkbox for “Stateful” and deactivate “Raise Veto”
5. Transaction Type must be B (URL transaction)
6. URL ID the URL ID we have defined in Define URL section
7. Skip the rest of the parameters and then Complete.

8. You can check the Class generated by the Wizard in transaction SE24
Define Logical Link for the URL (Navigation Bar Customizing)

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework -> Technical Role Definition -> Define Navigation Bar Profile -> Define Logical links

You must create a logical link of type C (Launch Transaction) linked to our transaction launcher ID

1. Click on New entries:
Logical link ID: Z_HZ_LL_GO0
Type: C Launch transaction
Target ID: EXECLTX
Parameter: Z_HZ_GOOGLE
Parameter class: CL_CRM_UI_LTX_NAVBAR_PARAM
Title: Hasan's Google LT
Description: Hasan's Google Launch Tran

2. Create a direct link group
3. Assign Logical Link to Direct Link Group

4. Assign Direct Link Group to our Navigation Bar Profile
5. Ensure that the Direct Link Group is visible at Business Role level
7. Select the Business Role and Double click Adjust Direct Link Groups
8. Ensure that the new Group ID has Visible checked

9. Click on Adjust Direct Link and ensure that Visible is checked for LogLink ID

Test in WebUI

Now let’s login to WebUI and test the newly created Launch Transaction.
1. Login to WebUI using the customized Business Role

2. You can see the newly created Launch Transaction “Hasan’s Google LT” published under Direct Link Group “Hasan’s DLG for LT”
HTTP Watch

Now let's test it using HTTP Watch (since we choose Stateful, the Google link will open in a separate browser window)

If you remember from the configuration, we defined the Launch Transaction as GET
Almost Everything about Transaction Launcher – Part I

On the right pane, you see that the URL http://www.google.com was redirected to http://www.google.ca.

Debugging

Now let's debug:

1. Execute transaction se24 - *abox*impl - F4
2. Click on the Transaction Launcher (CL_UICMP_LT_ABOXEXECUTION_IMPL)
1. Click Display
2. Double click IP_INBOUNDPLUG

5. Set a breakpoint where you get the ME handler
6. Now in WebUI execute the Launch Transaction
7. Once the breakpoint is hit use F6 key to step through the code until you reach me->handler. Execute past it and then double click on it to get Implementation Class of the Launch Transaction
8. Double click on ME->HANDLER in Variables
9. Here you’ll see the Launch Transaction Class = Z_HZ_GOOGLE_CL
   GV_ATYPE = B
   GV_URL_ID = Z_HZ_GOOGLE

10. In transaction SE24 se24 put the class name (= Z_HZ_GOOGLE_CL ) and hit display
11. Double click on Method BUILD_OTHER_URL_SIMPLE (since we are using Non-BSP URL) and put a breakpoint on gs_url-url_part1
12. Double click on the Variables to get their values
Scenario II - Using Transaction Launcher to launch MiddleWare transaction SMW01 within the same window on the same system

In this section we will use Transaction Launcher to execute MW transaction SMW01 within the same window. To accomplish that we have to pass transaction SMW01 to BOR object ZHZTSTC (copy of TSTC). There is a prepared BOR object in the delivery: TSTC_UIF which will help you skip some of the following steps, but we are not using it in this scenario. First we will make a copy of object TSTC, and then test it. Secondly we will configure Transaction Launcher, then test, debug, use HTTP Watch and finally fix the issues.

Create BOR Object Type

1. Execute transaction SE80
2. Click on Workbench -> Edit Object -> click on tab Business Engineering -> choose Business Object Type -> type TSTC -> click on the copy icon
3. You would have to provide the new Object Type, Program & Object Name (starting with X, Y or Z)
4. Click on copy icon

5. You will be prompted for Package.
6. For this exercise we will use Package “$TMP”
7. Click on save icon
Almost Everything about Transaction Launcher – Part I

Test the newly created BOR Object

To test the newly created BOR Object and its EXECUTE method:

1. Execute transaction SE80
2. Click on Repository Information System
3. Expand Business Engineering
4. Expand Business Object Browser
5. Double click on Business Object Types
6. Type ZHZTSTC (the newly created Object) in Object Type field
7. Click on execute icon
8. Double click on the object
9. Expand Methods
10. Click on the Edit (Pencil) icon
11. Double click on Method ZHZ_Transaction.Execute

12. Check Synchronous
13. Click on Continue icon

14. Now you have to change the Release Status of the newly created Object

15. Click on Edit -> Change Release Status -> Object Type -> To Implemented
16. Click on Yes when prompted to save Object Type

Check Logical Systems and ITS URL

Since we are planning to execute the Launch Transaction in our own system, we have to verify that the Logical System and the corresponding ITS URL is defined.

1. Execute transaction CRMS_IC_CROSS_SYS

2. Here you can see that an entry is already maintained for MappedLogSys OWNLOGSYS
3. Logical System is CRECLNT001
4. ITS Client is 001
5. And ITS URL is
   http://iwdfvm3088.wdf.sap.corp:8000/sap/bc/gui/sap/its/webgui?"transaction=IC_LTX"&"okcode=1CEEXECUTE
   Hint:
   Okcode is not needed any longer.
6. Remember the “**transaction=IC_LTX**” part in the above URL. We will be needing that during our debugging

**Configure Transaction Launcher**
Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework Definition -> Technical Role Definition -> Transaction Launcher -> Configure Transaction Launcher
1. Enter anything as Transaction Launcher ID (Z_HZ_SMW01 in our example)

2. Provide a meaningful description

3. Choose a class name, starting with X, Y or Z (it will be created automatically)

4. Activate checkbox for "Stateful" and deactivate "Raise Veto"
5. Since we are planning to execute SMW01 in our own system, we select OWNLOGSYS = CRECLNT001

6. BOR Object Type will be the one that we created earlier, ZHZTSTC
7. You can type ZHZ* and click on F4 help
8. Click on Information System

9. Check Implemented and uncheck Released
10. Click on continue
11. Select the Object you created from the results

12. For Method name, click on the F4 Help and select Execute
13. So it should look like

14. Click on continue

15. Now we will define what value will be passed to the parameter

16. For Parameter select “ICWEBCLIENTBORKEYPARAMETER Object Key”

17. For Value type smw01
18. Click on Continue twice and then Complete

19. Create a Customizing Request, when prompted

20. Check the Class generated by the Wizard in transaction SE24
Define Logical Link for the URL (Navigation Bar Customizing)

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework -> Technical Role Definition -> Define Navigation Bar Profile -> Define Logical links

You must create a logical link of type C (Launch Transaction) linked to our transaction launcher ID

1. Click on New entries:
Logical link ID: ZHZ1LLSMW0
Type: C Launch transaction
Target ID: EXECLTX
Parameter: Z_HZ_SMW01
Parameter class: CL_CRM_UI_LTX_NAVBAR_PARAM
Title: Hasan's SMW01 LT
Description: Hasan's SMW01 Launch Tran

2. Save it. You might be prompted for Customizing Request

3. Find the previously created Direct Link Group and select it
4. Double click on Assign Links and click on New Entries
5. Ensure that the Direct Link Group is visible at Business Role level
7. Select the Business Role and double click Adjust Direct Link Groups
8. Double click on Adjust Direct Links

Test in WebUI

Now let’s login to WebUI and test the newly created Launch Transaction.

1. Login to WebUI using the customized Business Role
2. You can see the newly created Launch Transaction “Hasan’s SMW01 LT” published under Direct Link Group “Hasan’s DLG for LT”

3. Once you click on it, you will notice that it is trying to Launch Transaction in a new window, instead of the same one.
4. And the window closes automatically

HTTP Watch

Now let's test it using HTTP Watch
1. Let’s save the HTTP Watch file and open it up so that we can investigate it in detail
2. Click on Find and type genericpostexecution.htm
3. Uncheck everything except for URLs

4. Click on Find Next and it will find the URL with string genericpostexecution.htm
5. Click on the Content tab
6. Copy the contents
7. Paste them in Notepad or another editor
8. And save it as an htm file

9. Let’s analyze the contents
10. Remember the ITS URL


You will see that in method POST
11. The BOR Object we created ZHZTSTC, its EXECUTE method in Logical System CRECLNT001

12. The parameter smw01 passed to ZHZTSTC
13. Let’s launch the newly created htm file (smw01.htm in our example)

14. Click on Yes, when prompted for ActiveX
15. Nothing but an empty window

16. By testing the contents of the htm file, we have bypassed CRM, thus proving that the issue is related to either the BOR Object itself or the parameter passed to it, smw01 in this instance.
17. At this point we have identified two issues. First, the Launch Transaction is opening in a new window. And second, we are getting a blank screen.

**Check Configuration for New Window**

If the Launch Transaction has a Stateful property checked, then it will open in a new window


2. Search for Launch Transaction Z_HZ_SMW01
3. Select it and click on Details

4. Uncheck Stateful and Save
5. Click on Continue when prompted for Customizing Request
Test # 2 in WebUI

Now let’s login to WebUI and test the Launch Transaction again.

1. Login to WebUI using the customized Business Role

2. You can see the newly created Launch Transaction “Hasan’s SMW01 LT” published under Direct Link Group “Hasan’s DLG for LT”
3. Once you click on it, you will notice that it is now being launched within the same window.
4. It tries to execute it, but is unsuccessful.
Almost Everything about Transaction Launcher – Part I
Test in SAPGUI

Let’s test the BOR Object and the parameter in SAPGUI

1. In SAPGUI execute transaction sw01 and type in ZHZTSTC for Object/Interface Type

2. Click on Test
3. Type smw01 for Transaction Code

4. Execute method EXECUTE

5. It works
Debugging

Now let's debug:

1. Execute transaction se24 -> *abox*impl -> F4
2. Click on the Transaction Launcher (CL_UICMP_LT_ABOXEXECUTION_IMPL)
3. Click Display
4. Double click IP_INBOUNDPLUG
5. Set a breakpoint where you get the ME handler

6. Now in WebUI execute the Launch Transaction

7. Once the breakpoint is hit use F6 key to step through the code until you reach me->handler. Execute past it and then double click on it to get Implementation Class of Launch Transaction
8. Double click on ME->HANDLER in Variables
9. Here you’ll see the Launch Transaction Class = Z_HZ_SMW01_CL
   GV_ATYPE = A
   GV_BORTYPE = ZHZTSTC
   GV_MAPPEDLOGSYS = OWNLOGSYS
   GV_METHOD = EXECUTE
10. In transaction SE24 se24 put the class name (Z_HZ_SMW01_CL) and hit display
11. Double click on method PREPARE_DATA_FLOW

12. Here you will see ICWEBCLIENTBORKEYPARAMETER = 'smw01'
13. So the correct parameter is passed
14. Remember in Check Logical Systems and ITS URL section, point 6, I told you to remember the “transaction=IC_LTX” part in ITS URL
15. Execute transaction SE80 and access Function Group IC_LTX
16. Double click on Subroutine USER_COMMAND_0100
17. Set a breakpoint where you execute method of BOR Object

18. Hit F8 in the Debugger to hit the new Breakpoint
19. Hit F6 until you pass the line where gt_swcont gets populated

20. Double click on GT_SWCONT

21. Correct value is received
22. Press F8 to exit Debugger

23. If you’re quick you will see the error at the bottom of the screen stating “You do not have authorization for transaction smw01”. But we know that we have the proper authorization

24. Execute transaction SE24 and display class Z_HZ_SMW01_CL
25. Put a Breakpoint in method PREPARE_DATA_FLOW

26. Execute the Launch Transaction from WebUI and hit F8 when the Debugger hits your first Breakpoint to hit the next Breakpoint

27. Double click on IWEBCLIENTBORKEYPARAMETER in the Source Code window
28. Hit F5 till you pass ICBWEBCLIENTBORKEYPARAMETER = 'smw01'

29. Double click on ICBWEBCLIENTBORKEYPARAMETER in the Variable window

30. Click on the pencil icon to edit
31. Change smw01 to SMW01 (lower to UPPERCASE) and hit enter

32. Hit F8 to exit debugger (you might have to do it twice since you have another breakpoint set in Subroutine USER_COMMAND_0100 of Function Group IC_LTX

33. Transaction is launched without any issues.
34. You will get error stating “You do not have authorization for transaction ...” in Launch Transaction if the parameter(s) passed are in lower case

Test using .htm File

To verify our findings, we will do another test by changing SWCONT-VALUE in the .htm file we created earlier

1. Open the .htm file in edit mode

2. Change smw01 in SWCONT-VALUE to SMW01 and save it
3. Execute the file

4. Click on Yes when prompted to allow ActiveX
5. And the transaction is launched without any problems

Edit Configuration

Now let's make the configuration changes
1. Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework Definition -> Technical Role Definition -> Transaction Launcher -> Configure Transaction Launcher

2. Click Continue

3. Launch Trans. ID = Z_HZ_SMW01
4. Click Continue

5. Leave everything the way it is on this screen

6. Click Continue

7. Leave everything the way it is on this screen

8. Click Continue
9. Parameter = ICWEBCLIENTBORKEYPARAMETER Object Key
10. Value SMW01 (all upper case)

11. Click on Continue twice and then Complete
12. Click on Continue when prompted for Customizing Request
WebUI Final Test Result

Execute the Launch Transaction “Hasan’s SMW01 LT” from Navigation Bar again. It should work without any issues.

![WebUI Final Test Result Image]

Scenario III - Launch ECC transaction VA01 in the same window

In this section we will use Transaction Launcher to execute ECC transaction VA01 within the same window. We will use an existing BOR Object in ECC. First we will TEST the BOR Object in ECC, and then we will configure Transaction Launcher, test, debug, use HTTP Watch and finally fix issues if there are any.

Find the RFC-Destination (ECC System – Does not apply to multiple backend scenarios)

1. In CRM execute transaction SMOEAC
2. From the Object Type dropdown select Site
3. Leave the Object Name blank and hit the enter key
4. Expand R/3
5. Double click on OLTP and the right window pane will populate

6. Click on Site Attributes and in the popup you will see the RFC-Destination
7. The ECC system that we will be connecting to is ALACLNT800
Connect to ECC

1. In CRM execute transaction SM59

2. Expand ABAP Connections
3. Double click on ALACLNT800 (the RFC-Destination we discovered in the previous step)
4. Click on Remote Logon
Create BOR Object Type in ECC

1. In the previous example we created our own copy of BOR Object TSTC in CRM. If WEBCUIF is installed on the ECC system, you can use an already prepared BOR Object TSTC_UIF (both in CRM and ECC). But since WEBCUIF is not installed on this particular ECC system, we would be creating a copy of TSTC.

2. Execute transaction SE80

3. Click on Workbench -> Edit Object -> click on tab Business Engineering -> choose Business Object Type -> type TSTC -> click on the copy icon
4. You would have to provide the new Object Type, Program & Object Name (starting with X, Y or Z)
5. Click on copy icon
You will be prompted for Package.

For this exercise we will use Package “$TMP”

Click on save icon

Test the newly created BOR Object

To test the newly created BOR Object and it's EXECUTE method:

1. Execute transaction SE80
2. Click on Repository Information System
3. Expand Business Engineering
4. Expand Business Object Browser
5. Double click on Business Object Types
6. Type ZHZECCTSTC (the newly created Object) in Object Type field
7. Click on execute icon
Double click on the Object

Expand Methods

Click on the Edit (Pencil) icon

Double click on Method ZHZ_ECC_Transaction.Execute
12. Check Synchronous

13. Click on Continue icon
14. Now you have to change the Release Status of the newly created Object.
15. Click on Edit -> Change Release Status -> Object Type -> To Implemented.

16. Click on Yes when prompted to save Object Type.
Check Logical Systems and ITS URL

Since we are planning to execute the Launch Transaction in our own system, we have to verify that the Logical System and the corresponding ITS URL are (?) defined.

1. Execute transaction CRMS_IC_CROSS_SYS
2. Here we see 2 Mapped Logical Systems defined for the same Logical System
3. We are interested in Mapped LogSys ERP_ALA800 as ITS URL:
   is using IC_LTXE where as ITS URL for Mapped LogSys ERP is using crm_cic_rabox which is obsolete (see Related Context section at the end of this article)
4. Lets verify to which RFC-Destination this Logical Systems is pointing to
5. Execute transaction BD97
6. Here you see Logical System T90CLNT090
7. Expand T90CLNT090
8. You can see that T90CLNT090 is connected to ALACLNT800 for both Standard RFC destination for BAPI calls & Standard RFC destination for dialog calls
Configure Transaction Launcher

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework Definition -> Technical Role Definition -> Transaction Launcher -> Configure Transaction Launcher

1. Enter anything as Transaction Launcher ID (Z_HZ_ECC_VA01 in our example)
2. Component Set = <empty> as we are not putting data from BOL into the LTXE parameters
3. Provide a meaningful description
4. Choose a class name, starting with X, Y or Z (it will be created automatically as local object)
5. Uncheck “Stateful” and "Raise Veto"

6. Since we are planning to execute VA01 in our own system, we select ERP_ALA800 = T90CLNT090
HINT:
The display of keys in dropdowns must be activated in the SAP GUI configuration. Otherwise only the second column will be displayed. In SAPLOGON, click upper right corner -> options (I use SAPLOGON 710)

7. You can type ZHZ* and click on F4 help
8. Click on Information System

9. Check Implemented and uncheck Released
10. Click on continue
11. Select the Object you created
12. For Method Name, click on the F4 Help and select Execute
13. It should look like ...

14. Click on continue

15. Now we will define what value will be passed to the parameter

16. For Parameter select “ICWEBCLIENTBORKEYPARAMETER Object Key”

17. For Value Type VA01 (upper case)
18. Click on Continue twice and then Complete

19. Create a Customizing Request, when prompted

20. Check the Class generated by the Wizard in transaction SE24
Define Logical Link for the URL (Navigation Bar Customizing)

Transaction SPRO -> SAP Reference IMG -> Customer Relationship Management -> UI Framework -> Technical Role Definition -> Define Navigation Bar Profile -> Define Logical links

You must create a logical link of type C (Launch Transaction) linked to our transaction launcher ID

1. Click on New entries:

   Logical link ID: ZHZLLVA01
   Type: C Launch transaction
   Target ID: EXECLTX
   Parameter: Z_HZ_ECC_VA01
   Parameter class: CL_CRM_UI_LTX_NAVBAR_PARAM
   Title: Hasan's VA01 LT
   Description: Hasan's VA01 Launch Tran
2. You might be prompted for Customizing Request

   ![Prompt for Customizing Request]

   Request: CREK998573
   Short Description: Hasan TL Tests

3. Find the originally created Direct Link Group and select it
4. Double click on Assign Links and click on New Entries

5. LogLink ID = ZHZLLVA01
   Position = 30

6. Hit enter and SAVE
7. Ensure that the Direct Link Group is visible at Business Role level
9. Select the Business Role and double click Adjust Direct Link Groups
10. Double click on Adjust Direct Links and set the visibility indicator

![Image of the Change View "Adjust Direct Links" window]

11. Click on Save

12. You might be prompted for Customizing Request

**Test in WebClient UI**

Now let's login to WebClient UI and test the newly created Launch Transaction

1. Login to WebClient UI using the customized Business Role
2. You can see the newly created Launch Transaction “Hasan’s VA01 LT” published under Direct Link Group “Hasan’s DLG for LT”

3. Upon launching the newly configured transaction, you encounter an error
4. Take a closer look at the error:

URL http://pwdf6373.wdf.sap.corp:8045/sap/bc/gui/sap/its/crm_cic_rabox call was terminated because the corresponding service is not available.

Why is the Launch Transaction launching the ITS URL with crm_cic_rabox? And the message service is not available?

HTTP Watch
Now let’s test it again using HTTP Watch
1. Let’s save the HTTP Watch file and open it up so that we can investigate it in detail

2. Click on Find and type `genericpostexecution.htm`

3. Uncheck everything except for URLs
4. Click on Find Next and it will find the URL with string `genericpostexecution.htm`

5. Click on the Content tab

6. Copy the contents
7. Paste them in Notepad or another editor
8. And save it as an htm file

9. Let’s analyze the contents
10. Remember the ITS URL
But instead we see URL


in the attribute action of form tag

11. The PROCLID starts with M (method call) followed by the BOR Object ZHZECCTSTC method EXECUTE in Logical System T90CLNT090
12. The parameter SWCONT-VALUE = T90CLNT090ZHZEECTSTCVA01 (LOGSYS = T90CLNT090; BOR Object = ZHZECTSTC; Parameter = VA01)

13. At this time it is safe to assume that incorrect ITS URL is called by the Launch Transaction

14. Before we start debugging and making configuration changes, let’s verify if the correct URL works
15. Change the URL to:

```
```

in the attribute action of form tag.

16. Let’s launch the newly created htm file.
17. Click on Yes, when prompted for ActiveX

HINT:
You can avoid this in the Internet Options – Advanced – Security – Allow Active Content to run in files on My Computer. But be sure you remove this indicator as soon as tests are done.

18. You might be prompted for login credentials
19. Rendering might be bad

20. But you can type transaction VA01 in transaction field and hit enter
21. Now it is confirmed that the problem is with the URL

**Check Configuration**

If the Launch Transaction has a Stateful property checked, then it will open in a new window

2. Search for Launch Transaction Z_HZ_ECC_VA01
3. Select it and click on Details
4. You can see in Definition 3 is ERP_ALA800 (Mapped Logical System)

5. To confirm it execute transaction CRMS_IC_CROSS_SYS
6. So why is it that the URL for Mapped Logical System ERP is called?

**Debugging**

Let's debug and find out.

1. Execute transaction se24 -> *abox*impl -> F4
2. Click on the Transaction Launcher (CL_UICMP_LT_ABOXEXECUTION_IMPL)
3. Click Display
4. Double click IP_INBOUNDPLUG
5. Set a breakpoint where you get the ME handler

6. Now in WebClient UI execute the Launch Transaction
7. Once the breakpoint is hit use F6 key to step through the code until you reach me->handler. Execute past it and then double click on it to get Implementation Class of Launch Transaction.

8. Double click on ME->HANDLER in Variables.
9. Here you’ll see the Launch Transaction Class = Z_HZ_ECC_VA01_C
   GV_ATYPE = A
   GV_BORTYPE = ZHZECCSTC
   GV_MAPPEDLOGSYS = ERP_ALA800
   GV_METHOD = EXECUTE

10. In transaction SE24 se24 put the class name (Z_HZ_ECC_VA01_CL) and hit display
11. Double click on method PREPARE_DATA_FLOW
12. Here you will see ICWEBCLIENTBORKEYPARAMETER = 'VA01'

13. So the correct parameter is passed but let’s verify that during debugging by putting a breakpoint in class Z_HZ_ECC_VA01_CL method IF_CRM_IC_ACTION_HANDLER~PREPARE_DATA_FLOW
14. Switch back to your debugger and hit the F8 key to hit the new breakpoint.
15. Double click on the ICWEBCOMMUNICATIONPARAMETER to see the value.
16. Since the ITS URL is


in the target ECC we use transaction SE80 to access Function Group IC_LTXE (for ERP 2005 and up - IC_LTXR for PI systems) and put a breakpoint in Subroutine USER_COMMAND_0100

Debugging in ECC

1. Logon to the target ECC system
2. Remember message service cannot be reached

3. That can be easily checked by executing transaction SICF in ECC
4. Type CRM_CIC_RABOX in Service Name and click on Execute
5. Here you will see that service CRM_CIC_RABOX is inactive (greyed out)
6. We can activate the service if we like, but the issue that we are trying to figure out is why the Launch Transaction is calling incorrect URL
7. Execute transaction SE80
8. From the dropdown select Function Group
9. Type IC_LTXE and hit enter
10. Expand Subroutines
11. Double click on USER_COMMAND_0100
12. And put a breakpoint where on call function ‘EWB_PROC_CREATE’
13. Switch back to your debugger and hit F8 to continue to this new breakpoint.
14. You will notice that the debugger window is closed and your breakpoint is never hit which makes sense since for some reason ITS is calling CRM_CIC_RABOX and not IC_LTXE.
Back to Debugging in CRM

1. Execute transaction se24 -> CL_CRM_IC_ACTION_BOR_UTIL and click on Display

2. Put a breakpoint in method CONSTRUCTOR where you get the me->handler
3. Hit the back button and place another breakpoint in method GET_ITS_URL

4. I am sure by this time either your WebClient UI or Debugger or both have timed out

5. Execute Launch Transaction Hasan’s VA01 LT again in WebClient UI
6. Hit F6 until you get past the line where you get logsys
7. Here you will see that:
   
   MAPPED_LOGSYS = ERP_ALA800
   LOGSYS = T90CLNT090
8. FOUND = ABAP_TRUE

9. Look at the highlighted code below

```abap
loop at cl_crm_ic_action_bor_util=>setup_tab assigning <fs> where logsys = logsys.
  if <fs>-its_url is not initial.
    found = abap_true.
  exit.
  else.
    found = abap_false.
  endif.
endloop.

found = abap_true based on logsys and not mapped_logsys
```

10. Let’s take it further and examine the case/endcase
case found.
  when abap_true.
    me->its_url = <fs>-its_url.
    me->its_client = <fs>-its_client.
    me->logsys = <fs>-logsys.
  when abap_false.
    raise exception type cx_crm_ui_ltx
      exporting
        textid = cx_crm_ui_ltx=>no_its_url
        mapped_logsys = mapped_logsys.
  when abap_undefined.
    raise exception type cx_crm_ui_ltx
      exporting
        textid = cx_crm_ui_ltx=>no_logsys_error
        mapped_logsys = mapped_logsys.
endcase.

11. Hit F6 until you pass the endcase
    ME->ITS_CLIENT = 800
    ME->LOGSYS = T90CLNT090
12. The whole CONSTRUCTOR is based on finding the correct URL on value of LOGSYS instead of MAPPED_LOGSYS, thus finding URL:
   instead of:
13. Hit F8 to hit the breakpoint in method GET_ITS_URL and then hit F6
14. You will notice that the ITS URL that is determined is incorrect already
   \[ RV\_RESULT = \text{http://pwdf6373.wdf.sap.corp:8045/sap/bc/gui/sap/its/crm\_cic\_rabox?sap-client=800} \]

15. Let’s take it a step further … hit F7 and you will reach class CL_CRM_UI_LTX_PAGE_LAYOUT_SRV
    method BUILD\_AND\_ADD\_EXECUTION\_LAYOUT
16. Hit F6 several times until you get the values for the following variables
   LV_ITS_CLIENT = 800
   LV_OKCODE = EXECUTE

17. Examine the code:

   IF lv_its_url CS 'WEBGUI'. " upper/lower case is ignored
   lv_okcode = 'ICEXECUTE'. " new ITS integration using WEBGUI
   ELSE.
   lv_okcode = 'EXECUTE'. " old ITS integration using *RABOX*
   ENDIF.

   LV_OKCODE = EXECUTE, meaning that old ITS integration is used.

18. I think that by now we have proved that the ITS URLs for the Launch Transactions are called based
    on LOGSYS and not MAPPED_LOGSYS. This is because the coding has been changed in the core
    application (don’t ask me when)

19. Hit F8 to end the debugging session

Changes in Configuration

Let’s remove the unwanted URL and retest

1. Execute transaction CRMS_IC_CROSS_SYS

2. Delete the ITS URL for MAPPED_LOGSYS = ERP
3. Click on Save and you will be prompted for Customizing Request.
4. Click on Continue

Test in WebClient UI

Logout of your current WebClient UI session and log back in again to avoid any issues due to caching
1. Execute Launch Transaction Hasan’s VA01 LT
2. Your first breakpoint in class \texttt{CL\_CRM\_IC\_ACTION\_BOR\_UTIL} method \texttt{CONSTRUCTOR} will be hit

3. Hit F6 until you get the values for the following:
\texttt{LOGSYS = T90CLNT090}
ME->ITS_CLIENT = 800
ME->LOGSYS = T90CLNT090
FOUND = X

4. As you can see that already it is choosing the correct ITS URL
5. Click on F8 to hit your next breakpoint in class CL_CRM_IC_ACTION_BOR_UTIL method GET_ITS_URL where you will see that:
6. Click on F8 to hit your next breakpoint.
7. This one will be in target ECC in Function Group SAPLIC_LTXE subroutine USER_COMMAND_0100

8. Hit F6 until you get past the line where variable LV_PROCLID gets populated
9. To find out what parameter is received by BOR Object ZECCTSTC method EXECUTE, execute transaction SE80 (in ECC)

10. From the dropdown select Program and type ZHZ_ECC_RSOWTSTC (remember step 5 of section Create BOR Object Type in ECC)
11. Right click on Object Name ZHZ_ECC_RSWTSTC select Display and then In Same Window

12. Click on Program
13. Unfortunately you can only set a Session breakpoint here and not an External

14. Switch back to the debugger screen

15. From the menu, click on Breakpoints -> Breakpoint at -> Breakpoint in Source Code
16. Fill in the following values
   Program = ZHZ_ECC_RSOWTSTC
   Include = <OBJECT>
   Row 19

17. Click Continue
18. Hit F8 so that the new breakpoint is hit
19. Hit F6 until you get the value for OBJECT-KEY-CODE
20. Hit F8 to exit debugger

21. And the Launch Transaction Hasan’s VA01 LT is successful
Related Content
Here are some helpful Notes that I usually reference:

1360904  SAP GUI for HTML: Application services do not start
914112  ITS service CRM_CIC_RABOX causes UI problems
1247621  LTX: No data loss popup of transaction launcher in portal
986257  CRM Web UI: Protocol switch in HTTPS scenarios
900814  CRM: Generating URLs in proxy scenarios
1385426  Use basis code for creating URLs
1360019  Check existence of function module remotely (URL creation)
1298695  Parameters not passed to launch transaction
1013376  FAQ Note: CRM IC Web Client Transaction Launcher
1388655  Transaction ICEXECUTE is unknown
1139385  Error message 00031 - Transaction ICEXECUTE is unknown
1362533  ITS based launch transactions do not close sessions
1135335  ITS based launch transactions do not close sessions
990216  FAQ: Display problems for SAPGUI transactions via ITS
1148829  Locally executed launch transactions ignore HTTPURLOC
1055324  POST parameters lost during redirect
985587  Launch transaction handler class fails with syntax errors
974794  'Hide Dialog' setting for a URL launch transaction
962265  Launch transaction wizard fails with syntax errors for BOR
913601  Generation error in PCUI based launch transactions
1248650  Back Button on Transaction Launched leads to a blank screen

For more information, visit the Customer Relationship Management homepage
Almost Everything about Transaction Launcher – Part I

Copyright

© Copyright 2011 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.