DBA Training Manual

WORKFLOW

ExpertEdge Software

Info.training@cwlgroup.com

Document number:		VersionRev:	1.1
Authorized by:	Olawuwo Shade	Signature/Date:	

Document revision list

Ver.Rev	Date	Author	Description
1.00	01-02-2011	Arije Abayomi	*Original version*

Expertedge Software Limited,
14c Kayode Abraham Street,
Off Ligali Ayorinde,
Victoria Island,
Lagos.

TABLE OF CONTENTS

1 OVERVIEW

The FINACLE application software has a number of individual functions that are required for operations at a bank. These are accessed through different menu options. However, since these options are highly granular, the completion of a specific business transaction in a branch might involve a number of these menu options to be executed. The workflow script engine allows you to thread these menu options to form one logical and complete sequence. For example, to open a savings account, you can write a workflow script to thread the *CUMM*, *OAAC*, *OAACAU*, *TM* and *ICHB* menu options and create a new menu option for the same. Once the workflow menu option is initiated, the above menu options will be automatically and sequentially invoked.

1.1 FEATURES OF A WORKFLOW SCRIPT

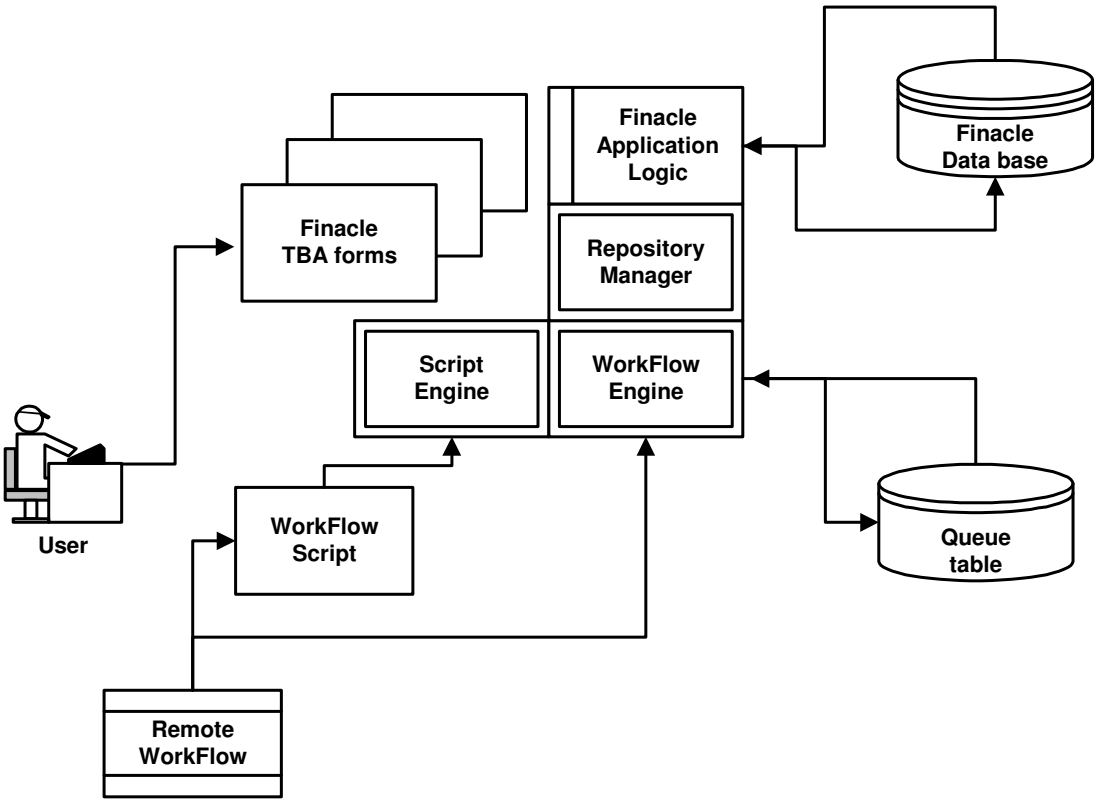
- Banks can define their own 'Threaded Menu Options' by specifying the appropriate script names.
- The WorkFlow scripting syntax will allow a generic Form (called the parameter acceptance form) to be displayed to the user to accept user-defined field values which could be used for 'branching' or providing default values and signature display.
- The WorkFlow scripting syntax will allow the execution of the base menu options one by one without the user having to invoke it
- The threaded menu options will be executed in a sequence automatically.
- The WorkFlow scripting syntax will allow the specification of default values for specific fields in the Forms. To reduce the time taken in completing an operation, you can specify default values for certain fields. You can later modify the default values populated for mandatory fields. This is the concept of ***DELAYED MANDATORY.***
- The WorkFlow scripting syntax will allow the user to suspend a 'Thread' so that he can resume it later
- The WorkFlow scripting syntax will allow the transferring of a thread from a user to another(or workgroup) to support 'threads' in which multiple personnel are involved
- You can create and queue up a new thread for another user or workgroup.

- Users will be notified immediately when a new 'thread' is queued up against their userId. When a 'thread' is queued up against a workgroup (as against a specific user), any user in that workgroup can access and work on that thread.
- When a thread is queued up against a workgroup (as against a specific user), any user in that workgroup can access and work on that thread.
- Facility to abandon a 'thread' is provided. However, this will be possible only if no 'Commits' have taken place in that thread.
- All completed threads will be audited for future reference.
- Purge functionality for the audit table has been provided
- Application Forms has been enhanced to disable the security check of 'verifier same as enterer' based on a scripting option. When this happens, an exception will be written, again, based on a scripting option.
- The creation and editing of the WorkFlow scripts can be done using any source editor
- The WorkFlow engine supports both the Character mode and the Web-based interfaces.
- WorkFlow can also be executed from remote applications by calling the scripts through remote applications and user defined executables.
- User defined listing facility in the workflow forms
- Signature display in the workflow parameter acceptance form.
- Auto verification for menu options IMC, ACM and INTTM

2 PROCESSES INVOLVED

The process involved and the process diagram is explained as below

2.1 PROCESS DIAGRAM



2.2 TERMINOLOGY EXPLAINED

TBAFORMS - This is the module that controls all online interactions with FINACLE through the interface form.

REPOSITORY MGR – This module provides the capability of storing variable values and to access them by name. It is used by the Workflow Script engine to store and pass values between menu options and by TBAForms to pass values to and from the form fields. The structure of a repository field reference is as follows:

[Repository Name].[Class Name]. [Field Name]

WORKFLOW SCRIPT ENGINE – This module is responsible for executing the specified Workflow Script (using a menu option), using interpretative logic at runtime. It also provides the capability of suspending a 'thread', forwarding a 'thread' to a specified user or workgroup and to restart threads from the point where it was previously suspended or forwarded.

The Workflow Script engine also provides the capability of executing programs written by the Bank while passing data and receiving data back from these programs.

The Workflow Script engine also provides the capability to queue up a 'new' thread for "later / back office" processing, which is especially useful in supporting the 'delayed mandatory' feature.

QUEUE TABLE – This is the data store where all uncompleted Workflow are temporarily stored, before some user pulls them up for further processing. After completion of the workflow, it is deleted from here and optionally an audit record is created.

2.3 INITIAL SETUP - OPERATING SYSTEM LEVEL

2.3.1 CUST-OPTIONS

1☐ SB_STRING
2☐ CA_STRING
3☐ LL_STRING
4☐ CC_STRING

☞ *The values for the above cust options will be available as values in the BANCS repository in the class STDIN and no other business validations are done on the same.*

2.3.2 ENVIRONMENT VARIABLES

1☐ HOST_ID
2☐ HOST_NAME
3☐ HOST_MACHINE_TYPE
4☐ WFSRE_DATA_DIR

☞ *These values are essential to ensure that the workflow is transferred to other users as workflow scripts cannot be transferred across machines or platforms OS various operating systems.*

2.4 INITIAL SETUP - APPLICATION LEVEL

2.4.1 REFERENCE CODE

None

2.4.2 EXCEPTIONS

Pending Business Trans Excp: This exception will be set in SCFM and raised whenever a workflow with a priority value higher than that set for the field "*WFS EOD Exception Priority From" is* pending completion.

2.5 PARAMETER SETUP

User defined list can be defined using generic list form bafl0001. The select clause, from clause and where clauses have to be populated before calling the generic form bafl0001. The form title and the column format can be specified in the script. A demonstration script (demoForListAndSignature.sscr) has been provided to understand this functionality.

Signature Display: Signature display facility has been provided on F9 key for workflow parameter acceptance forms. A new pg0 field "acct_num" has been added in workflow parameter acceptance forms and the signature will be displayed for the account(Foracid) in the field. This field has to be populated with appropriate account number on pressing F9 key to get signature display.

Auto verification of the modifications/entry made through IMC/INTTM/AMC can be handled using the following scripts.

IMCAutoVerification.sscr - IMC

INTTMV.sscr - INTTM

ACMV.sscr - ACM

Note: ACMV.sscr calls another two scripts getAcctNum.scr and acm_sub.scr existing in the sample scripts directory.

The parameter setup for workflow are discussed in detail below

2.5.1 STEPS INVOLVED IN CREATION OF A WORKFLOW

1☐ Decide on the work flow required

2☐ List out the menu options to be threaded

3☐ Identify the form names for the menu option (you can invoke the menu option through the application and the form name will be displayed at the to right corner)

4☐ Identify the fields to set the values

5☐ Get the value of the field name in the form for the above fields using the utility dispfields

6☐ Decide on the field to be brought up in the initial parameter form to be entered by the user

7☐ Using any unix editor like vi create a workflow script. The filename must end with .scr. the script must be located either in the $TBA_PROD_ROOT/cust/scripts directory or in the $TBA_PROD_ROOT/cust/INFENG/scripts directory.

8☐ Create a mod file for the above workflow and add the same in tba_minp.dat file

9☐ Rerun the menu generation program to add the new menu option

10☐ Login to the application and test the script.

2.6 INVOKING A WORKFLOW ITEM

Workflow can be invoked from any of the following methods

1☐ Through a menu option calling a Workflow Script

2☐ By a workflow script creating a new workflow item for a user and queuing the same to that particular user. the other user would execute the menu option DSPWFQ and explode from that queued up item to continue the workflow

3☐ By a FINACLE Non-workflow event script creating a workflow item for a user.

4☐ By a remote application or FINACLE Workflow script executing a remote workflow item. This could be from some custom application built by the bank or an application such as **Bankaway™**.

2.7 REPOSITORY STRUCTURE FOR WORKFLOW EXECUTION

The WFSINPUT, WFSOUTPUT, WFSINPARAMVAL, WFSINPARAMLEN, WFSLIST and WFSOUTPARAM classes in the STDWFS repository can be accessed by all WorkFlow scripts.

The following fields in the STDWFS.WFSINPUT class have default values as mentioned below when the workflow item starts. Any changes done to them are preserved during the lifetime of that workflow item.

Field Name	Description
"CurrentUserId"	The FINACLE User Id of the process. It is used for reference only.
CurrentTranId	The Workflow Transaction Id assigned to the current item. It is used for reference only.
CallScriptName	The name of the script associated with the workflow (either through a menu option definition, or through a parameter for CreateWFItem or ExecuteWFS functions). It is used for reference only.
IgnoreSameUserVerifyFlg	Indicates if auto-verification by the same user will be allowed. Default value of this field is *N.* It can be modified. **Valid Values:** *Y* – Yes *N* – No
PostVerificationChkReqd	If this is set to *Y*, a VAT record will be written in the auto-verification process. Default value of this field is *N.* It can be modified. **Valid Values:** Y – Yes N – No
LastInvokedMenuOption	Indicates the last menu option called from within the script. It is set when the user function CallMenuOption is used. It is used for reference only.
NextStep	The next step to be executed in the script. Set when the user function CallmenuOption or CheckPoint is called. It is used for reference only.
CalledMenuOption	Indicates the menu option that started the work flow item, if started from a menu option or created from a menu option script using WFS_CreateWfItem function. Else, *EVTWFS* if started through EVTSCR_CreateWfItem user function, or through WFSRE_Execute_WFS function. It is used for reference only.
WfsStatus	Indicates the status of the workflow. If the status is *D* when exiting the script, the workflow item is deleted from the system, else it is retained in BTQ table. Default value of this field is ' '. Valid Values: D – Delete ' ' – Retain in BTQ table
CurrAuthUserId	Modifiable. Used in TransferUser function if userid parameter is not specified. By default, the supervisor of the current user as defined in the UPR table.

WorkFlowDesc	Reference only. By default set, to the menu title of the menu that started the workflow, if workflow started from a menu option, else ' '.
StepCommitFlg	Reference only. Indicates whether in a CallmenuOption function a commit to the database was done or not. Set to 'N' at the start of every CallmenuOption function. Can be used to retry a previous step if an expected commit was not done.
WfsMode	Reference only. If workflow is being executed by daemon then "DM", else if being executed remotely then "RM" else " ".

The following fields in the STDWFS.WFSOUTPUT class has the following default values.

Field Name	Remarks
ErrorCode	" "
WorkFlowDesc	Menu Option Description
ErrorMesg	" "

For WFSINPARAMVAL, WFSINPARAMLEN and WFSOUTPARAM classes there are no standard fields. These classes will be used inside a script to accept initial data by adding appropriate field in WFSINPARAMVAL and WFSINPARAMLEN. Output will be available in WFSOUTPARAM.

Also, for WFSLIST there are no standard fields. It can be used for listing on a particular field. The following listing is possible:

1. **Listing on all RRCDM Codes**

   Eg: STDWFS.WFSLIST.CustTitle = "REF_CODE_LIST|45"

   Where RRCDM Type 45 is for Customer Tittle. If any other other list is to be defined then only RRCDM type can be changed. For eg: if instead of 45 a user puts 01 then listing is on City code.

2. **Listing on Account Numbers**

   Eg: STDWFS.WFSLIST.AccountNum = "ACCT_MAST_LIST"

3. **Listing on Customer ID**

   Eg: STDWFS.WFSLIST.CustId = "CUST_MAST_LIST"

4. **Listing on Scheme Code**

   Eg: STDWFS.WFSLIST.SchemeCode = "SCHM_MAST_LIST"

5. **Listing on Set ID**

Eg: STDWFS.WFSLIST.SetId = "SET_MAST_LIST"

6. **Listing on SOL**

Eg: STDWFS.WFSLIST.SOL = "SOL_MAST_LIST"

7. **Listing on Bank Code**

Eg: STDWFS.WFSLIST.Bank = "BANK_MAST_LIST"

8. **Listing on GL Sub Head**

Eg: STDWFS.WFSLIST.GlSub = "GLSUB_MAST_LIST"

Another repository TEMPWFS is also available at the start of a workflow script. It has the same structure as STDWFS. It has been provided to populate data when a new WFS item is created from within the work flow script using WFS_createWFItem or EVTSCR_CreateWFItem user functions. To begin with, all the standard fields (as described above) will have the same values as the default values in the STDWFS repository. Modifications done to the above fields or new class.fields created in the WFSTEMP repository will be preserved and appear automatically under the STDWFS repository when the created item is taken up for execution.

ADTWFS is a repository, which has to be created in the script if at the time of writing the Workflow audit additional audit information needs to be stored. Any number of 'string' classes and fields can be created in this repository for storing along with the audit. If at the end of script execution (STDWFS.WFSINPUT.WfsStatus = 'D') this repository is found to exist an audit will be written in the BTA table with all fields in the repository preserved for later viewing.

Apart from these repositories, repositories INTBAF and OUTTBAF are used to interact with Finacle forms. (e.g. get the customer id after commit in CUMM menu option)

For these repositories there are no standard classes and fields. These repositories will be used to populate values from repository to application forms and vice versa. See document on Userhooks for more details of TBAF_SetValue, TBAF_SetAttrib, TBAF_GetValue.

INTEMP and OUTTEMP repositories have the same structure as INTABF and OUTTBAF repositories. They have been provided to populate data when a new WFS item is created from within the workflow script script using WFS_CreateWFItem or EVTSCR_CreateWFItem user functions. All class.fields created in these repositories will be preserved and appear automatically under INTBAF and OUTTBAF repositories when the created item is taken up for execution.

2.8 PROVIDING HELP ON A PARTICULAR FIELD

In work flow parameter acceptance form, we can provide help on a particular field. If user press key F1 that help message is displayed on the message area.

Eg: If user wants to provide help message on Customer Tittle code, in that same it appears in CUMM screen then can use user hook urhk_TBAF_SetAttrib.

sv_a = urhk_TBAF_SetAttrib("bafi2020.datablk.field_1|?490")

where bafi2020 is the form number of parameter acceptance form . In that in data block and the specific field help message will be provided. Code 490 is taken from mmsg table where HLP490 is for Help message "Enter valid title code. Press <LIST> to list valid codes".

2.9 UNATTENDED WORKFLOW EXECUTION

This feature is an extension of the existing workflow scripting capabilities. It allows those workflows which can execute without any user input to be queued up to a daemon (virtual user) process, which will execute the work flow just like any user would have, except that no terminal is locked up for this.

2.9.1 ASSIGNING WORK GROUP

A work group called "WFSDM", is by default assigned to the daemon and this is the work group under which it looks for items to execute. However, it is possible for the bank to assign different work groups to this daemon by assigning it in the GetuserWFSWorkGroup.scr

2.9.2 ERROR HANDLING

It is possible that during execution of a workflow script by the daemon, some errors may be encountered. In case of fatal errors, the workflow item may have

actually executed some steps and committed updates to the database before the fatal error occurred. If so, this will have to be reversed manually at the back-end.

This has to be done manually by using the MNTWF menu option and inquiring on all locked records under the workgroup(s) assigned to 'unattended' executions. After determining aborted items from this list, the lock needs to be released and the work group of the item changed to a suitable back-office work group so that a back-office personnel can examine the state of the work flow and determine the appropriate course of action.

In case of controlled errors (not fatal errors), the work flow script is queued up for corrective action under a workgroup which determined by replacing the last character of the assigned work group by 'R'. For example if "WFSDM" is the workgroup assigned for "unattended" execution, then all the rejects will be queued up under the work group "WFSDR" for taking corrective action. The appropriate back-office personnel need to be assigned to this (these) workgroup(s) so that they can use the DSPWFQ menu option to process these rejects.

2.9.3 SETTING-UP FOR UNATTENDED WORKFLOW EXECUTION

Unattended workflow is supported through the daemon executable "dameonce" located in the directory *${TBA_PROD_ROOT}/cust/INFENG/exe/dameonce.* If a language other than INFENG is being used, the executable translated for that language should be used. The executable will be present in *{TBA_PROD_ROOT}/cust/<lang_code>/exe/ directory.* The Bancs@Web server must be configured to start up and maintain the desired number of such daemons through the following steps:

1☐ Set up the web application server (b2kcomp).

2☐ Open the Web application server (b2kcomp) configuration file (termtypes.cfg) located in the data directory of b2kcomp setup.

3☐ Create a new service section for this service by adding the following line: "UTP_SECTION=TERM_DA"

4☐ Set the path of the daemon exe by adding the following line below that: "PATH=<full path for dameonce>"

5☐ Based on the activity on your system determine the number of daemons that need to be brought up.

6☐ Set both the starting and maximum number of instances of the daemon to that number by adding the following two lines:

7☐ "START_NUM_INST=<number>"

8☐ "MAX_NUM_INST=<number>"

9☐ Add the following line to the STDENV section

- WFSDM_SLEEP_TIME=nnn is the number of seconds the work flow daemon should sleep before checking on the queue again when it is empty. Default value is 30 seconds.

10☐ Restart the server if it is running.

2.9.4 CLEANING UP FILES LEFT BY REMOTE EXECUTION / UNATTENDED EXECUTION PROCESSES

The Remote workflow execution and unattended workflow execution processes create working directories for themselves when they come up. Since they are daemons they typically, keep running until the entire server is brought down or in the case of some fatal errors. Hence, there is a need for cleaning up the directories that are left behind by these processes when they are terminated. An executable called cleandame has been provided which periodically keeps looking for directories that have been created by these daemon processes and cleans them up if the corresponding processes have been terminated.

A com script called `startcleandame.com` has been provided in $TBA_PROD_ROOT/cust/INFENG/com directory which brings up the cleanup daemon. This com script should be executed as a part of the database bring up procedure in all datacenters which use the remote work flow execution or unattended work flow execution feature.

3 GENERAL USER ACTIVITIES

3.1 DISPLAY PENDING WORKFLOW ITEMS - DSPWFQ

This menu options enables the user to view the workflow items which are pending against his user id. The user can explode on any item displayed in the list if the user id logged in is same as the next user id or the user belongs to the work group which has the requisite permission to continue with the operation. The list does not display items that are locked. The items are displayed sorted in the order of logged on user id, priority value and tran id and then by logged on user workgroup, priority value and tran id.

After completion of a workflow item, the cursor comes back to the screen with the completed workflow not listed.

```
+-------------------------------------------------------------------------------+
|bafi2007                Display Pending Workflow Items              12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
| Workflow Id Workflow Description                     Next UserId     Group    |
| ----------- ---------------------------------------- --------------- -----    |
| AAX13099    ATM SDS Maintenance                      BALU1                    |
| AAX13100    ATM SDS Maintenance                      BALU1                    |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+

.
```

3.2 WORKFLOW ITEMS INQUIRY

MENU OPTION: DSPFWI

This menu option is useful when the user would like to inquire using criteria such as Initiating user id, Initiating sol id or range of workflow audit date or by status of workflow transaction or workflow item locked by a user or workflow transaction with a description which matches the string that is entered.

```
+-------------------------------------------------------------------------------+
|bafi2008                    Workflow Items Inquiry                  12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|                                                                               |
| Initiating User Id              : BALU1                                       |
|                                                                               |
| Initiating Sol Id               : SCGL                                        |
| Work Group                      :                                             |
|                                                                               |
| Workflow Tran Audit Date (From) : 26-11-1998                                  |
|                                                                               |
| Workflow Tran Audit Date (To)   :                                             |
|                                                                               |
|                                                                               |
| Status of Workflow Transaction  : A ALL                                       |
|                                                                               |
| Locked User Id                  :                                             |
|                                                                               |
|                                                                               |
| Workflow Tran. Description like :                                             |
|                                                                               |
+-------------------------------------------------------------------------------+
```

Upon entering the criteria and accept, the following screen would be displayed. If the audit date is available it indicates that particular work flow id is complete otherwise a work flow id is incomplete

```
+-------------------------------------------------------------------------------+
|bafi2008                    Workflow Items Inquiry                  12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
| Workflow Id    Audit Date  ----- Initiating -----      Last Worked UserId     |
| Description                User Id         Sol Id      Locked UserId          |
|-------------------------------------------------------------------------------|
| AAX13083                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                                           |
| AAX13084                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                    BALU1                  |
| AAX13082                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                    BALU1                  |
| AAX13080                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                    BALU1                  |
| AAX13081                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                                           |
| AAX13079                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                    BALU1                  |
| AAX13077                   BALU1           SCGL        BALU1                  |
| ATM SDS Maintenance                                    BALU1                  |
|                                                                               |
+-------------------------------------------------------------------------------+
```

The explode on any item will display the following screen.

```
+-------------------------------------------------------------------------------+
|bafi2009                  Workflow Items Inquiry                12-02-2000  |
|                                                                               |
|-------------------------------------------------------------------------------|
| Function    I       INQUIRE                                                   |
| Workflow Id AAX13083                  Audited date                            |
|-------------------------------------------------------------------------------|
|                                                                               |
| Workflow description         ATM SDS Maintenance                              |
| Initiating.....                                                               |
|   User Id                BALU1                                                |
|   Branch                 KUMTA                                                |
|   Host Id                IBMS                Locked User Id                   |
|   Host Type              SCOSV               Next WorkGroup                   |
| Last Worked User Id      BALU1               Next User Id   BALU1             |
| Completed Flag           N                   Priority Value                   |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

Any item can be chosen and explode to obtain further information such as workflow description, Initiating user id, branch, host id, host type, last worked user id, Locked user id, next work group, next user id, completed flag and priority value.

At this stage itself, only for incomplete workflow, the user can choose an item that is locked and unlock it and also go to modify mode and enter the next workgroup, next user id and priority value.

If the user presses explode key from this item, another screen is displayed which indicates the repository name, repository class, field name and field value.

```
+-------------------------------------------------------------------------------+
|bafi2025                   Workflow Items Inquiry                 12-02-2000   |
|                          Repository Data Display Form                         |
|-------------------------------------------------------------------------------|
|  Repository Name  STDWFS         Class Name  WFSINPUT                         |
|-------------------------------------------------------------------------------|
|  Field Name                      Field Value                                  |
|  WfsMode                         []                                           |
|  CurrentUserId                   [BALU1]                                      |
|  NextAppUserId                   [BALU1]                                      |
|  NextAppUserWorkGroup            [ ]                                          |
|  CurrAuthUserId                  [INSTALL]                                    |
|  StepCommitFlg                   [N]                                          |
|  NextStepAfterCommit             [0]                                          |
|  WfsStatus                       [ ]                                          |
|  RestartMenuOption               [ ]                                          |
|  WorkFlowDesc                    [          FINACLE User Menu]             |
|  CurrentTranId                   [AAX12775]                                   |
|  CallScriptName                  [wfstest.scr]                                |
|  PostVerificationChkReqd         [N]                                          |
|  IgnoreSameUserVerifyFlg         [N]                                          |
|                                                          2 of  7 Classes      |
+-------------------------------------------------------------------------------+
```

This is a multi rec and the user can use the key for next block to see repository details. For incomplete workflow items, STDWFS is shown whereas for completed workflow ADTWFS will be shown.

The repository classes available under STDWFS are

1☐ WFSINPUT
2☐ WFSOUTPUT
3☐ WFSINPARAMVAL
4☐ WFSINPARALEN
5☐ WFSOUTPARAM

3.3 WORKFLOW MAINTENANCE

MENU OPTION: MNTWF

This menu option is used for Inquiring and modifying the status of a workflow id if the workflow id is known. Using this option, the user can unlock a workflow which has been locked and the workflow item cannot be completed by the user who has locked the same. Through this menu option, the user can also modify the next work group, next user id and priority value.

```
+-------------------------------------------------------------------------------+
|bafi2009                 Workflow Items Maintenance              12-02-2000  |
|                                                                               |
|-------------------------------------------------------------------------------|
| Function    M       MODIFY                                                    |
| Workflow Id AAX13083                   Audited date                           |
|-------------------------------------------------------------------------------|
|                                                                               |
| Workflow description         ATM SDS Maintenance                              |
| Initiating.....                                                               |
|   User Id                BALU1                                                |
|   Branch                 KUMTA                                                |
|   Host Id                IBMS                Locked User Id                   |
|   Host Type              SCOSV               Next WorkGroup                   |
| Last Worked User Id      BALU1               Next User Id   BALU1             |
| Completed Flag           N                   Priority Value                   |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

3.4 VERIFICATION AUDIT MAINTENANCE

MENU OPTION: VAM

```
+-------------------------------------------------------------------------------+
|bafe2021                Verification Audit Maintenance             12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|                                                                               |
| Function                        :                                             |
|                                                                               |
| Exception Date From             : 26-11-1998                                  |
| Exception Date To               :                                             |
|                                                                               |
| Verifier SolSet Id              :                                             |
|                                                                               |
| Verifier User Id                :                                             |
|                                                                               |
| Approver User Id                :                                             |
| Approval Date                   :                                             |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+


Valid values I-Inquire, A-Approve
```

This menu option can be used to Inquire or Approve a workflow item which was verified by the user who has created and authorised an customer master.,. The restriction that the same user cannot create and verify a customer master in workflow by setting the user hook IgnoreSameUserVerifyFlg to Y in the workflow

script. When the flag is set, the user who has created an customer master can also verify the same. Even though this facilitates quicker completion of a task, the same needs to be veified by another user. This is achieved by setting the PostVerificationChkReqd to Y. All such verifications due are displayed by the menu option VAM. The user can approve by selecting an item from the list and pressing the commit key after viewing the details.

Press the <ACCEPT> key to get the list of workflow items which need to be approved. If the approver user id and date are displayed then the record has been approved else it has to be approved.

```
+-------------------------------------------------------------------------------+
|bafe2021                Verification Audit Maintenance              12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|                    ------- Verifier ------------   ------- Approver --------- |
|   Exception Date  User Id         Sol Id          User Id         Date       |
|-------------------------------------------------------------------------------|
|   02-12-1998      BALU2           SCGL                                       |
|   02-12-1998      BALU2           SCGL                                       |
|   02-12-1998      BALU2           SCGL                                       |
|   02-12-1998      BALU2           SCGL                                       |
|   02-12-1998      BALU1           SCGL                                       |
|   02-12-1998      BALU1           SCGL                                       |
|   03-12-1998      BALU1           SCGL                                       |
|   03-12-1998      BALU1           SCGL                                       |
|   03-12-1998      KP1             SCGL                                       |
|   03-12-1998      KP1             SCGL                                       |
|   03-12-1998      TRG24           SCGL                                       |
|   03-12-1998      TRG24           SCGL                                       |
|   03-12-1998      BALU1           SCGL                                       |
|   03-12-1998      BALU1           SCGL                                       |
|                                                                               |
+-------------------------------------------------------------------------------+
```

<EXPLODE> on any of the items brings up the following screen :

```
+-------------------------------------------------------------------------------+
|bafe3023                Verification Audit Maintenance              12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|  Function         I INQUIRE               Reference No.     AA1802            |
|-------------------------------------------------------------------------------|
|  Function         A ADD                                                       |
|  Service Outlet   SCGL     INFOSYS TOWERS                                     |
|  A/c No.                                                                      |
|  Gl Subhead Code                                                              |
|                                                                               |
|   Table    Key                                                                |
|     CMC     ABC000049/INR                                                     |
|     CMG     ABC000049                                                         |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                        Entered By BALU2       |
|  Exceptns? N                                           Entered On 04-08-1999  |
|  Remarks   New Record Created                          Authrzd By BALU2       |
+-------------------------------------------------------------------------------+
```

From the above the user can explode to see the particulars as indicated in the following screen:

```
+-------------------------------------------------------------------------------+
|bafe3023                Verification Audit Maintenance              12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|  Function         I INQUIRE               Reference No.     AA1802            |
|-------------------------------------------------------------------------------|
|  Field Name                       Old Value/New Value                         |
|  ------------------------------  ------------------------------------------   |
|  RECORD ADDED                                                                 |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

3.5 WORKFLOW AUDIT PURGE

MENU OPTION: PUWF

This menu option is used to purge workflow items. The workflow items keep increasing both in the cases of display of workflow items and verification audit maintenance. Hence the related tables need to be purged for efficient handling of the same. The tables that get purged are

a) BTA
b) VAT

```
+-------------------------------------------------------------------------------+
| bafp6006                    Workflow Audit Purge                   12-02-2000|
|                                                                               |
|-------------------------------------------------------------------------------|
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|             Date                                                              |
|                                                                               |
|             Do Purge ?                                                        |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

3.6 IDENTIFYING THE FIELD NAMES IN A FORM

A utility called "dispfields" is provided to identify the field names in a form. The syntax for dispfields is as follows

dispfields –1$TBAF_DEFAULT/P.crt –2 $TBA_FORMS–3 <form name>

The environment variable TBA_FORMS point to the forms directory and TBAF_DEFAULT/P.crt point to the key mapping file. If the site setup is different e.g. instead of P.crt the key map Z.crt might be used. In that case change the syntax according.

The dispfields utility also accepts certain additional parameter that are optional

-4 is for listfilename which contains the forms to be displayed.

-5 is for all the forms in the forms directory.

-6 is for the print file name where the output can be directed to a file instead of on screen.

-7 is for the error file name to redirect the errors to a file instead of screen

By default TBA_FORMS will point to /wd/dev/cust/forms

TBAF_DEFAULT will point to /wd/dev/tbaf/default but it could be different at sites where /wd/dev/is the value of $TBA_PROD_ROOT

AN EXAMPLE.

To find the value of the field name for customer short name field in CUMM menu option follow the following steps.

1☐ Identify the form name. the form name will appear on the right corner of the form. the form name for CUMM is baff0009.

2☐ at the unix prompt invoke the program dispfields with syntac as shown below

dispfields –1$TBAF_DEFAULT/P.crt –2 $TBA_FORMS –3 baff0009

3☐ The output will be as shown below.

```
+-------------------------------------------------------------------------------+
| baff0009          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx         xxxxxxxxxx|
|                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx                   |
|-------------------------------------------------------------------------------|
| Function     _   xxxxxxxxxxxxxxxxxxxx                                         |
| Customer Id  _________                                                        |
|-------------------------------------------------------------------------------|
| Customer Name _____ _______________________ * Short Name         __________   |
| Type          _____ xxxxxxxxxxxxxxxxxxxxxx   A/c Manager     _______________ |
| Status        _____ xxxxxxxxxxxxxxxxxxxxxx   Status as on       __________   |
| Group         _____ xxxxxxxxxxxxxxxxxxxxxx   Gender             _ xxxxxxxx   |
| Occupation    _____ xxxxxxxxxxxxxxxxxxxxxx   Non Resident?      _            |
| Constitution  _____ xxxxxxxxxxxxxxxxxxxxxx   Staff?   _                       |
| Staff No      __________ xxxxxxxxxxxxxxxxxx * Minor?   _ Suspen? x            |
| Bank Code     ______ xxxxxxxxxxxxxxxxxxxxx   Trade Finance Customer? _        |
|  Introducer's Details:                       Nat.Id.Card No _________________ |
| Customer Id.  _________ xxxxxxxxxxxxxxxxx  * Date Of Birth  __________       |
| Name          _____  _______________________ * Marital Status?  _             |
| Introd.Status _____  xxxxxxxxxxxxxxxxxxxxxx   First A/c Date     xxxxxxxxxx   |
| Frequency For Statement  _/_/_/__/_          Modified Times     xxx          |
| Enter Option  _  xxxxxxxxxxxxxxxxxxxxxxxxx                                    |
+-----------------------------------------------------------------( ==>> )--+

 0001/0001 |             datablk1.cust_short_name              | baff0009 01/12
```

4☐ Position the cursor at the field for which you want to know the field name in the form. the program will display the field name at the bottom as shown in the screen above.

5☐ To know more details about the field press the F9 key and the program will display all the attributes of the field as shown in the screen below.

```
+-----------------------------------------------------------------------------+
| baff0009          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx         xxxxxxxxxx|
|                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx                   |
|-----------------------------------------------------------------------------|
| Function     _   xxxxxxxxxxxxxxxxxxxx                                       |
| Customer Id  _________                                                      |
|-----------------------------------------------------------------------------|
| Custome+-------------------------------------------------------------+_______  |
| Type   |Form Name    - baff0009                                      |_________ |
| Status |Block Name   - datablk1                                      |_______   |
| Group  |Field Name   - cust_short_name                               |xxxxxxx   |
| Occupat|Page/Row/Col - 01/08/67     Total/Display Length - 010/10    |          |
| Constit|Protected    - NO           Mandatory            - YES       |          |
| Staff N|Hidden       - NO           Entry Allowed        - YES       |          |
| Bank Co+-------------------------------------------------------------+? _       |
|  Introducer's Details:                        Nat.Id.Card No _______________ |
| Customer Id.  _________ xxxxxxxxxxxxxxxxx  * Date Of Birth  __________       |
| Name          _____  _______________________ * Marital Status?  _             |
| Introd.Status _____  xxxxxxxxxxxxxxxxxxxxx   First A/c Date     xxxxxxxxxx   |
| Frequency For Statement  _/_/_/__/_          Modified Times     xxx          |
| Enter Option  _  xxxxxxxxxxxxxxxxxxxxxxxxx                                  |
+-----------------------------------------------------------------( ==>> )--+

 0001/0001 |             datablk1.cust_short_name              | baff0009 01/12
```

6☐ To find help press F1 key.

☞ *If the key map file is different or customised by the bank then the appropriate key set for the field key-f9 must be used for more details.*

7☐ in the above example the field name will be baff0009. datablk1.cust_short_name

3.7 MENU GENERATION FOR A NEWLY CREATED WORKFLOW MENU OPTION

The workflow script created if executed through the application (other than Remote Workflow or Workflow daemon process) should be called from a menu option. A Sample mod file for a workflow menu option is as shown below

3.8 A SAMPLE MOD FILE

A Sample Mod file for menu generation is given below.

```
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
; MOD record for (ATM SDS Maintenance) Appl: GU
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
MOD
WATM
W
ATM SDS Maintenance
NULL wfstest.scr NULL
C
26 160
BT TT FT MT
NULL
M
NULL
NULL
NULL
NULL
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NULL
NULL
C
~
```

The type must be "W" and the script name must be mentioned as shown in the field above.

4 SAMPLE WORKFLOW SCRIPT

A sample workflow script present in the $TBA_PROD_ROOT/sample/scripts directory is given below with explanations provided at appropriate places. The script name is SBCAOP.scr.

SCRIPT NAME: SBCAOP.SCR

```
<--start
#*****************************************************************************
#  This is a Workflow script for opening Savings Bank and Current Accounts.
#
#  This script calls the following menu options in the order mentioned.
#  1. parameter acceptance form
#  2. Customer Master creation  if customer is new.
#  3. Auto-verification of Customer Creation
#  4. Queueing up Customer modification work flow for entering other details
#  5. Account opening for the above mentioned type of accounts.
#  6. Transaction Maintenance for posting inital deposit transaction posting.
#  7. Verification of the account opened
#  8. If cheque book is to be issued to account (decided based on
#                 scheme level parameters) then issue cheque book.
#  9. Verification of cheque book issue.
# 10. Create audit trail of the work flow.
#
#  Transfer of control to the verifier allows the user to choose the
#  user to whom the control should be passed and it creates a entry
#  for the flow in the pending transactions queue of that user.
#
#  Also the script defines what field values will be stored in the audit
#  repository for the workflow.
#*****************************************************************************

TRACE ON
# the above command TRACE ON start writing a trace file by the scipt name.trc file
#in the users home directory


#  --------------------
#  Initialise variables
#  --------------------

# TM Func Code
sv_r = "A"
# tran_type
sv_s = "C"
# tran_sub_type
sv_t = "NR"




#  --------------------------------------
#  Restart logic based on steps completed
#  --------------------------------------

sv_b = cint(STDWFS.WFSINPUT.NextStep)

if (sv_b == 1) then
   GOTO STEP1
endif

if (sv_b == 2) then
   GOTO STEP2
endif

if (sv_b == 3) then
   GOTO STEP3
endif
```

```
if (sv_b == 4) then
   GOTO STEP4
endif

if (sv_b == 5) then
   GOTO STEP5
endif

if (sv_b == 6) then
   GOTO STEP6
endif

if (sv_b == 10) then
   GOTO STEP10
endif

if (sv_b == 11) then
   GOTO STEP11
endif

if (sv_b == 12) then
   GOTO STEP12
endif

if (sv_b == 13) then
   GOTO STEP13
endif

if (sv_b == 14) then
   GOTO STEP14
endif

if (sv_b == 15) then
   GOTO STEP15
endif

if (sv_b == 16) then
   GOTO STEP16
endif

if (sv_b == 17) then
   GOTO STEP17
endif

if (sv_b == 18) then
   GOTO STEP18
endif

if (sv_b == 19) then
   GOTO STEP19
endif

# STEp1, STEP2 etc.. are labels defined later in the script.
#as shown in the line below


STEP1:
STDWFS.WFSOUTPUT.WorkFlowDesc="SB/CA Account Opening  - "
#  ---------------------------------------
#  Create the required repository Classes. class GLBDATA to hold all the
#  data required across steps of the script.
# the user is free to name the CLASS with s different name also.
# But the repository must be STDWFS only.
#  ---------------------------------------
```

```
sv_d = CLASSEXISTS("STDWFS", "GLBDATA")
if (sv_d == 0) then
   CREATECLASS("STDWFS", "GLBDATA", 5)
# the number 5 indicates the data type of class . 5 stands for String Type data.
endif

#  ------------------------------------------------------------
#  Accept the default values for all the menu options in the flow
#  ------------------------------------------------------------

#  ----------------------
#  Accept Screen literals
#  ----------------------

STDWFS.WFSINPARAMVAL.FormTitle="Parameter Acceptance Form"
STDWFS.WFSINPARAMVAL.ErrorMesg=""
STDWFS.WFSINPARAMVAL.custId="Customer Id"
STDWFS.WFSINPARAMVAL.glSubHead="GL Sub Head"
STDWFS.WFSINPARAMVAL.schemeCode="Scheme Code"
STDWFS.WFSINPARAMVAL.depositAmount="Deposit Amount"
STDWFS.WFSINPARAMVAL.tranType="Cash/Transfer"
STDWFS.WFSINPARAMVAL.acctNumber="Transfer A/c Num"
STDWFS.WFSINPARAMVAL.chqBookReqd="Cheque Book Reqd?"

#  ---------------------------------
#  Accept Data Length for parameters
#  ---------------------------------

STDWFS.WFSINPARAMLEN.custId="9"
STDWFS.WFSINPARAMLEN.glSubHead="5"
STDWFS.WFSINPARAMLEN.schemeCode="5"
STDWFS.WFSINPARAMLEN.depositAmount="15"
STDWFS.WFSINPARAMLEN.tranType="1"
STDWFS.WFSINPARAMLEN.acctNumber="16"
STDWFS.WFSINPARAMLEN.chqBookReqd="1"

#  -------------------------------------
#  Put in  Default Values for parameters
#  -------------------------------------

STDWFS.WFSOUTPARAM.custId=""
STDWFS.WFSOUTPARAM.glSubHead="SBGEN"
STDWFS.WFSOUTPARAM.schemeCode="SBGEN"
STDWFS.WFSOUTPARAM.depositAmount="100.00"
STDWFS.WFSOUTPARAM.tranType="C"
STDWFS.WFSOUTPARAM.acctNumber=""
STDWFS.WFSOUTPARAM.chqBookReqd="Y"

#  ----------------------------------
#  Call the Parameter Acceptance form in a loop so that if the user
#  enters some wrong values he will get a chance to correct them before
#  procedding with the work flow
#  ----------------------------------
STDWFS.GLBDATA.ErrorFlg = "Y"
while (STDWFS.GLBDATA.ErrorFlg =="Y")
sv_a = urhk_WFS_ShowParamAcptFrm("")
STDWFS.WFSINPARAMVAL.ErrorMesg=""
#  ---------------------------------------------------------------------------
#  Do some validations on the parameters accepted
#  ---------------------------------------------------------------------------
if (STDWFS.WFSOUTPARAM.tranType != "C") then
if (STDWFS.WFSOUTPARAM.tranType != "T") then
       STDWFS.WFSINPARAMVAL.ErrorMesg="Transaction type can be Cash or Transfer
only"
endif
```

```
endif
if (STDWFS.WFSOUTPARAM.tranType == "C") then
if  (LTRIM(STDWFS.WFSOUTPARAM.acctNumber) != "") then
       STDWFS.WFSINPARAMVAL.ErrorMesg="Account number is to be entered only if
transaction type is Transfer"
endif
endif
if  (LTRIM(STDWFS.WFSINPARAMVAL.ErrorMesg) == "") then
   STDWFS.GLBDATA.ErrorFlg="N"
endif
do


#  ---------------------------------------------------------------------
#  Set Ignore same user error flag to allow the same user to enter and
#  verify details in a menu option.
#  Also set Post verification check required to yes so
#  that there is a Verification Audit record created for
#  same user verification
#  ---------------------------------------------------------------------

STDWFS.WFSINPUT.PostVerificationChkReqd="Y"
STDWFS.WFSINPUT.IgnoreSameUserVerifyFlg="Y"

STDWFS.GLBDATA.custId = STDWFS.WFSOUTPARAM.custId
STDWFS.GLBDATA.glSubHead = STDWFS.WFSOUTPARAM.glSubHead
STDWFS.GLBDATA.schemeCode = STDWFS.WFSOUTPARAM.schemeCode
STDWFS.GLBDATA.tranType = STDWFS.WFSOUTPARAM.tranType
STDWFS.GLBDATA.chqBookReqd = STDWFS.WFSOUTPARAM.chqBookReqd
STDWFS.GLBDATA.tranAmount = STDWFS.WFSOUTPARAM.depositAmount
STDWFS.GLBDATA.acctNumber = STDWFS.WFSOUTPARAM.acctNumber
STDWFS.GLBDATA.oldCust="N"
#  ---------------------------------------------
#  Set priority level for the current workflow
#  ---------------------------------------------

STDWFS.WFSINPUT.WFItemPriority="5"
#  ---------------------------------------------
#  Let us do a check point here, since we want to save the accepted parameters
#  so that in case of an abort we restart from step 2
#  ---------------------------------------------
sv_a = urhk_WFS_CheckPoint("2")

STEP2:
#  ------------------------------------------------
#  If Existing Customer then skip customer creation
#  ------------------------------------------------

if (LTRIM(STDWFS.GLBDATA.custId) != "") then
   STDWFS.GLBDATA.oldCust="Y"
   GOTO STEP4
endif

#  -------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Customer creation form, he has a chance to
#  get back to that step before proceeding with the work flow
#  -------------------------------------------------

STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
print(STDWFS.GLBDATA.RetryFlg)

#  ---------------------------------------------
#  Populate fields for Customer Master Creation
```

```
#  We do not need to specify the repository name because INTBAF is assumed
#  as default.
#  ---------------------------------------------

sv_a = urhk_TBAF_SetValue("baff0009.funcblk.func_code|A")

#  ---------------------------------------------------------------
#  define fields for which output values are required after commit.
#  We need to specify the repository name(OUTTBAF) INTBAF is assumed
#  as default.
#  ---------------------------------------------------------------

sv_a = urhk_TBAF_SetValue("baff0009.datablk6.cust_id| |OUTTBAF")
sv_a = urhk_TBAF_SetValue("baff0009.datablk1.cust_introd_name| |OUTTBAF")
sv_a = urhk_TBAF_SetValue("baff0009.datablk1.cust_introd_cust_id| |OUTTBAF")

#  ---------------------------------------------------
#  Set key Accept to be done on entry into funcblk
#  Set key Exit to be done on entry into funcblk again
#  ---------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f1")

#  ---------------------------------------------
#  Call the Customer Master Creation Menu Option
#  ---------------------------------------------
sv_a = urhk_WFS_CallMenuOption("CUMM|2|3")
if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Customer creation"
    Call("skippedstep.scr")
   if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP3:

#  ---------------------------------------------------------------------------
#  Copy all the output data from the previous menu option into GLBDATA and
#  delete class INTBAF.baff0009 and OUTTBAF.baff0009
#  ---------------------------------------------------------------------------

sv_d = CLASSEXISTS("INTBAF","baff0009")
if (sv_d == 1) then
   DELETECLASS("INTBAF","baff0009")
endif

sv_d = CLASSEXISTS("OUTTBAF","baff0009")
if (sv_d == 1) then
    sv_a = urhk_TBAF_GetValue("baff0009.datablk6.cust_id")
   STDWFS.GLBDATA.custId = B2KTEMP.TEMPSTD.TBAFRESULT
   sv_a = urhk_TBAF_GetValue("baff0009.datablk1.cust_introd_name")
   STDWFS.GLBDATA.introd_custname = B2KTEMP.TEMPSTD.TBAFRESULT
   sv_a = urhk_TBAF_GetValue("baff0009.datablk1.cust_introd_cust_id")
   STDWFS.GLBDATA.introd_cust_id = B2KTEMP.TEMPSTD.TBAFRESULT
   DELETECLASS("OUTTBAF","baff0009")
endif

# ------------------------------------
# Queue up an entry for back office personnel to fill in all the details
# of the customer. Assume a workgroup called "BCKOF" exists
```

```
# Copy all the required fields into TEMPWFS since that is what is queued up
# and not STDWFS
# ------------------
sv_d = CLASSEXISTS("TEMPWFS", "GLBDATA")
if (sv_d == 0) then
   CREATECLASS("TEMPWFS", "GLBDATA", 5)
endif
TEMPWFS.WFSOUTPUT.WorkFlowDesc="Customer creation - " + STDWFS.GLBDATA.custId
TEMPWFS.GLBDATA.custId=STDWFS.GLBDATA.custId
sv_a = urhk_WFS_CreateWFItem("15|TEST1|BCKOF")

#  -----
#  Let us check point here so that in case the next step aborts for some
#  reason we will not queue up the above item again
#  ---------

sv_a = urhk_WFS_CheckPoint("18")

STEP18:
#  ------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Customer verification form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ------------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  -----------------------------------------------
#  Populate fields for Customer Master Verification
#  -----------------------------------------------

sv_a = urhk_TBAF_SetValue("baff0009.funcblk.func_code|V")
sv_v = "baff0009.funcblk.cust_id|" + STDWFS.GLBDATA.custId
sv_a = urhk_TBAF_SetValue(sv_v)

#  ----------------------------------------------------
#  Set key Accept to be done on entry into funcblk
#  Set key nxt-fields to visit all the pages
#  Set key commit to be done on entry into decisionblk
#  Set key Exit to be done on entry into funcblk again
#  ----------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("baff0009.decision_blk.key-commit")
sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f1")

#  ------------------------------------------------
#  Call the Customer Master Verification Menu Option
#  ------------------------------------------------

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("CUMM|18|4")
```

```
if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Customer verification"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP4:
#  ------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Account opening form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ------------------------------------------------

STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  ------------------------------------------------
#  Populate fields for SB/CA account opening
#  ------------------------------------------------

sv_v = "bafe3013.funcblk.cust_id|" + STDWFS.GLBDATA.custId
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3013.funcblk.gl_sub_head_code|" + STDWFS.GLBDATA.glSubHead
sv_a = urhk_TBAF_SetValue(sv_v)
sv_v = "bafe3013.funcblk.schm_code|" + STDWFS.GLBDATA.schemeCode
sv_a = urhk_TBAF_SetValue(sv_v)

# ---------------------------------------------------------------
# define fields for whichoutput values are required after commit.
# ---------------------------------------------------------------

sv_a = urhk_TBAF_SetValue("sbfe3201.lastblk.acct_num| |OUTTBAF")
sv_a = urhk_TBAF_SetValue("sbfe3201.funcblk.cust_name_1| |OUTTBAF")

#  ----------------------------------------------------
#  Set key Accept to be done on entry into funcblk
#  Set key Accept to be done on entry into datablk
#  ----------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("sbfe3201.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("sbfe3201.datablk1.key-f2")

#  ----------------------------------------
#  Call the SB/CA Account Opening Menu Option
#  ----------------------------------------

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("OAAC|4|5")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Account opening"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do
STEP5:
```

```
#  ---------------------------------------------------------------------------
#  Copy all the output data from the previous menu into GLBDATA and delete class
#  ---------------------------------------------------------------------------

sv_d = CLASSEXISTS("INTBAF","bafe3013")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3013")
endif

sv_d = CLASSEXISTS("INTBAF","bafe3201")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3201")
endif

sv_d = CLASSEXISTS("INTBAF","sbfe3201")
if (sv_d == 1) then
   DELETECLASS("INTBAF","sbfe3201")
endif

sv_d = CLASSEXISTS("OUTTBAF","sbfe3201")
if (sv_d == 1) then
   sv_a = urhk_TBAF_GetValue("sbfe3201.lastblk.acct_num")
   STDWFS.GLBDATA.tmpAcctNum = B2KTEMP.TEMPSTD.TBAFRESULT
   sv_a = urhk_TBAF_GetValue("sbfe3201.funcblk.cust_name_1")
    STDWFS.GLBDATA.custName = B2KTEMP.TEMPSTD.TBAFRESULT
   STDWFS.WFSOUTPUT.WorkFlowDesc = STDWFS.WFSOUTPUT.WorkFlowDesc +
B2KTEMP.TEMPSTD.TBAFRESULT
   DELETECLASS("OUTTBAF","sbfe3201")
endif


#  -------------------------------------------
#  Copy current User Id as A/c Opening User Id so that the workflow item can
#  be transferred back from the teller to this user for further processing
#  -------------------------------------------
STDWFS.GLBDATA.acctOpenUser=STDWFS.WFSINPUT.CurrentUserId
# ---------------------------------------------
# We assume that all tellers have been assigned to workgroup "TELR"
#  ---------------------------------------------
if  (STDWFS.GLBDATA.tranType == "C") then
   sv_a = urhk_WFS_TransferUser("6||TELR")
endif
sv_a = urhk_WFS_CheckPoint("10")
GOTO STEP10

STEP6:
# ---------------------------------------------------
# Cash transaction executed by a teller
#---------------------------------------------------
#  ---------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the cash deposit form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ---------------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")

#  -------------------
#  Set part tran types
#  -------------------
STDWFS.GLBDATA.tmpAcctPtranType = "C"
STDWFS.GLBDATA.contraAcctPtranType = "D"

#  ---------------------------------------------
#  Populate fields for Financial Transaction entry
```

```
#  ----------------------------------------------

sv_v = "bafe3012.funcblk.func_code|A"
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.funcblk.tran_type|C"
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.funcblk.tran_sub_type|NR"
sv_a = urhk_TBAF_SetValue(sv_v)


# ------------------------------------------------------------
# define fields for whichoutput values are required after commit.
# ------------------------------------------------------------

sv_a = urhk_TBAF_SetValue("bafe3012.dispblk3.tran_id| |OUTTBAF")

#  -------------------------------------------------------
#  Enter first Part tran details
#  -------------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3012.funcblk.key-f2| | |SBCAOPtm.scr|1")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.datablk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-prvrec")

#  ------------------------------------------------------------
#  Move between part trans, Request posting for each part tran
#  then commit the part tran posting request and exit from TM
#  ------------------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-f2|SBCAOPtm.scr|3")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-nxtrec")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-f2")
#sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-commit")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.dispblk3.key-f1")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.funcblk.key-f1")

#  ------------------------------------------------
#  Call the Financial Transaction entry Menu Option
#  ------------------------------------------------

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("TM|6|16")
if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="First deposit for acount"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP16:
# ---------------------------------------------
# Transfer workflow back to original user starting with STEP11
#  ---------------------------------------------
sv_a = urhk_WFS_TransferUser("11|STDWFS.GLBDATA.acctOpenUser|")

STEP10:
# ---------------------------------------------------
# Transfer transaction executed by the account opener
#---------------------------------------------------
```

```
#  ------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the transfer amount form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ------------------------------------------------

STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  -------------------
#  Set part tran types
#  -------------------
STDWFS.GLBDATA.tmpAcctPtranType = "C"
STDWFS.GLBDATA.contraAcctPtranType = "D"

#  ------------------------------------------------
#  Populate fields for Financial Transaction entry
#  ------------------------------------------------

sv_v = "bafe3012.funcblk.func_code|A"
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.funcblk.tran_type|T"
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.funcblk.tran_sub_type|CI"
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.datablk.tran_amt|" + STDWFS.GLBDATA.tranAmount
sv_a = urhk_TBAF_SetValue(sv_v)

sv_v = "bafe3012.datablk.acct_num|" + STDWFS.GLBDATA.tmpAcctNum
sv_a = urhk_TBAF_SetValue(sv_v)

# ---------------------------------------------------------------
# define fields for whichoutput values are required after commit.
# ---------------------------------------------------------------

sv_a = urhk_TBAF_SetValue("bafe3012.dispblk3.tran_id| |OUTTBAF")

#  ---------------------------------------------------------
#  Enter first Part tran details
#  ---------------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3012.funcblk.key-f2| | |SBCAOPtm.scr|1")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.datablk.key-f2")

#  ---------------------------------------------------------
#  Enter details for the second part tran also
#  ---------------------------------------------------------

   sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-f2| | |SBCAOPtm.scr|2")
   sv_a = urhk_TBAF_SetReplayKey("bafe3012.datablk.key-f2")
   sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-prvrec")

#  ---------------------------------------------------------------
#  Move between part trans, Request posting for each part tran
#  then commit the part tran posting request and exit from TM
#  ---------------------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-f2|SBCAOPtm.scr|3")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-nxtrec")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.optionblk.key-commit")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.dispblk3.key-f1")
sv_a = urhk_TBAF_SetReplayKey("bafe3012.funcblk.key-f1")
```

```
#  ----------------------------------------------
#  Call the Financial Transaction entry Menu Option
#  ----------------------------------------------

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("TM|10|11")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="First deposit for acount"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP11:

#  ------------------------------------------------------------------------------
#  Copy all the output data from the previous menu into GLBDATA and delete class
#  ------------------------------------------------------------------------------

sv_d = CLASSEXISTS("INTBAF","bafe3012")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3012")
endif

sv_d = CLASSEXISTS("OUTTBAF","bafe3012")
if (sv_d == 1) then
   sv_a = urhk_TBAF_GetValue("bafe3012.dispblk3.tran_id")
   STDWFS.GLBDATA.tranId = B2KTEMP.TEMPSTD.TBAFRESULT
   DELETECLASS("OUTTBAF","bafe3012")
endif

#  ---------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Account Verification form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ---------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  ----------------------------------------------------
#  Populate fields for SB/CA Account Verification
#  ----------------------------------------------------

sv_v = "bafe3019.funcblk.dum_acct_num|" + STDWFS.GLBDATA.tmpAcctNum
sv_a = urhk_TBAF_SetValue(sv_v)

sv_a = urhk_TBAF_SetValue("bafe3019.funcblk.func_code|V")

# ---------------------------------------------------------------
# define fields for whichoutput values are required after commit.
# ---------------------------------------------------------------

sv_a = urhk_TBAF_SetValue("bafe3019.funcblk.dum_acct_num| |OUTTBAF")

sv_a = urhk_TBAF_SetReplayKey("bafe3019.funcblk.key-f2")

#  ---------------------------------------------
#  Call the Account Verification Menu Option
#  ---------------------------------------------
```

```
sv_a = 0
sv_a = urhk_WFS_CallMenuOption("OAACAU|11|12")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Account verification"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP12:

#  ---------------------------------------------------------------------------
#  Copy all the output data from the previous menu into GLBDATA and delete class
#  ---------------------------------------------------------------------------

sv_d = CLASSEXISTS("INTBAF","bafe3019")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3019")
endif

sv_d = CLASSEXISTS("OUTTBAF","bafe3019")
if (sv_d == 1) then
   sv_a = urhk_TBAF_GetValue("bafe3019.funcblk.dum_acct_num")
   STDWFS.GLBDATA.acctNum = B2KTEMP.TEMPSTD.TBAFRESULT
   DELETECLASS("OUTTBAF","bafe3019")
endif

#  ------------------------------------------------------------------
#  If check book is to be issued for the account being opened then continue
#  else skip issue and verification of cheque books
#  ------------------------------------------------------------------

if (STDWFS.GLBDATA.chqBookReqd !="Y") then
   GOTO STEP14
endif
#  ---------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Chequebook Issue form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ---------------------------------------------------

STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  ----------------------------------------
#  Populate Fields for Cheque Book Issue
#  ----------------------------------------

sv_a = urhk_TBAF_SetValue("bafe3008.funcblk.func_code|I")
sv_v = "bafe3008.funcblk.acct_num|" + STDWFS.GLBDATA.acctNum
sv_a = urhk_TBAF_SetValue(sv_v)

#  ---------------------------------------
#  Call Cheque Book Issue Menu Option
#  ---------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3008.funcblk.key-f2")
#sv_a = urhk_TBAF_SetReplayKey("bafe3008.hdrblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3008.funcblk.key-f1")

sv_a = 0
```

```
sv_a = urhk_WFS_CallMenuOption("ICHB|12|13")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Chequebook Issue"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do
STEP13:

#  ---------------------------------------------------------------------------
#  Copy all the output data from the previous menu into GLBDATA and delete class
#  ---------------------------------------------------------------------------

sv_d = CLASSEXISTS("INTBAF","bafe3008")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3008")
endif

#  ---------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Chequebook Verification form, he has a chance to
#  get back to that step before proceeding with the work flow
#  ---------------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  ---------------------------------------------------
#  Populate Fields for Cheque Book Issue Verification
#  ---------------------------------------------------

sv_a = urhk_TBAF_SetValue("bafe3008.funcblk.func_code|V")
sv_v = "bafe3008.funcblk.acct_num|" + STDWFS.GLBDATA.acctNum
sv_a = urhk_TBAF_SetValue(sv_v)

#  ---------------------------------------
#  Call Cheque Book Issue Menu Option
#  ---------------------------------------

sv_a = urhk_TBAF_SetReplayKey("bafe3008.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3008.hdrblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3008.funcblk.key-f1")

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("ICHB|13|14")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Chequebook Verification"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do
STEP14:

#  ----------------------------------------------------
#  Copy all necessary audit data into AUDIT repository
#  ----------------------------------------------------
```

```
sv_d = REPEXISTS("ADTWFS")
if (sv_d == 0) then
   CREATEREP("ADTWFS")
endif

sv_d = CLASSEXISTS("ADTWFS","CUMM")
if (sv_d == 0) then
   CREATECLASS("ADTWFS","CUMM",5)
endif

sv_d = CLASSEXISTS("ADTWFS","OAAC")
if (sv_d == 0) then
   CREATECLASS("ADTWFS","OAAC",5)
endif

sv_d = CLASSEXISTS("ADTWFS","TM")
if (sv_d == 0) then
   CREATECLASS("ADTWFS","TM",5)
endif

sv_d = CLASSEXISTS("ADTWFS","ICHB")
if (sv_d == 0) then
   CREATECLASS("ADTWFS","ICHB",5)
endif

ADTWFS.CUMM.cust_id = STDWFS.GLBDATA.custId
if( STDWFS.GLBDATA.oldCust == "Y") then
   ADTWFS.CUMM.operation = "Open New account for Customer"
else
   ADTWFS.CUMM.operation = "Add and Verify Customer"
   ADTWFS.CUMM.cust_name = STDWFS.GLBDATA.custName
endif

ADTWFS.OAAC.operation = "New A/c Creation"
ADTWFS.OAAC.tmp_acct_num = STDWFS.GLBDATA.tmpAcctNum
ADTWFS.OAAC.acct_num = STDWFS.GLBDATA.acctNum
ADTWFS.OAAC.acct_introd_id = STDWFS.GLBDATA.introd_cust_id
ADTWFS.OAAC.acct_introd_name = STDWFS.GLBDATA.introd_custname
ADTWFS.OAAC.acctOpenUser = STDWFS.GLBDATA.acctOpenUser

ADTWFS.TM.operation = "A/c Opening transaction"
ADTWFS.TM.tran_id = STDWFS.GLBDATA.tranId

if (STDWFS.GLBDATA.chqBookReqd == "Y") then
   ADTWFS.ICHB.operation = "Issue Cheque Book "
endif

#  --------------
#  End the WorkFlow
#  --------------

STDWFS.WFSINPUT.NextStep = "0"
STDWFS.WFSINPUT.WfsStatus = "D"
exitscript


STEP15:
sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f1")

STEP19:
#  ------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Customer modification form, he has a chance to
#  get back to that step before proceeding with the work flow
```

```
#  -------------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  -------------------------------------------------
#  Call Customer Master Maintenance for Modification
#  -------------------------------------------------
sv_a = urhk_TBAF_SetValue("baff0009.funcblk.func_code|M")
sv_v = "baff0009.funcblk.cust_id|" + STDWFS.GLBDATA.custId
sv_a = urhk_TBAF_SetValue(sv_v)
sv_a = urhk_WFS_CallMenuOption("CUMM|19|17")
if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Customer master modification"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do
STDWFS.WFSINPUT.PostVerificationChkReqd="N"
STDWFS.WFSINPUT.IgnoreSameUserVerifyFlg="Y"
#  -------------------------------------------------
#  The following execution lines are done in a loop so that if the user
#  accidently quits from the Customer verification form, he has a chance to
#  get back to that step before proceeding with the work flow
#  -------------------------------------------------
STDWFS.GLBDATA.RetryFlg="Y"
while (STDWFS.GLBDATA.RetryFlg == "Y")
#  ------------------------------------------------
#  Populate fields for Customer Master Verification
#  ------------------------------------------------

sv_a = urhk_TBAF_SetValue("baff0009.funcblk.func_code|V")
sv_v = "baff0009.funcblk.cust_id|" + STDWFS.GLBDATA.custId
sv_a = urhk_TBAF_SetValue(sv_v)

#  ----------------------------------------------------
#  Set key Accept to be done on entry into funcblk
#  Set key nxt-fields to visit all the pages
#  Set key commit to be done on entry into decisionblk
#  Set key Exit to be done on entry into funcblk again
#  ----------------------------------------------------

sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("bafe3023..key-f1")
sv_a = urhk_TBAF_SetReplayKey("bafe3023..key-f1")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-nxtfld")
sv_a = urhk_TBAF_SetReplayKey("baff0009.dummyblk.key-f2")
sv_a = urhk_TBAF_SetReplayKey("baff0009.decision_blk.key-commit")
sv_a = urhk_TBAF_SetReplayKey("baff0009.funcblk.key-f1")

#  -------------------------------------------------
#  Call the Customer Master Verification Menu Option
```

```
#  ------------------------------------------------

sv_a = 0
sv_a = urhk_WFS_CallMenuOption("CUMM|18|4")

if  (STDWFS.WFSINPUT.StepCommitFlg == "Y") then
    STDWFS.GLBDATA.RetryFlg="N"
else
    STDWFS.GLBDATA.Stepdesc="Customer mod verification"
    Call("skippedstep.scr")
    if  (STDWFS.WFSOUTPUT.ErrorMesg == "ExitScript") then
        exitscript
    endif
endif
do

STEP17:

sv_d = CLASSEXISTS("INTBAF","baff0009")
if (sv_d == 1) then
   DELETECLASS("INTBAF","baff0009")
endif
sv_d = CLASSEXISTS("INTBAF","bafe3023")
if (sv_d == 1) then
   DELETECLASS("INTBAF","bafe3023")
endif

STDWFS.WFSINPUT.NextStep="0"
STDWFS.WFSINPUT.WfsStatus="D"
exitscript

end-->
```

SCRIPT: SKIPPEDSTEP.SCR

```
<--start
#----------------
# This script allows the user to continue execution or abandon/suspend the
#  workflow in case he quits from a menu option inadvertently (skippedstep.scr)
#-------------------------

STDWFS.GLBDATA.ContinueFlg="N"
while (STDWFS.GLBDATA.ContinueFlg == "N")
BANCS.FRMVALUE.multiRecFlg = "N"
BANCS.FRMVALUE.FuncBlkLiteral_1 = ""
BANCS.FRMVALUE.FuncBlkLiteral_2 = ""
BANCS.FRMVALUE.FuncBlkLiteral_3 = ""
BANCS.FRMVALUE.FuncBlkValue_1 = ""
BANCS.FRMVALUE.FuncBlkValue_2 = ""
BANCS.FRMVALUE.FuncBlkValue_3 = ""
BANCS.FRMVALUE.FormTitle="Abandon/Suspend form"
BANCS.INPARAM.ErrorMesg="Use the <ABANDON/SUSPEND> key to terminate the workflow"
BANCS.FRMATTRIB.a1=" |70|P"
BANCS.FRMVALUE.a1="The previous step (" + STDWFS.GLBDATA.Stepdesc + ") "
BANCS.FRMATTRIB.a2=" |70|P"
BANCS.FRMVALUE.a2="has been skipped without <COMMIT> key having been hit."
BANCS.FRMATTRIB.a3=" |70|P"
BANCS.FRMVALUE.a3="This could be due to some system problems or accidental."
BANCS.FRMATTRIB.a4=" |70|P"
BANCS.FRMVALUE.a4="If you would like to retry the step enter 'Y'"
BANCS.FRMATTRIB.a5=" |70|P"
BANCS.FRMVALUE.a5="in the field below, else, use the <ABANDON/SUSPEND>"
BANCS.FRMATTRIB.a6=" |70|P"
BANCS.FRMVALUE.a6="key to stop the workflow."
BANCS.FRMATTRIB.a7=" |1|P"
```

```
BANCS.FRMVALUE.a7=" "
BANCS.FRMATTRIB.a8="Retry previous step?|1|N"
BANCS.FRMVALUE.a8="N"
sv_a = urhk_B2K_ShowParamAcptFrm("bafi2028")
STDWFS.GLBDATA.ContinueFlg=BANCS.FRMVALUE.a8
do
exitscript
end-->
```

SCRIPT NAME: SBCAOPTM.SCR

```
<--start
TRACE ON
#******************************************************************************
#  This is a Workflow script for form event script logic used by the workflow
#  script SBCAOP.scr(SBCAOPtm.scr)
#*****************************************************************************
sv_b = cint(INTBAF.INTBAFC.TbafEventStep)

if (sv_b == 1) then
    GOTO STEP1
endif

if (sv_b == 2) then
    GOTO STEP2
endif


if (sv_b == 3) then
    GOTO STEP3
endif

exitscript
# ----------------------------------------------
# Populate the fields of the new account part tran
#  ----------------------------------------------
STEP1:
sv_v = "datablk.acct_num|" + STDWFS.GLBDATA.tmpAcctNum
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
sv_v = "datablk.part_tran_type|" + STDWFS.GLBDATA.tmpAcctPtranType
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
sv_v = "datablk.tran_amt|" + STDWFS.GLBDATA.tranAmount
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
exitscript
# ----------------------------------------------
# Populate the fields of the transfer account part tran
#  ----------------------------------------------
STEP2:
sv_v = "datablk.acct_num|" + STDWFS.GLBDATA.acctNumber
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
sv_v = "datablk.part_tran_type|" + STDWFS.GLBDATA.contraAcctPtranType
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
sv_v = "datablk.tran_amt|" + STDWFS.GLBDATA.tranAmount
sv_a = urhk_TBAF_ChangeFieldValue(sv_v)
exitscript
# ----------------------------------------------
# Populate "P" in the option block of TM to request posting
#  ----------------------------------------------
STEP3:
sv_a = urhk_TBAF_ChangeFieldValue("optionblk.option_code|P")
exitscript
end-->
```

*****{----}*****