

Working with eCATT (Extended Computer Aided Test Tool)

(By: [KVR Prasad Babu](#)^[11], email: prasadbabu.koribilli@gmail.com^[13])
(Revisado por Cássia Lima e Mauricio Rolim)

Hello friends my name is **Prasad Babu**. I would like to share some of my experiences with eCATT, in which am currently working on.

Description:

In this article first I will introduce some of the basic concepts regarding eCATT^[14] and then we will see how to develop a test script to upload a test data file with a practical example.

So in precise in this article you will see what eCATT is, how we can use it; how we can load a test data from a file in 4.7X and it end with an example.

After reading this article you will have a clear idea about eCATT tool. And you will be able to write a test script in which test data can be loaded from a file.

This article is divided into following sections,

1. **What is eCATT?**
2. **How to load test data from a file in 4.7X with an example?**

What is eCATT?

eCATT stands for *extended Computer Aided Test Tool (eCATT)* which is built is testing tool to test SAP system. By using testing tool we can test the entire business process, and we can also use this tool with a third party ^[15]testing tool (I am not covering this topic). Execution of every test script ends with a log, which explains the results of the test script.

By using eCATT we can do following operations,

- Test transactions, reports, and scenarios
- Call BAPIs and function modules
- Test remote systems
- Check authorizations (user profiles)
- Test updates (database, applications, GUI)
- Test the effect of changes to customizing settings
- Check system messages

For more information go to: [eCATT::Extended Computer Aided Test Tool \(sdn.com\)](#)

Here I am discussing fundamentals about eCATT in detail. You can find very good documentation in sdn.com.

To develop a test script in eCATT we need to follow the following steps,

1. **Creating Test Scripts.**
2. **Creating Test Data Containers**^[16].
3. **Understanding System Data Containers**^[17].
4. **Executing Test Configurations.**

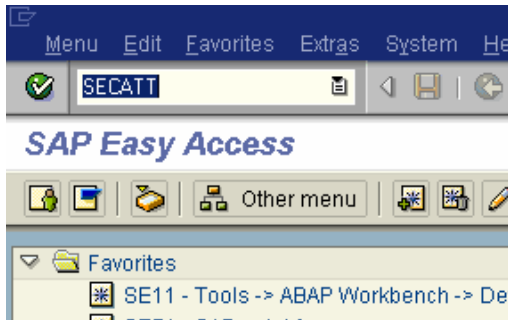
There is a very good web blog on eCATT in sdn.com which explains eCATT with necessary screen shots. To read document click here [Blog on eCATT in sdn.com](#)


Loading test data from a file in 4.7X with example:

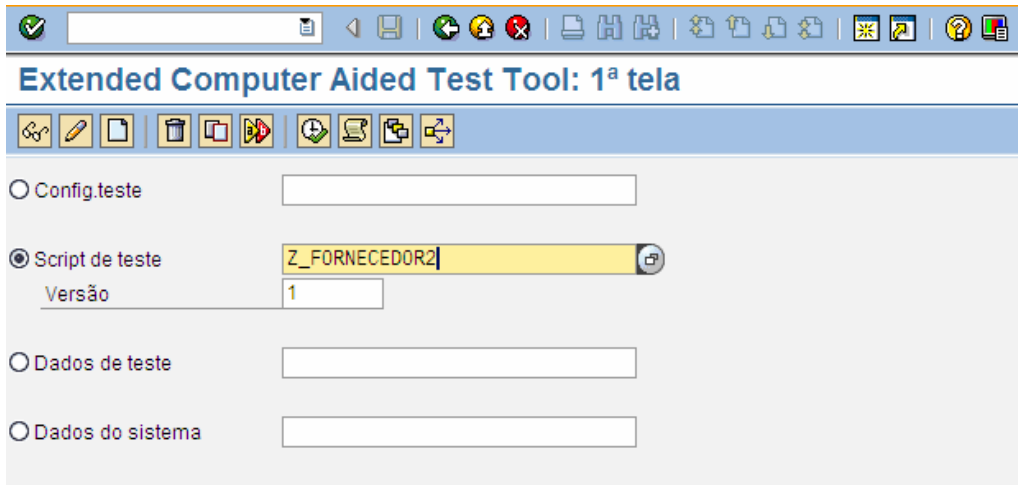
We can load test data from file using the `ABAP...ENDABAP` statements in eCATT. In this article I have given an example on transaction code `XK01` with sample code. To upload the test data file follow the steps given,

1. Record the transaction

Open eCATT tool. This can be done using 'SECATT' tcode.



Give the script name and version number example '`Z_FORNECEDOR2`'. Version number can be used to maintain different program constructs under same program name. And choose create  button.



Give description and component name as 'MM'.

Script de teste	Z_FORNECEDOR2	Versão	1	Sistema des
Título				Instância

Editor Atributos

Dados gerais Dados de criação versões Suplementos Restrições

Dados do cabeçalho

Título	Criar fornecedor		
Pacote			
Responsável	MYSAPTMG04	Tipo	B
Componente	MM		

Select editor tab panel. Then click on 'Modelo' button. Or go to Edit->Pattern or Press Ctrl+F6. This opens a 'insert statement' dialog box.

Selecionar grupo UI, comando TCD REC TCD (Record), transação a ser mapeada e o nome da interface a ser criado.

Inserir modelo

Grupo	UI Controle interface de usuário
Comando	TCD REC TCD (Record)
Transação	XK01
Interface:	XK01_1

✓ ✗ ⓘ

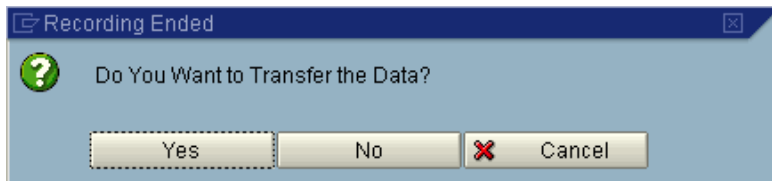
Then 'Create Vendor: Initial screen' will appear. Enter the necessary fields. And be careful while recording, if not recording may fail. Note that while recording no error messages should pop up. If they happen restart the recording process. Here in this example I have considered a simple recording process. You can do any complex recording. If you have any doubts do feel free to mail me. I will reply to you.

criar Fornecedor: 1ª tela

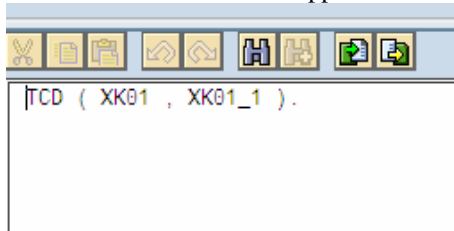
Fornecedor	CASSIAAAB	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Empresa	TMGA	TERMAG
Organização de compras	TMGA	Cargas dos Navios
Grupo de contas	ZNAV	

Modelo	
Fornecedor	
Empresa	
Organização de compras	

After completing the recording process recording ended dialog will appear. Choose 'Yes'.



Then a TCD statement will appear in the editor area.



With this we have finished recording
Now let us see the variable declaration, assignment and programming part.

After developing as number of scripts I found one simple method to develop these test scripts. If you feel comfortable with this method you can also use it.

First note down the all screen fields in which you are entering values during recording process. Then create local variables in eCATT with the same name as the screen field technical name. (This method makes assignment easier).

Example:

Create Vendor: Initial Screen

Vendor	CASSIAAAB
Company Code	TMGA
Purch. organization	TMGA
Account group	znav


In XK01 (vendor master) I have entered values for material, industry sector and material type. And their respective technical screen field values are,

'RF02K-LIFNR'

'RF02K-BUKRS'



'RF02K-EKORG'

'RF02K-KTOKK'

To find out technical value of the screen field select the field press F1, then clicks on technical information button. 

And now create the local variable as

V_LIFNR
V_BUKRS
V_EKORG
V_KTOKK

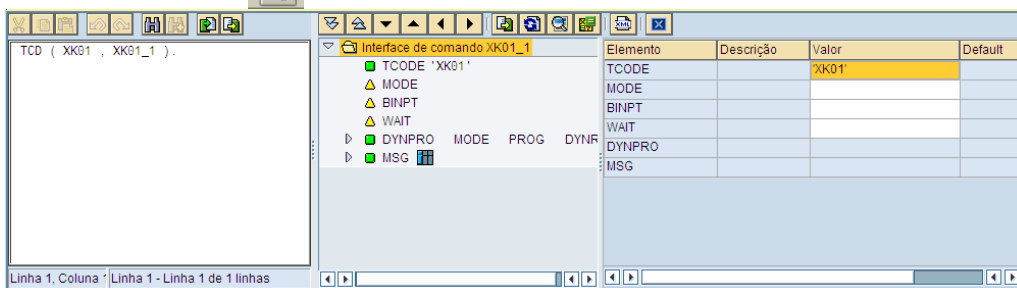
To create local variables first click  button, and then to create new variable click 

button. And enter the variable name [\[19\]](#) as (V_LIFNR), Visibility of parameter [\[110\]](#) as 'V' and finally parameter reference [\[111\]](#) (name of the actual parameter). After declaring all the parameter it will look like this,

Parameter	Descrição	Valor	IEA	Referência do parâmetro	Sistema destino	Ctg. ABAP	Compr	Cas...	Grupo
V_EKORG	Org Compras		V	EKORG		C	4		
V_KTOKK	Grupo Contas		V	KTOKK		C	4		
V_NAME1	Nome Fornecedor		V	NAME1		C	30		
V_SORTL	Termo pesquisa		V	SORTL		C	10		
V_LAND1	País		V	LAND1		C	3		
V_WAERS	Moeda Fornecedor		V	WAERS		C	5		
FILE_V	IBIP: data transfer parameters	<INITIAL>	V	IBIPPARMS					
COUNT			V			I	10		
INT_LOC		1	V			I	10		
V_KUNNR	Cliente		V	KUNNR		C	10		

After declaring the local variables we need to assign them to screen field values.

To do that again press  or double click on **XXK01_1** in **TCD (XXK01, XXK01_1)** statement.



The screenshot shows the SAP TCD editor interface. On the left, the command line contains the statement: `TCD (XXK01 , XXK01_1).`

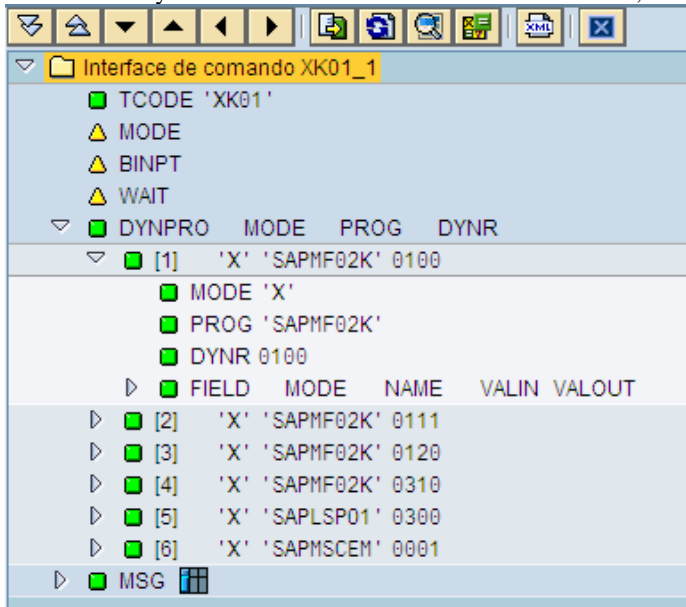
The central pane shows the 'Interface de comando XXK01_1' with a tree view of variables:

- TCODE 'XXK01'
- MODE
- BINPT
- WAIT
- DYNPRO MODE PROG DYNR
- MSG

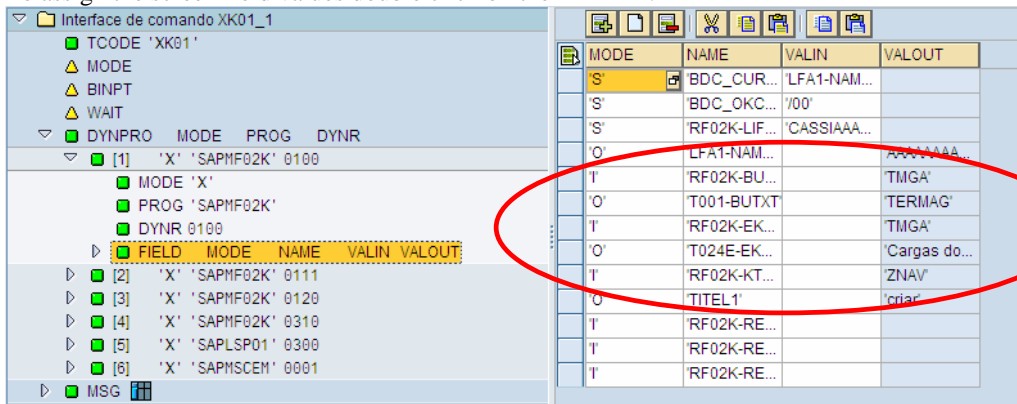
On the right, a table lists the elements and their values:

Elemento	Descrição	Valor	Default
TCODE		XXK01	
MODE			
BINPT			
WAIT			
DYNPRO			
MSG			

This will take you to the command interface which look like,

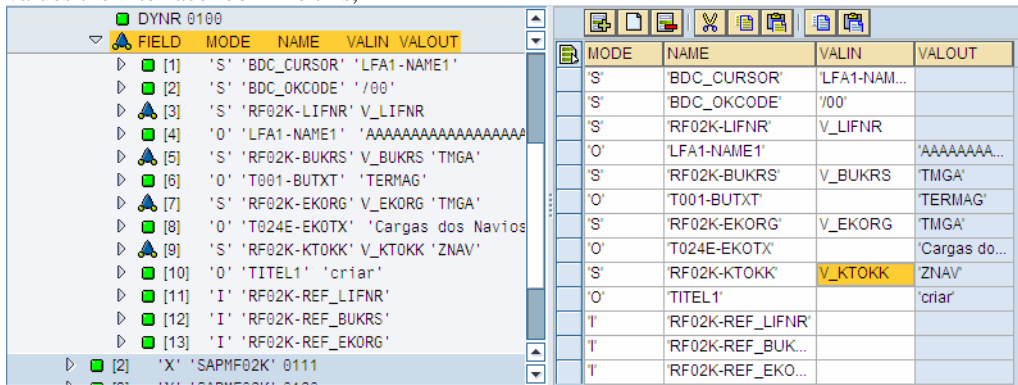


To assign the screen field values double click on the 'FIELD'.



Caso apareça na primeira coluna valores `T` com a coluna VALOUT preenchida, substituir por `S`.

Then replace the screen [VALIN^{\[112\]}](#) values with the local variable names. After changing the values the interface look like this,



Repeat the above step with all the screen field values. With this we have finished the process of declaring and assigning variables.

Now we will see how to program and run the script successfully.

To write the program we need to follow two steps. They are,

1. **Get the number (count) of records in file.**
2. **Loop through count number of times and load the data from the file pass to TCD through local variables of eCATT.**

Let us see how to handle the things,

By using **ABAP...ENDABAP** statements we can do that.

Before we start writing the program we need to declare some local variables (to run the example program given) which help us to store values between two ABAP blocks.

Because in eCATT each ABAP...ENDABAP block creates different internal function module with different contexts. So we need to declare some intermediate variables they are,

Parameter	Description	Value	VE/V	Parameter Reference	Test System	ABAP T...	Length	Dec...	Group
COUNT			V			I	10		
FILE_V	IBIP: Data transfer parameters	<INITIAL>	V	IBIPPARMS					
INT_LOC		1	V			I	10		

COUNT^[113]: Holds the number of records.

FILE_V^[114]: Holds file path of the test data file.

INT_LOC^[115]: Index to read next from the file.

I am giving sample code to get the number of records from file in eCATT. Use this code and try for **XXK01** for Basic view. It will work fine.
This is very simple ABAP code. For understanding purpose necessary comments are provided.

Step 1:

First ABAP...ENDABAP block, to get the number of records from the file and store the value in the COUNT local variable.

Obs: Este código já foi adaptado as normas de qualidade da fabrica da Aspen.

```
* ABAP BLOCK TO GET THE NUMBER OF TEST CASES
ABAP.
* TOT : holds total number of records
* FILE: holds file path of test file
DATA : TOT TYPE I VALUE 0,
      FILE TYPE STRING.
* TYPE TO HOLD TEST DATA FROM FILE
TYPES: BEGIN OF Y_LFA1,
      V_LIFNR TYPE LFA1-LIFNR , " FORNECEDOR
      V_BUKRS TYPE LFB1-BUKRS , " EMPRESA
      V_EKORG TYPE LFM1-EKORG , " ORG COMPRAS
      V_KTOKK TYPE T077K-KTOKK, " GRP CONTAS
      V_NAME1 TYPE LFA1-NAME1 , " NOME FORNEC
      V_SORTL TYPE LFA1-SORTL , " TERMO PESQUISA
      V_LAND1 TYPE LFA1-LAND1 , " PAIS
      V_KUNNR TYPE KNA1-KUNNR , " CLIENTE
      V_WAERS TYPE LFM1-WAERS , " MOEDA
      END OF Y_LFA1.

DATA : I_LFA1 TYPE TABLE OF Y_LFA1.
CONSTANTS: C_X(01) TYPE C VALUE 'X' ,
          C_FILE TYPE DYNREAD-FIELDNAME VALUE 'FILE'.

* TO OPEN FILE DIALOG FOR TEST DATA FILE
CALL FUNCTION 'F4_FILENAME'
  EXPORTING
    PROGRAM_NAME = SYST-CPROG
    DYNPRO_NUMBER = SYST-DYNNR
    FIELD_NAME   = C_FILE
  IMPORTING
    FILE_NAME    = FILE_V-PATH.

FILE = FILE_V-PATH.

* LOADING DATA FROM THE FILE
CALL FUNCTION 'GUI_UPLOAD'
  EXPORTING
    FILENAME           = FILE
    HAS_FIELD_SEPARATOR = C_X
  TABLES
    DATA_TAB          = I_LFA1.

* GETTING NUMBER OF RECORDS IN THE TABLE
DESCRIBE TABLE I_LFA1 LINES TOT.
* STORING NUMBER OF RECORDS IN LOCAL VARIABLE
COUNT = TOT.
* CLEARING INTERNAL TABLE
FREE I_LFA1.
ENDABAP.
```

Step 2:

Looping through the records count number of times and reading from the internal table and passing them to the screen field values.

This sample code explains how to read, and pass values to the screen.

```
* LOOPING THROUGH (COUNT) NUMBER OF RECORDS
DO (COUNT).

ABAP.

* V_READINDX : holds index number to read the internal table
* FILE: holds file path of test file
DATA : V_READINDX   TYPE I,
       FILE         TYPE STRING,
       INDX         TYPE I VALUE 0.
* TYPE TO HOLD TEST DATA FROM FILE
TYPES: BEGIN OF Y_LFA1,
       V_LIFNR TYPE LFA1-LIFNR , " FORNECEDOR
       V_BUKRS TYPE LFB1-BUKRS , " EMPRESA
       V_EKORG TYPE LFM1-EKORG , " ORG COMPRAS
       V_KTOKK TYPE T077K-KTOKK, " GRP CONTAS
       V_NAME1 TYPE LFA1-NAME1 , " NOME FORNEC
       V_SORTL TYPE LFA1-SORTL , " TERMO PESQUISA
       V_LAND1 TYPE LFA1-LAND1 , " PAIS
       V_KUNNR TYPE KNA1-KUNNR , " CLIENTE
       V_WAERS TYPE LFM1-WAERS , " MOEDA
       END OF Y_LFA1.

DATA : I_LFA1 TYPE TABLE OF Y_LFA1.

* WORKAREA TO HOLD THE I_LFA1 DATA
DATA : WA TYPE Y_LFA1.
FILE = FILE_V-PATH.

CONSTANTS: C_X(01)   TYPE C VALUE 'X'.

* LOADING MASTER DATA FROM THE FILE
CALL FUNCTION 'GUI_UPLOAD'
  EXPORTING
    FILENAME           = FILE
    HAS_FIELD_SEPARATOR = C_X
  TABLES
    DATA_TAB          = I_LFA1.

* INT_LOC : is a local variable hold the current index to read I_LFA1
V_READINDX = INT_LOC.
* READING I_MARA UGING ITS INDEX
READ TABLE I_LFA1 INDEX V_READINDX INTO WA.
* assigning work area values to the screen field values
V_LIFNR = WA-V_LIFNR. " FORNECEDOR
V_BUKRS = WA-V_BUKRS. " EMPRESA
V_EKORG = WA-V_EKORG. " ORG COMPRAS
V_KTOKK = WA-V_KTOKK. " GRP CONTAS
V_NAME1 = WA-V_NAME1. " NOME FORNEC
V_SORTL = WA-V_SORTL. " TERMO PESQUISA
V_LAND1 = WA-V_LAND1. " PAIS
V_KUNNR = WA-V_KUNNR. " CLIENTE
V_WAERS = WA-V_WAERS. " MOEDA
```

ENDABAP.

```
* TCD TRANSACTION  
TCD (XK01,XK01_1).
```

```
* move index position by one  
INT_LOC = INT_LOC + 1.
```

ENDDO.

With this we have finished programming. Finally we need to prepare the test data file and execute the program either in Foreground or Background mode.

Please note that data in test file should resemble the order of the elements in the internal table.

Other wise it won't work.

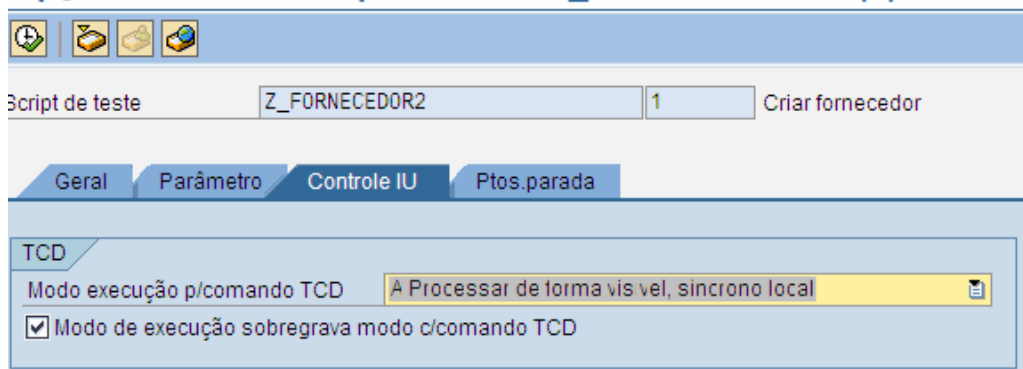
To execute the given test script, follow the steps, copy the code given and declare the necessary variables.

Criar um arquivo no excel formatando as colunas como texto e salvando como .txt (separado por tabulação)

	A	B	C	D	E	F	G	H	I
1	CASSIAAAAC	TMGA	TMGA	ZNAV	CASSIA VALERIA DE O LIMA	VALERIA	BR	CASSIAAAAC	BRL
2	ROLIMAAAAA	TMGA	TMGA	ZNAV	MAURICIO ROLIM	ROLIM	BR	ROLIMAAAAA	BRL
3									
4									

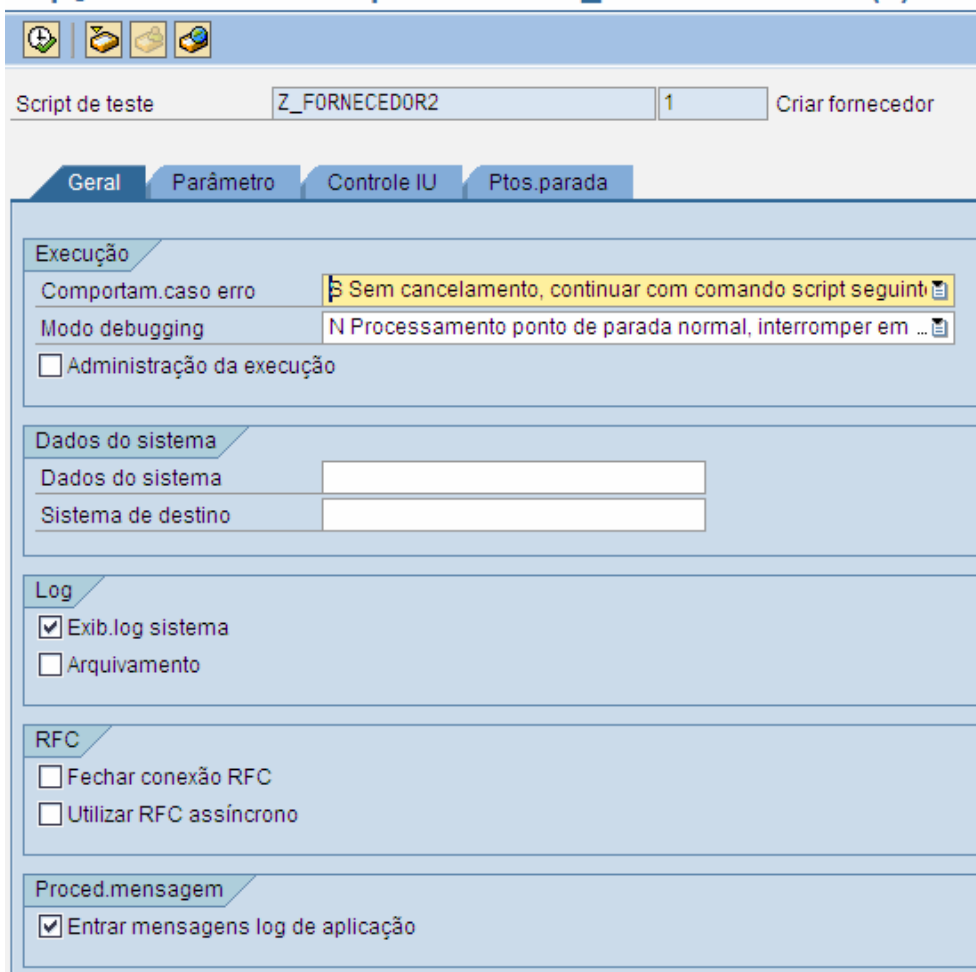
Iniciar os testes da eCATT, para visualizar a chamada da transação XK01, voltar nas Opções Iniciais da SECATT e na aba `Controle IU` e no campo `Modo de exibição p/comando TCD` selecionar `A – Processar de forma visível, síncrono local`.

Opções iniciais - Script de teste: Z_FORNECEDOR2 (1)



Opções Iniciais da SECATT e na aba `Geral` deixar as seguintes configurações.

Opções iniciais - Script de teste: Z_FORNECEDOR2 (1)



Script de teste: Z_FORNECEDOR2 1 Criar fornecedor

Geral | Parâmetro | Controle IU | Ptos.parada

Execução

Comportam.caso erro: Sem cancelamento, continuar com comando script seguinte

Modo debugging: N Processamento ponto de parada normal, interromper em ...

Administração da execução

Dados do sistema

Dados do sistema: []

Sistema de destino: []

Log

Exib.log sistema

Arquivamento

RFC

Fechar conexão RFC

Utilizar RFC assíncrono

Proced.mensagem

Entrar mensagens log de aplicação

Após a execução a seguinte tela de log será apresentada:

Exibição de log de sistema de ECATT 000000020

The screenshot displays the SAP system log for the transaction ECATT 000000020. The log is organized into a tree view with the following structure:

- 000000020 ScriptTst. Z_FORNECEDOR2 Versão 1 - SECATT [Sem interrupção] [7 sec]
 - ECD 100 MYSAPTMG04 P 640 ecc02 Windows NT ORACLE 19.07.2006 14:25:50
 - Startprofil XML-DATA-01
 - Z_FORNECEDOR2 [6,719 sec] Versão 1 Criar fornecedor
 - IMPORT Z_FORNECEDOR2 14:25:50
 - LOCAL VARIABLES
 - * ABAP BLOCK TO GET THE NUMBER OF TEST CASES
 - ABAP [3,344 sec]
 - * LOOPING THROUGH (COUNT) NUMBER OF RECORDS
 - DO 2 <- COUNT
 - 0001. 2
 - ABAP [0,141 sec]
 - * TCD TRANSACTION
 - TCD XK01 [1,984 sec N] Criar fornecedor (central)
 - SF2175 Cliente CASSIAAAG foi criado p/empresa TMGA e organização de compras TMGA.
 - * move index position by one
 - = INT_LOC = INT_LOC + 1
 - 0002. 2
 - ABAP [0,156 sec]
 - * TCD TRANSACTION
 - TCD XK01 [0,203 sec N] Criar fornecedor (central)
 - SF2175 Cliente ROLIMAAAAA foi criado p/empresa TMGA e organização de compras TMGA.
 - * move index position by one
 - = INT_LOC = INT_LOC + 1
 - ENDDO
 - EXPORT Z_FORNECEDOR2 14:25:57