

TASK TRIGGERS: User Manual

SHUTTLE ENTERPRISE SERVER

Introduction

Task Desk introduces an orderly, controlled way of working. SHUTTLE will allow you to work with or without the Task Desk, but there are many benefits to the use of the Task Desk, including -

Knowing exactly what still needs to be done, and by whom
Knowing exactly what has been done, when and by whom
No chance of overlooking or forgetting Tasks to accomplish
Automated Task Flow Control, e.g. Document or Procedural Controls,
Month End Routines, etc.

This Manual is # 4 in the Task Desk series, and covers the subject of Task Triggers. The Manual should be read together with the other (preceding) Manuals in the Series.

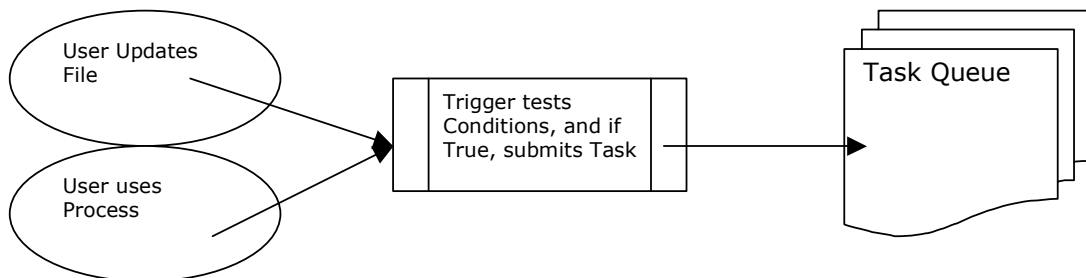
Purpose of the Manual

The purpose of this User Manual is to introduce you to the use of TASK TRIGGERS.

The Task Desk Manual is presented in 4 parts.

- a) Use of the Task Desk
- b) Use of Task Profiles
- c) Use of Task Schedules
- d) Use of Task Triggers (current Document)

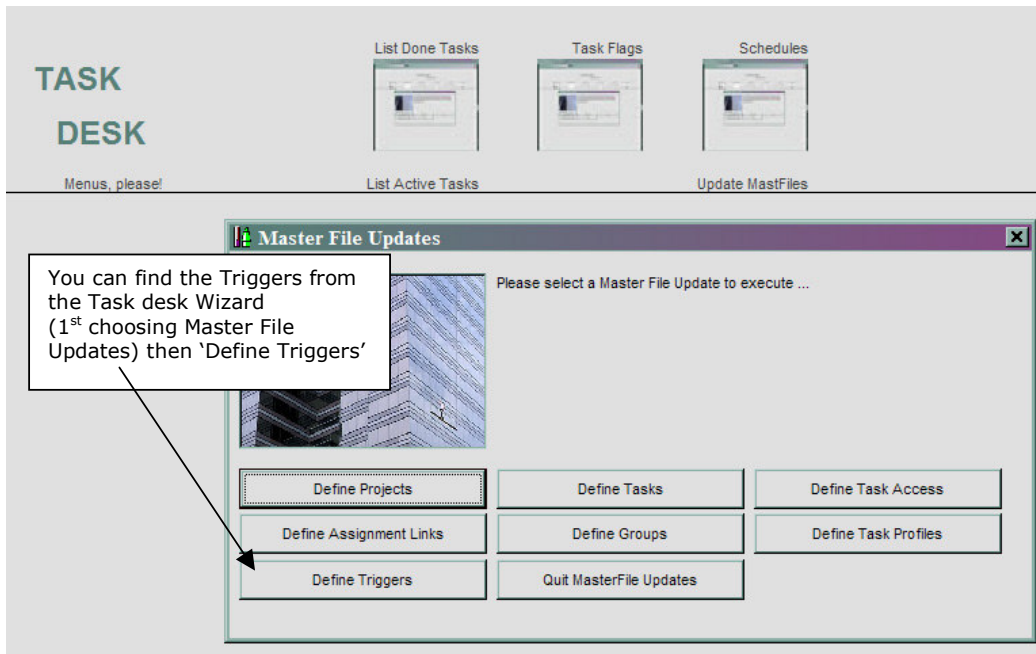
What is it? Task Triggers



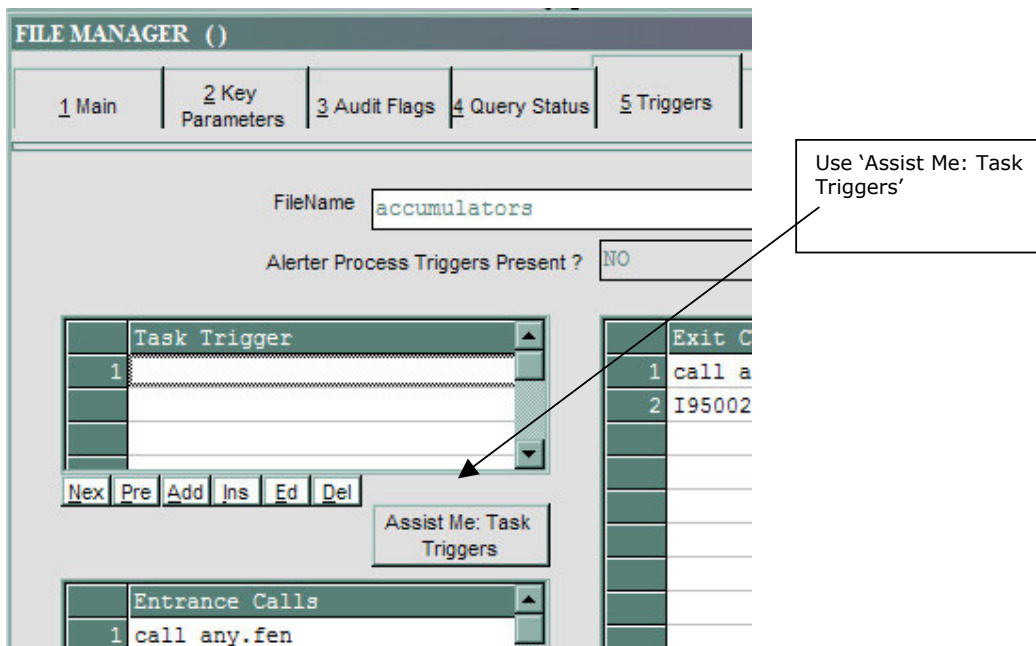
A Task Trigger is an Object that is linked to a File or Menu Process Object. When the File is Updated by a Screen (being the primary File), or the Menu Process is used, the Task Trigger is called. The Trigger tests certain pre-defined Conditions, and if these are True, it submits a Task. That is the simple logic of it.

An example of using a Task Trigger

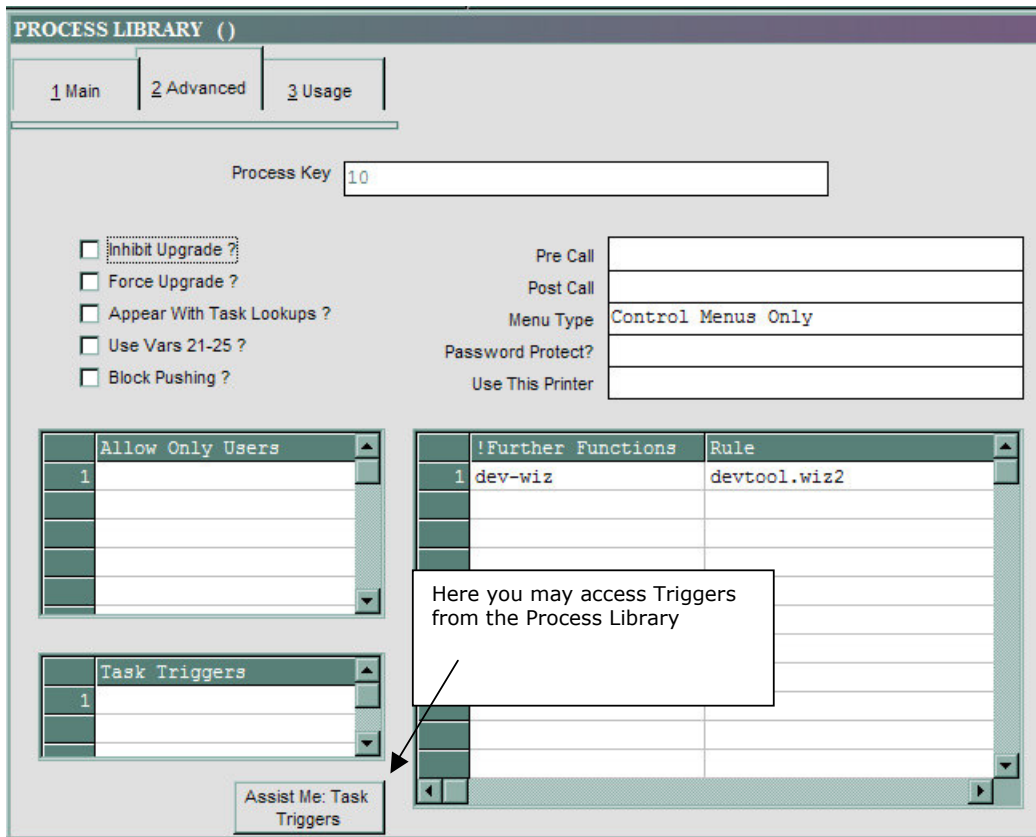
A Task Trigger can be defined or re-defined from the options provided at Master File Updates, on the Task Desk Wizard (or accessed from the Menu), OR such a Trigger can be accessed directly from the File Manager or Menu Object where it is linked or to be linked.



Or from File Manager -

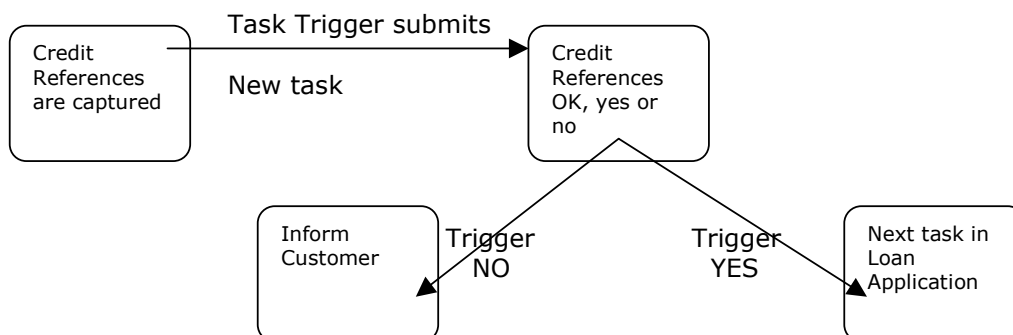


Or from the Menu Object -



In a moment we will have a look at what a Task Trigger definition looks like. 1st let's consider the following fictitious example in Company ABC, a Finance Company. They get new Loan Applications every day. The 1st Step is that the Customer fills in a Loan Application Form. Based on this information, which includes Credit References, an Operator captures the Credit references into the system. This step triggers a Task to the next Section in the Application cycle, i.e. the people who follow up the Credit references as a 1st step in verifying credit worthiness of the Customer. The new Task must be completed by indicating, in the system, 'Yes' or 'No' to proceed with the next step of the Loan Application. Another set of Triggers then kicks in: 1) If the answer is 'yes', a Task is submitted to Proceed with the next step in the Loan Application, 2) If the answer is 'no', then a Task is submitted to someone else to inform the Customer that the Loan Application was not successful.

Clearly, there can be many more steps in this process, but this is sufficient for us to demonstrate how Task Triggers can be applied to accomplish the above.



We will now proceed to look at a way to define this control flow, but please note that there are multiple ways this can be done, not just the one way we will show in this example.

Consider the following Screen :-

The screenshot shows a window titled "NEW LOAN APPLICATION ()". Inside the window, there are several data entry fields:

- Application #**: A text box containing the value "2553".
- Family Name**: A text box containing the value "van der Hough".
- Given Names**: A list box containing two entries: "Trevor" and "Gavin".
- Credit References**: A list box containing two entries: "Walmart, Account # 12-4432-121" and "ABC Telephony, Account # 55732321".

At the bottom right of the window, there are two buttons: "Exit" and "Submit".

Not necessarily complete, but sufficient to give the impression that we are capturing New Loan Application data for a Customer, including Trade References.

When SUBMIT is selected, the data is written to file. The primary file in this case is called 'creditappli', and we can choose whether to install a Task Trigger on the File itself, or on the Menu Process that is used to capture this information.

Although in each case we may have multiple Task Triggers, it is usually easier to separate the Triggers when attaching them to the Menu Process, especially if the same File is going to be used again and again for different Screens where we wish to apply Task Triggers. The benefit to attaching Triggers to the File itself is that they will be applied for any and all Screens that we may define with this File as the primary File to update. Certainly there are scenarios where that would be the better way. In our example, we will attach Triggers to the different Menu Processes that we use.

The Screen above uses the Menu Process 'creditappli', and that is where we will define the Task Trigger that submits a Task for 'someone' who will follow up and verify the Trade References.

DEFINE TASK TRIGGERS ()

Trigger Key:

*Description:

Descr To Derive:

Type:

Active

Auto Sign-Off ?

Conditions	
1	a0 = a0

Condition Guide:

Profile(S)	
1	New Loan Appli: Verify Credit

Parameters To Pass On	
1	a0

Notes: This Trigger submits a Task whenever a new Loan Application is filed. The Task is assigned to those that Perform Trade Refs Verification.

Buttons: Give me a Key, Assist Me, Remove Me, Linked Objects, Quit, Update

The above pic is our Task Trigger definition. We will now focus on different sections of the Screen to see it more clearly.

Trigger Key:

*Description:

Descr To Derive:

Type:

Active

Auto Sign-Off ?

Conditions	
1	a0 = a0

Condition Guide:

Note 1: Points to Descr To Derive field

Note 2: Points to Type field

Note 3: Points to Active checkbox

Note 4: Points to Auto Sign-Off checkbox

Note 5: Points to the first condition row in the Conditions table

Note 6: Points to Condition Guide field

Callout text: You can devise the Key or ask the system to do it, and the Trigger needs a Description

Note (1) Descr to Derive: It will not be meaningful to see a 100 Tasks in your Queue that all say "New Loan Application". This Field may be parameterized to build a unique Description for each Task that is submitted. In this particular case, we build a Description that includes the Loan Application Number and the text "Verify Trade Refs".

Note (2) Type: There are 't'ask and 's'ign off types. The 1st submits a Task, the 2nd Signs if Off when all conditions have been met, i.e. when the Task has been completed successfully.

Note (3) Active: Before a Triggers performs it's work, it checks whether it is supposed to be active. If not, it simply terminates.

Note (4) Auto Sign Off: If checked, this signals to the system that there is an automated Trigger in place (type = Sign Off) that will decide when the Task is complete, and then the User cannot Sign it Off, only the system.

Note (5) Conditions: In our current example, we want a Task submitted each time, and no conditions are set. However, since at least 1 Condition is required, we simply set a Condition that will always be true, i.e. Record Key = Record Key.

Note (6) Condition Guide: You can tell the system which and / or how many of the Conditions listed must be satisfied for the Trigger to submit a Task.

All these Fields have further Help on-line, and you can also use the ASSIST ME wizard to formulate the contents for the various Fields on the Trigger.

The Profiles determines the content of the Task, and who will perform it.

Profile (S)	
1	New Loan Appli: Verify Credit

Parameters To Pass On

Parameters To Pass On	
1	a0

We can pass any required Parameters to the Task. In this case, we pass only the Record Key, i.e. Loan Application Number.

Notes

This Trigger submits a Task whenever a new Loan Application is filed. The Task is assigned to those that Perform Trade Refs Verification.

Notes on the Trigger are useful to explain what the Trigger is about, and to explain anything that may not be obvious when we visit this Trigger Definition again in future.

UPDATE and QUIT need no explanation, and the other Buttons are listed below.

Give me a Key Assist Me Remove Me

Linked Objects Quit Update

GIVE ME A KEY will produce a Unique Key for the Trigger if you do not wish to define the Key yourself.

ASSIST ME is a Wizard that will help you define any or all Fields on the Trigger.

REMOVE ME will remove the Trigger from the system, as long as it is not in use on a File or Menu Process.

LINKED OBJECTS will display the Files and / or Menu Processes where the Trigger is currently applied. (A Trigger may be applied in more than 1 place.)

In the Trigger that we have shown above, we used a 'Profile' that will determine the Task Content, and who will perform this. Although Profiles have already been discussed in a prior Manual, it will be useful to have a look at what this Profile does –

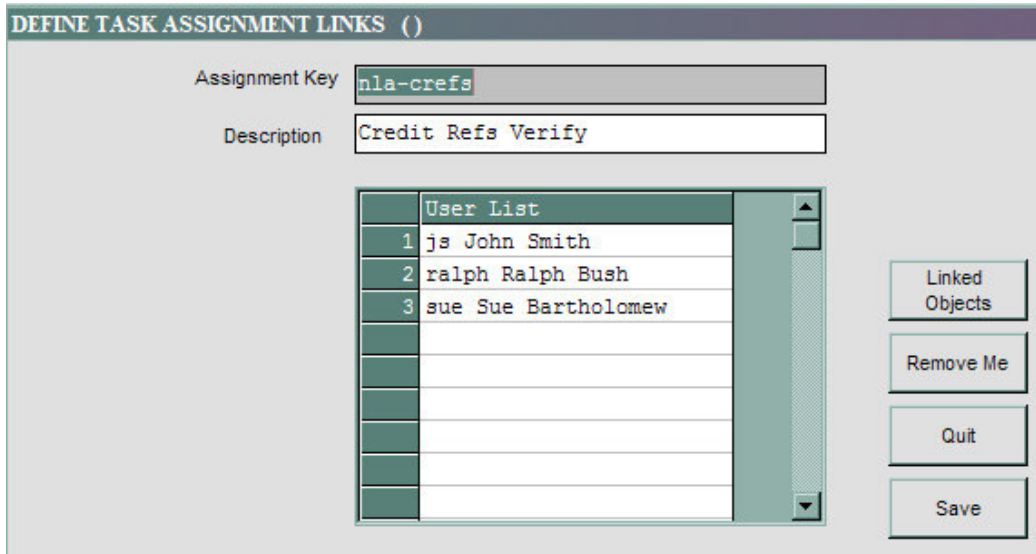
Menu Links	
1	creditappli-2 Verify Trade Refs

Note (1) Assign To: This Profile 'connects' to an Assignment Link that lists the User(s) who may perform this Task. Below, we will look at this Assignment Link.

Note(2) Project: The Profile is part of our 'New Loan Applications' Project, therefore all the Tasks can be easily found to check Progress, what's been done, what remains, etc.

Note (3) Menu Links: The Menu Link Specified here will be included on each Task submitted by the Trigger which uses this Profile. Since we also pass the Loan Number from the Trigger as a Parameter (have a look at the Trigger again, higher up), that means when the User opens the Task from his / her Queue, the system has enough information to automatically open the correct Menu Process (Screen), and the correct Record (e.g. Loan Number).

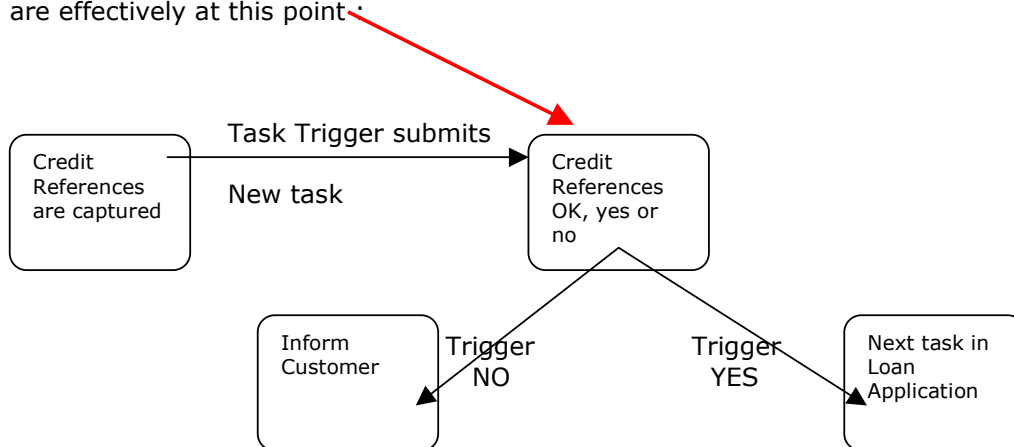
Here's what the Assignment Link on the above Profile looks like –



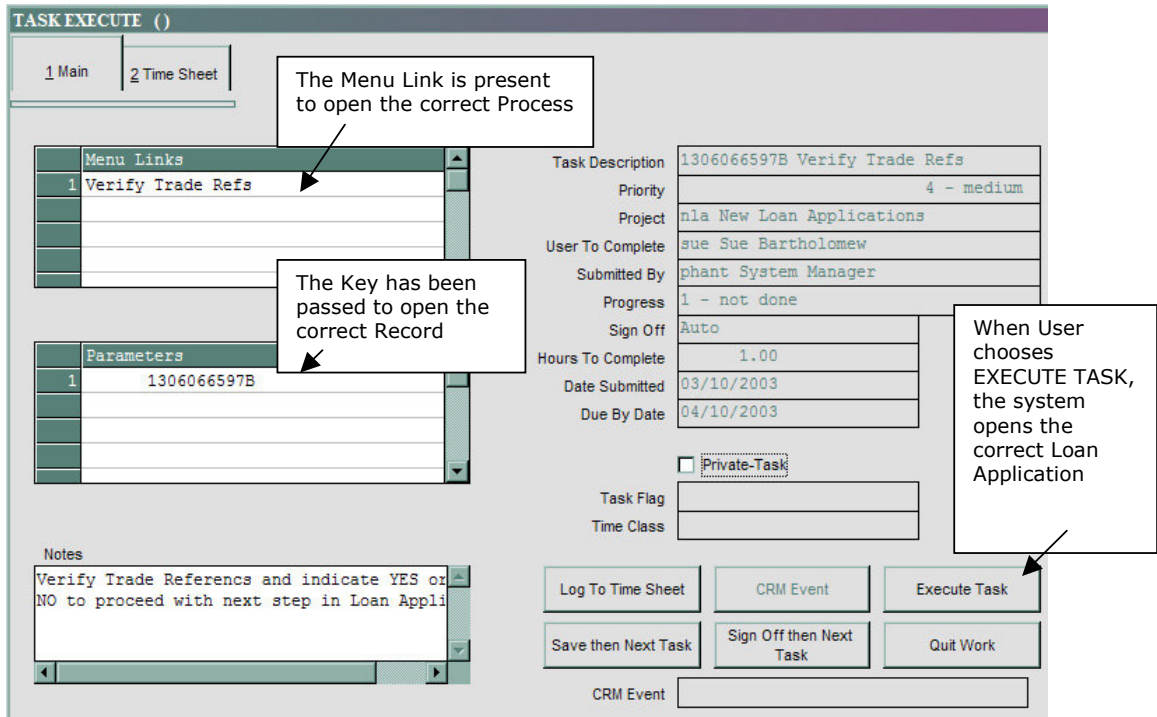
Assignment Links have a single purpose in life, to serve 1 or more Profiles that serve 1 or more Task Triggers. An Assignment link is like a Task Function (when we discussed Task Schedules in the previous Manual), only it has the capacity to indicate more than 1 User. If more than 1 User is listed, then the Task Scheduler will automatically give each person an equal amount of work, i.e. 1st Loan to John, then to Ralph, then to Sue, then to John again ...

So, from the above we can begin to see the power of Task Triggers, even if we may not yet understand all of it. The point is, a Task Trigger can automatically Schedule work, and each step of an otherwise manual set of procedures can actually be automated (and more effectively controlled and documented) with Task Triggers.

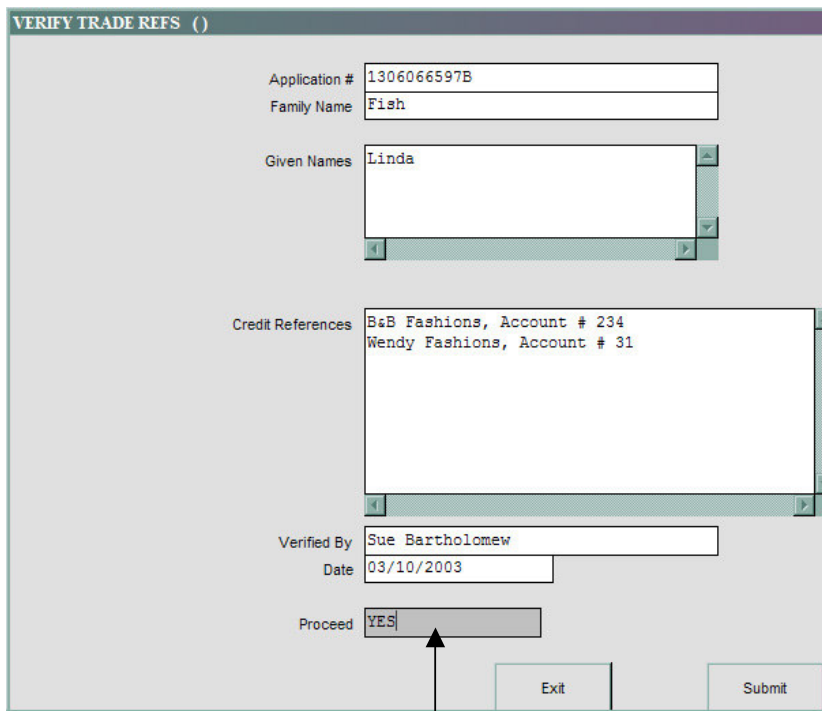
Below we show again the Task Flow that we showed before. We had a look at the Screen that a User may capture with a Menu Process to which is attached a Task Trigger, and we have had a look at the Trigger. When that Trigger submits a Task, it lands in some User's Queue, and when that User opens his / her Task, we are effectively at this point:



Now let us see what the Task looks like -



And the system opens the correct Screen with the Loan Application to be Verified



For our control flow, we want to introduce 2 Task Triggers with the Menu Process shown above. If YES, we submit 1 type of Task, if NO, we submit a different Task. Also, we attach a 'Sign Off' type trigger on this Process. Once the Operator finishes this Task by indicating YES or NO, then the original Task is complete.

This is what it looks like –

The screenshot shows the 'DEFINE TASK TRIGGERS' window with the following configuration:

- Trigger Key: nla-yes
- Description: Loan App Verify: YES/NO
- Descr To Derive: "Loan Proceed =":a7
- Type: t.task
- Active:
- Auto Sign-Off?:
- Conditions:
 - 1 a7 = "YES"
 - 2 a7 = "NO"
- Condition Guide: 1 Condition 1 Profile
- Profile(S):
 - 1 Loan App Verify: YES
 - 2 Loan App Verify: NO
- Parameters To Pass On:
 - 1 a0

Buttons: Give me a Key, Assist Me, Remove Me, Linked Objects, Quit, Update.

Notes: This Trigger submits a Task whenever a new Loan Application has Trade refs verified and PROCEED = YES. If NO, it submits a different Task.

This Trigger is of type = `task. It tests 2 Conditions, and whichever is True, calls the corresponding Profile to submit a Task. Hint: If you have Number of Conditions = Number of Profiles, you may choose a Condition Guide = 1 Condition 1 Profile. This means the system will test each condition, and the 1st one that is True, it will use the corresponding Profile to submit the next Task. Therefore, if Condition 2 is the 1st to be True, it uses the 2nd Profile specified.

In our Trigger, we first test for PROCEED = YES. If this is True, we use Profile = Loan App Verify: YES, which submits a Task accordingly, and if PROCEED = NO, then we use the 2nd Profile, to submit a Task for that situation.

To be sure, we could have used 2 separate Triggers to do the same, but why should we if it is not necessary?

We also need a Trigger of type `signoff, to signal whether the original Task is now complete, and the only condition we need is that PROCEED (a7, e.g. Field/Attribute 7) be not equal to NULL. The Trigger looks like this –

Trigger Key

*Description

Descr To Derive

Type

Active

Auto Sign-Off ?

Conditions	
1	a7 # ""

Condition Guide

Profile(S)	
1	

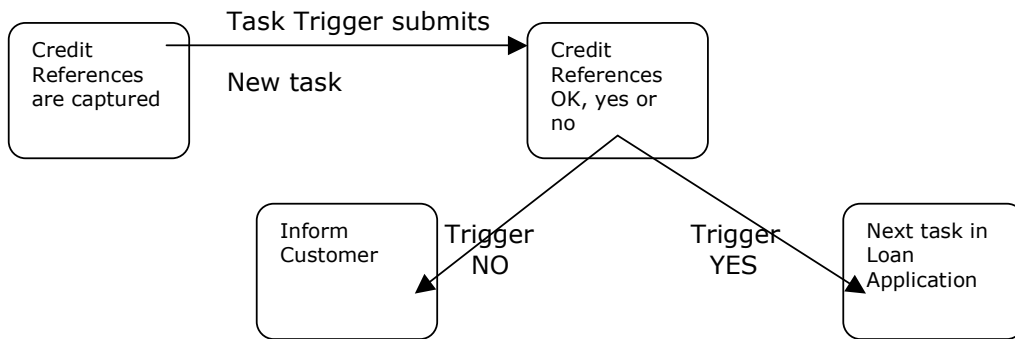
Parameters To Pass On	
1	* ignore if 'sign-off' trigger

If PROCEED # NULL (Attribute 7), then the Loan has been verified, and the original Task is complete, so the system will sign it off automatically.

And we specify both of these Triggers with the Menu Process –

Task Triggers	
1	nla-signoff Sign OFF: N
2	nla-yes Loan App Verify

This is part of the Menu Process definition for the Process that opens the original Task that was submitted, according to our Control Flow



We have done everything up to here -



And this is an example of how Task Triggers can be employed to control Task Flow.

© Infolab, 2003.

This Documentation is copyrighted by Infolab (Pty) Ltd. [www.infolab.cc] All rights are reserved. Licensed Shuttle™ Users are granted permission, for internal use ONLY, to reproduce the Documentation, and to include amendments dealing with specific instructions local to your installation.