

In This Chapter

Definitions

The Observer Process™

Create a Filter/Group Historical Activity

Triggers

Actions

Tid Bits

Wrap Up

Definitions

Note

As of GoldMine Premium 8.5, you may notice some terminology changes. For instance: You can no longer **Remove a Track**, but, instead, you can now **Remove a Process**.

Tracks have now been rebranded as **Processes**.

Automated Processes is a tough chapter to write as a single chapter. I previously wrote an entire book, **Sales Force Automation with GoldMine 4.0**, as a blueprint guide to working with Automated Processes. Pretty much everything that I said in there then, still holds true for GoldMine Premium.

I still maintain that GoldMine is not going to tell you, or suggest to you, how to run your business nor are any of the GoldMine Partners, including myself. I further state, that one must have a process before one may begin to automate that process within GoldMine. A process should consist of more than a single task that you want to automate. A process should be a series of tasks that a user performs within GoldMine on a regular basis against any number of contact records.

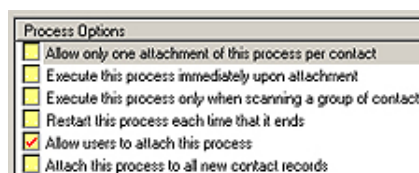
In **Sales Force Automation with GoldMine 4.0**, still available as an eBook from **DJ@DJHunt.US**, I suggested that you create one Automated Process to watch over all the activity within GoldMine, and that Automated Process would assign the appropriate other Processes based upon a particular GoldMine activity. This particular process would be automatically attached to every **New** record created within GoldMine. The default Automated Process is identified in GoldMine through the corporate level GM.ini. A typical GM.ini statement identifying the Automated Process to be attached to all New records would be something like:

[GoldMine]
NewContactAP=9KUMSUA*XZ/XR#Y

Many readers of my past books have asked me to expand this chapter to include more examples. I plan to do that in this book again, however, I want to begin with a rehash of my former books.

Automated Processes - a collection of **Processes** that could be scanned, and that are created/grouped by **UserID**.

Processes - a series of steps, **Events**, that are to be taken while the process is active. Each Process can be assigned certain attributes as shown here in Figure 10-1.



Of these options, which are all self explanatory, the last option is important. The last option may only be used once per GoldMine installation. As you read above, this option is written directly into the GM.ini file. A user will be warned, when selecting this option, if any other Process has already been defined for this GoldMine installation. I will suggest, later, that you select this option for the **Observer Process™** that we will be discussing in this chapter. This is the same exact paradigm that I employed when I wrote **Sales Force Automation with GoldMine 4.0**.

Event - the individual components of a **Process**, the ties of the processes, so to speak. Each **Event** possesses a **Description**, **Trigger**, and an appropriate **Action**.

Preemptive - Events that are analyzed each and every time that a **Process** is scanned, and that are always processed before any Sequential Events are even considered. This is an important concept to grasp. As you will see later in our Observer Process, I will make extensive use of Preemptive Events. Preemptive Events must have a Sequence Number between **10** and **99**.

Sequential - Events that are analyzed when a **Process** is scanned, however, each **Event** designated as a Sequential Event, may not be analyzed until the previous Sequential Event has been Triggered, and the associated Action has been executed. Sequential Events must have a Sequence Number between **100** and **999**.

The Observer Process™

Note

Processes are automatically removed from the Processes tab by GoldMine when the last **Sequential Event** has been successfully executed.

Note

Process Codes are used by GoldMine to create a display order for the **Processes** defined by a given user. This is an alpha ordering of the **Processes** in the **Automated Process Center**.

Note

Those of you who will be Synchronizing your Automated Processes, and want your Remote-side to attach the Observer Process to all newly created records, will have to append the Remote-side GM.ini with the same statement that appears in the Server-side GM.ini. This statement, in my case, looks like:

[GoldMine]
NewContactAP=BOHRNWR\$Y2UR#Y

Note

You may want to consider the ☐ **Execute this process immediately upon attachment** option if you are attaching the Observer Process via the WebImport as discussed in a previous chapter.

Trigger - an activity that evaluates to a **True** causing the **Action** to be processed, or to **False** where the **Action** will not be processed.

Action - the activity that is performed once the **Trigger** has been evaluated, and when the evaluated **Trigger** returns a **True**.

I have long advocated the use of, and what I've trademarked as the **Observer Process™**. This Process will automatically be attached to each newly created contact record by GoldMine, and will be present during each scanning of the Automated Processes to test for activity against the Contact record to which this process has been attached. If one of the Events on the Observer Process Triggers, it should do nothing more than assign a new Process appropriate to the cause of the triggering.

The Observer Process will be made up of nothing more than Preemptive Events with one Sequential Event to hold the Process to the record so that it will never be removed from the record unless removed by an individual. Let's look at the creation of this Process.

I begin in the **Automated Process Center** as any user. In my case I selected the (**public**) UserID, however, I could just as easily used **DJ** or **MASTER**. I then clicked upon:

Tools

[Automated Processes ►](#)
[Manage Processes...](#)



Figure 10-2

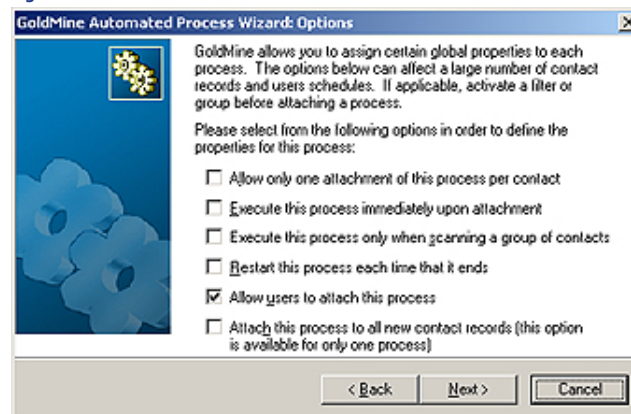


Figure 10-3

contact, select this option so the events will only be checked upon full contact set scans. I want to allow users to scan this process for single contacts, and have, therefore, not selected this option.

In case my **Hold Process**, discussed later, fails for any reason, I would want GoldMine to reattach this process to the Contact record. I have, therefore, selected to ☒ **Restart this process each time that it ends**. Further, I have opted to ☒ **Allow users to attach this process**, as there may be a need to do so, because of some previous failure(s). I want my users, in this case, to have the ability to assist me in my GoldMine maintenance tasks. Lastly, I have selected the option that may only be selected once per GoldMine installation, which is to ☒ **Attach this process to all new contact records (this option is available for only one process)**. Clicking on the **Next >** button brings us to the dialog form shown in Figure 10-04 on the next page.

This will result in the dialog form as shown here in Figure 10-2. The **Process Name**: that I have established for this exercise is that of the **Observer Process**, while the **Process Code**: I set to **OBS**. The **Owner**: field is, in this instance, assigned to the (**public**) UserID. Clicking the **Next >** button brings up the dialog form shown in Figure 10-3.

As this is the Observer Process that will be attached to every Contact record in GoldMine, I do not want to allow more than one instance of this particular Process per Contact record. I will select to ☒ **Allow only one attachment of this process per contact**. Whereas, I do not want to ☐ **Execute this process immediately upon attachment**, so I will not select that option.

As well, I will not select to ☐ **Execute this process only when scanning a group of contacts**. This particular terminology is a bit confusing to some users. If you do not want the users to have the ability to scan this process for a single

Note

I add a **Hold** Event to the Observer Process so that the Observer Process will never detach itself automatically from the record to which it is attached. Naturally, you may always detach the Observer Process by hand.

Note

As you are using the Builder, **AccountNo is Empty** will be all that you need to enter. I only supply the **Syntax**: here for your edification.

I would ask you to click the **New** button here, and create the **Hold Process to Record** Event. I do this so that it will be available from the tree for creating new Events.

Event: Hold Process to Record
Type: Sequential
Sequence: 999

Trigger: dBASE Condition

Options:

English: AccountNo is Empty

Syntax: trim(contact1->accountno) < ' ' .or. "SQL_WHERE"="(UPPER(c1.ACCOUNTNO) = " OR UPPER(c1.ACCOUNTNO) <= ' ' OR UPPER(c1.ACCOUNTNO) IS NULL)"

Attempt to trigger only once: ☐

Trigger Filter: ☐

Filter: None

Perform Action: Schedule Activity

Options: I usually schedule an Alarmed Other Action notifying DJ that the Observer Process was inadvertently removed.

When through, you can click on the **OK** button to be returned to the dialog form as shown here in Figure 10-4.

I will just click on the **Next >** button when I get back here as I will be creating the rest of the Events for the Observer Process later on in this chapter. That action brings us to the next dialog form from which I can do little more than click on the **Finish** button.

This will then create a tree similar to that shown, expanded, in Figure 10-5.

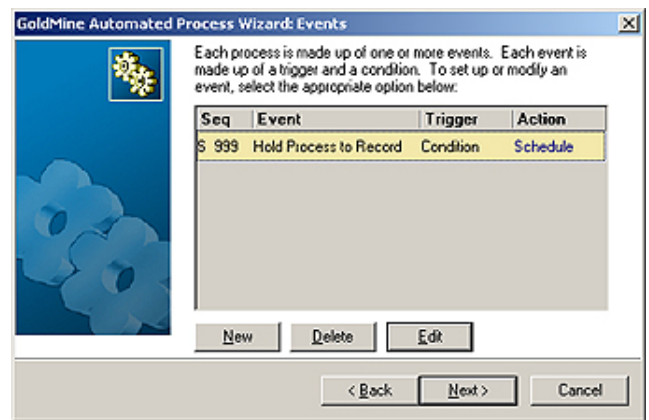


Figure 10-4

Note

It is important that you create your Processes prior to assigning them to the Observer Process for later dissemination.

From here I can continue on with my Events. Let's understand first what I will try to do here in my first Process. I am going to create a Birthday Process. This process will perform the appropriate actions for the Contact record.

For this to work as I would like, I will require the creation of a user defined field:

Field Name: uDOB
Description: Birthday
Field Type: Date, 8

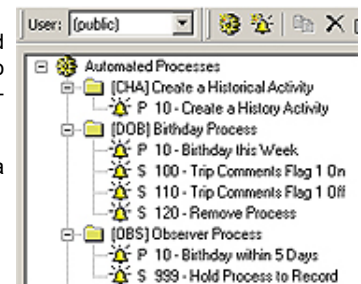


Figure 10-5

I would ask you to add this field to your GoldMine if you are to follow along on this example.

I can now continue on, and define our **Birthday Process**. Once I have tested the Birthday Process, I can then add a Preemptive Event to our Observer Process that will watch for certain conditions, and when Triggered, will attach our Birthday Process. If this sounds too simple, don't be tricked into believing it. I have had the Automated Process logic laid out for years, and I'm writing this as a Step-by-Step match exactly as I have created these many times in the past. But let's continue on.

Let's now add the Birthday Process. You remember how to do this from my directions for adding the Observer Process, do you not? Create your own Birthday Process now with one bogus Preemptive Event (we will modify that later to meet the specific needs for this Process). Here is the Process criterion:

Process Name: Birthday Process

Process Code: DOB

Owner: (public)

Options:

- ☒ Allow only one attachment of this process per contact
- ☒ Execute this process immediately upon attachment
- ☐ Execute this process only when scanning a group of contacts
- ☐ Restart this process each time that it ends
- ☒ Allow users to attach this process
- ☐ Attach this process to all new contact records (this option is available for only one process)

Trick

You can create any dBase expression that you want to in the AP's if you follow these simple procedures:

A. Create any expression using the **Expression Builder** shown later in Figure 10-8.

B. Click the **OK** button.

C. Click the **Options** button, shown later in Figure 10-7, to bring up the **Expression Builder**, Figure 10-8.

D. Highlight everything in the expression box, and begin typing your own expression by hand. **Do not** use the **Expression Builder** this time.

E. Click the **OK** button when finished.

If you have typed a valid expression, even if it is not against the **Contact1/Contact2** tables, which is all that the **Expression Builder** allows, your expression will be saved, and utilized.

Tip

Notice that I have left sufficient space between my sequencing in case I later find out that I have to add more Events in between these Events. This will save a lot of reshuffling later on.

Note

You may want to add a Branch to Event such that if this Event doesn't fire, no E-mail Address, then to print a birthday document instead for postal mailing.

Note

You may have noticed that I prefer to utilize the old dBase [*an*] braces where possible. I find that the ' , when doubled, can be confused with the " when using formatted text as I do in this book.

Note

Pay particular attention to my usage of a **Flag** byte to prevent the multiple firing of events on subsequent scans of the AP's. You may find this to be a most useful trick in your development of your AP's.

When finished, select the bogus Preemptive Event from this tree. Now, right-click, and select **Properties...** from the local menu. Edit this Preemptive Event to be the same as the first Event that is shown below, and then right-click on that Event again, in the tree, only this time selecting **New...** from the local menu. Do this for the final two Events for this Process.

Here is the list of Events that should be created for your Birthday Process:

Event:	Birthday this Week	
Type:	Sequential	
Sequence:	100	
Trigger:	Detail Record	
Options:	Detail: E-mail Address	
Trigger Filter:	<input checked="" type="checkbox"/>	
Filter:	substr(ContSupp->Zip, 2, 1) = [1] .and. empty(left(Contact2->Comments, 1))	
Action:	E-mail message	
Options:	Select your predefined Birthday E-mail Template	
Event:	Trip Comments Flag 1 On	
Type:	Sequential	
Sequence:	105	
Trigger:	Immediate	
Options:		
Trigger Filter:	<input type="checkbox"/>	
Filter:		
Action:	Update Field	
Options:	Field: Comments	Expression: [1]+right(Contact2->Comments, 64)
Event:	Trip Comments Flag 1 Off	
Type:	Sequential	
Sequence:	110	
Trigger:	Elapsed Days	
Options:	6	
Trigger Filter:	<input type="checkbox"/>	
Filter:		
Action:	Update Field	
Options:	Field: Comments	Expression: []+right(Contact2->Comments, 64)
Event:	Remove this Process	
Type:	Sequential	
Sequence:	115	
Trigger:	Immediate	
Options:		
Trigger Filter:	<input type="checkbox"/>	
Filter:		
Action:	Remove Process	
Options:	Field:	Expression:

And here is the verbose explanation of what we have just create in the Birthday Process.

I already know that the birthday criterion was met, or this Process would not have been added to the Contact record in the first place by the Observer Process. I'll discuss that later when I ask you to attach the Event to the Observer Process that will attach this, the Birthday Process. Further testing would be redundant. I first test for the existence of an E-mail Address, and then make sure that there is a Primary E-mail Address. I do the Primary E-mail Address check in the Trigger Filter of Event 100, a Sequential Event. As you may remember from the chapter titled **The Tables**, the second byte of the **Zip** field, when set to **1**, indicates that this E-mail Address is the Primary Contact E-mail Address. If this Contact record has a Primary E-mail Address, I want to send them my birthday E-mail Template. At the same time, I also make sure that our Flag position is empty so that I do not send repetitive e-mails, and I do this via the addition of the **.and. empty(left(Contact2->Comments, 1))** clause.

My next Event in this Process, is our first Sequential Event, number 105, and is to **Update** the **Comments** field with a character (**1**). This will prevent this process from being re-added later today or tomorrow should the Automated Processes be scanned through another time. We will look at this **Flag** in the **Trigger Filter** of our Observer Process Event when the time comes. I use the Immediate trigger as there is nothing really to test for in this case as we just want the Flag to be updated immediately.

The next Event is Sequential Event number 110, and it is here that we hold the process in place for a period of time, in this case 6 days. This is the second way that we prevent the Observer Process from re-adding the Birthday Process. Remember, in our Process Options, that I asked you to select: ☒ **Allow only one attachment of this process per contact?** Well as long as the Birthday Process is attached to the Contact record, no other copy of the Birthday Process may be attached to the Contact record. Combine this with our Flag, and we have doubled our protection against duplication of activities. At the end of the 6 days, this Event will trigger, and the Action is to reset our flag field to be utilized next year by this process.

The last Event is to remove the Process. We don't like to rely on GoldMine removing the Process after the last Event has fired, although in theory and in practice, GoldMine removes the Process after the last Event has processed. As a safety precaution, however, I force the removal of the Process through Sequential Event, number 115 of the Birthday Process. It doesn't hurt to be overly cautious when programming your Automated Processes.

At this point you should fill in the Contact2->uDOB field with a valid birthday that falls within the next 5 days. You should be able to test the Birthday Process to assure that it is functioning as expected. Go to your Contact record in GoldMine, or any testable record, and click on the Processes tab. Assure that there is no other Birthday Process under this tab, and if there is, right-click and remove that Process. Right-click now under the Processes tab, and select **Attach a Process** from the local menu. The **Attach an Automated Process** dialog form will appear, and all of the processes contained within GoldMine will be listed, alphabetized by the Code (remember I suggested that you always employ codes, well there is your reason). Select the DOB Birthday Process for the list. Your process should begin its scan immediately. Remember, you selected the option: ☒ **Execute this process immediately upon attachment.** You should see the E-mail departing immediately unless you had selected to queue your message. You should notice that the 1st byte of the Comments field is displaying a 1. You should see the Birthday Process under the Processes tab with the next Event registering as **Trip Comments Flag 1 Off**.

Let me say, right here and now, that I would never advocate the use of Automated Processes for a single Event Process. Processes should consist of a series of events that are to be processed in a sequential and logical order. True, I could have done all that I had just done in a single Process, however, then you are not looking at the whole picture, the future of your Automated Processes. Each Process has a finite number of Preemptive and Sequential Events, and this, alone, promotes our philosophy of using an Observer Process to watch for events, and then fire the appropriate Process based on the Event that is being triggered. It also allows you to adhere to our **Tip** (see sidebar). Solve each Process individually, and then once solved, execute it from the Observer Process. This whole strategy makes excellent sense in a corporate environment where one could have hundreds of processes.

All of this has prepared us to begin adding an Event to our Observer Process that will attach the Birthday Process at the appropriate time. I am going to ask the Observer Process to watch for a birthday falling within the next 5 days, and if it does trigger on this condition, then to have it add the Birthday Process and scan it immediately.

Right-click on the **Observer Process, S 100 - Hold Process to Record**, and select **New...** from the local menu. This will bring us into the GoldMine **Automated Process Event Wizard** as shown here in Figure 10-6.

Notice that I have inserted **Birthday Within 5 Days** into the **Event Description:**, and that I have changed the **Event Type:** to **Preemptive** while setting the **Sequence No:** to 10. Click on the **Next >** button to move to the next page of the Wizard.

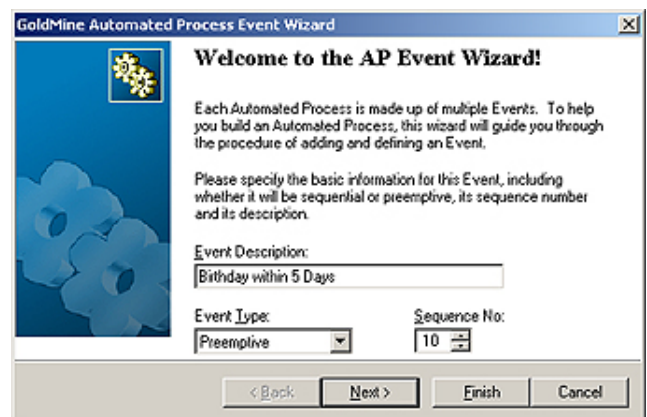


Figure 10-6

Figure 10-7, presented on the next page of this book, shows the next page in this Wizard. Notice that I have selected **dBase Condition to Trigger on:**. Next I would ask you to click on the **Options** button to produce the dialog form shown in Figure 10-8, well almost, refer to the **Trick** that I talked about in a sidebar note on the previous/next pages.

This particular expression, as shown in Figure 10-8 on the next page, could have been entered using the Expression Builder, however, there will be cases where you will need/want to create ex-

Tip

In the **Conclusion** section of this chapter, I will discuss how to turn on a **Process Monitor** feature in GoldMine that will display the Step-by-Step triggering of Events.

Tip

Do not attempt to solve the whole puzzle in one sitting. Take little pieces, and solve those. Eventually the whole picture will come into focus.

Note

You may be asking yourself: **Where did DJ come up with the 5 days?**

I make the ascertainment that the AP's are not scanned over the weekends, or weekends with holidays preceding or following them. By utilizing a 5 day frame, I believe that I have caught most of the Birthdays that would have otherwise been missed when not scanning the AP's over the weekends.

I do hope that you find my logic to be solid.

Note

Preemptive events must have a **Sequence No:** between 10 and 99. In theory, 1 thru 9 work as well, but I have witnessed Event ordering difficulties when using those numbers.

Trick

You can create any dBase expression that you want to in the AP's if you follow these simple procedures:

- Create any expression using the **Expression Builder** shown in Figure 10-8.
- Click the **OK** button.
- Click the **Options** button, shown in Figure 10-7, to bring up the **Expression Builder**, Figure 10-8.
- Highlight everything in the expression box, and begin typing your own expression by hand. **Do not use the Expression Builder.**
- Click the **OK** button when finished.

If you have typed a valid expression, even if it is not against the **Contact1/Contact2** tables, which is all that the **Expression Builder** allows, your expression will be saved, and utilized.

Note

A previous reader expressed concern that she did not realize that I was using two single quotes (') in my expression. She interpreted the two single quotes to be a single double quote ("). Hence, in this edition of the book at least, I have changed the expression to utilize the dBase alternative to ' or ":

`dtos(Contact2.uDOB) > []`

...the opposing square braces.

Note

Unlike the **Filters**, which utilizes a similar dialog form to that shown in Figure 10-8, this **Expression Builder** does not require balanced dBase & SQL statements. A simple dBase expression will suffice.

Note

Because I am looking for a birthday that occurs in a range of 5 days, and I would only want to add the **Birthday Process** once during that range. I, therefore, employ a **Flag** field. The first **Sequential Event** of the **Birthday Process** will stuff this field with a character so that this **Filter** condition can not be met, hence, the **Birthday Process** will not be attached a second time during the period that the **Flag** is in place.

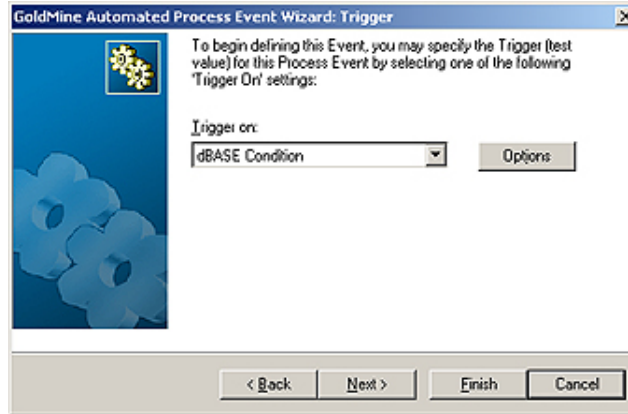


Figure 10-7

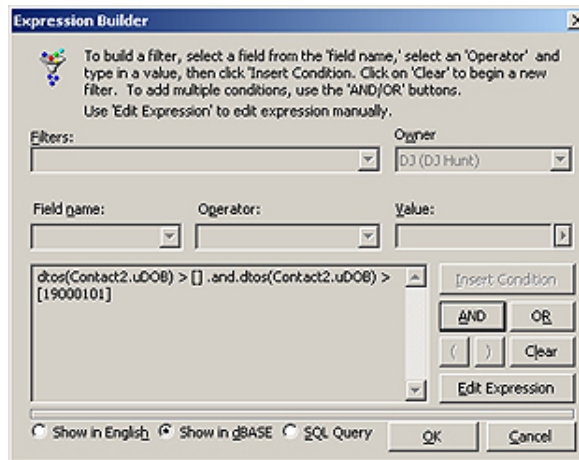


Figure 10-8

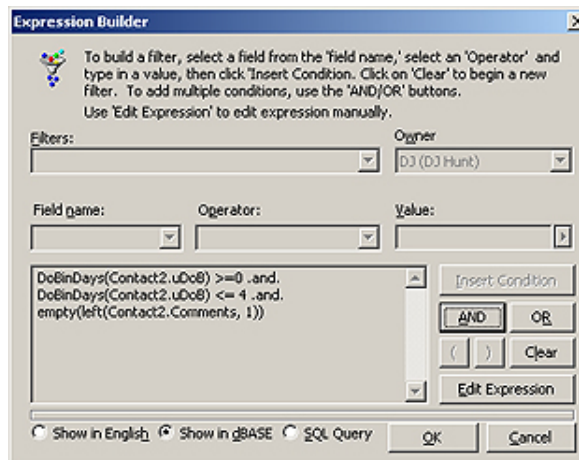


Figure 10-9

and then on the **Next >** button to bring up the next page of the Wizard.

If you are following along, you should select the **Perform Action:** of **Add a New Process** if this trigger is pulled. I have then clicked on the **Options** button to bring up the dialog form. From the dialog form I would ask you to select the, already established, **Birthday Process**, while checking the option to ☒ **Process immediately**. Now click **OK**, then **Next >**, and then **Finish** which concludes the building of this Event.

As I eluded to earlier, I am going to expand this chapter for this book, as many of you are still baffled by the complexities of the GoldMine Automated Processes, and rightly so. What I thought that I would do for this edition of the book was to show you how to utilize a single step Process to attach a historical activity en massé to a Filter/Group. This was so often requested that Richard Young of Prior Analytics Limited created a little applet to do just this, however, you know my philosophy:

expressions by hand. Follow the sidebar Trick, and you should be okay for all future hand built expressions. In the expression, I am just testing two possible conditions to assure that the field contains a birth date. You must test for both cases, especially as your backend is SQL. A SQL date based field has a default string date of 19000000, hence, it is not empty, yet it is not a valid birth date. Although I don't actually have to test for an empty date field, I do. That is the old dBase programmer in me, and it doesn't hurt to test for empty in the off chance that the field is actually empty. Again, not likely, but there is always that chance. At this point I would click on the **OK** button to save this expression, and then click the **Next >** button to proceed along the Wizards path.

Heavy duty stuff isn't it? I'm getting a brain cramp just trying to write about this. Yet, we haven't seen the worst of the expressions.

In the next Wizard screen, no figure shown, I do select the ☒ **Trigger Filter** option. I then click on the **Filter** button to bring up that Expression Builder a second time. I follow the Trick discussed in the sidebar, and then enter my filter expression by hand as this expression could not have been created using the Expression Builder.

Look at the expression that I've employed in my trigger, Figure 10-9. I have employed the **DoBinDays()** function (Appendix A) to make sure that the birthday is greater than or equal to 0 days away, and less than or equal to 4 days away, and that our Flag field (see sidebar Note) Contact2.Comments, first byte, is empty. Notice that these are boolean **.and.** conditions such that all must return a **True** before the expression itself can be considered to be **True** or triggered. Having finished that, I click on the **OK** button,

Create a Filter/Group Historical Activity

"When it can be done within GoldMine, do not do it outside of GoldMine".

So let's begin by creating a single step Process. Right-click at the top of the tree in the Automated Process Manager over the words **Automated Processes**, and select **New...** from the local menu. As you have already seen in the past, this begins the **GoldMine Automated Process Wizard**. In the **Process Name**: field enter **Create a Historical Activity**, and why don't we enter CHA into the **Process Code**: field, while leaving the default **Owner**: value alone. Now let's click on the **Next >** button. That's it, Step 1 of the wizard is finished.

On the screen of the wizard, we will only select 3 of the options:

- ☒ **Allow only one attachment of this process per contact**
- ☒ **Execute this process immediately upon attachment**
- ☒ **Allow users to attach this process**

All of the other options should be unselected, and now you may click on the **Next >** button for this step of the dialog form.

You should now be at the **Events** step of the dialog form, and it is here that you would like to click on the **New** button. I'm going to show you the completed dialog form here in Figure 10-10 so that you can see exactly what I've created.

As you can see above, I have entered a **Event Description**: of **Create a History Activity**, while making this a **Preemptive** event with a **Sequence**: number of **10**. Naturally, I would want the **Trigger on**: to be **Immediate**, and, as you would expect, the **Perform Action**: is set to **Create History**. After all, isn't that what we are trying to achieve here?

Figure 10-10

Now I want you to click on the **Options** button. I will now be referring to Figure 10-11 shown here. You will notice, for the **Activity Type**, that I have selected **Others**. When you are creating this, you may select any RecType to your liking. For our **User**, I have selected to utilize the **Attaching user**. Again, you may not find this appropriate for your activity, and that decision remains with you at the time that you are modifying your Automated Process. To continue on, you would want to click upon the **Activity Details** button to bring up the **Complete an Other** dialog form as shown populated in Figure 10-12.

Figure 10-11

Many of the fields will be populated at runtime by GoldMine so I have only pre-populated the **Reference**:, the **Code**:, and the **Result**: fields. You could, if you want to, even include **Notes**: for this particular series of completed activities. When finished, click upon the **OK** button, and then upon the **OK** button of the **History Action** dialog form. Finally, you can click upon the **OK** button of the **Event Properties** dialog form, and then upon the **Next >** button of the **Events Wizard**. Yes, there is one more button, the **Fin-**

Figure 10-12

Recommendation

*Yes I always use a **Code** as well as a **Result** value in all of my Completed activities, and I highly recommend that you do this as well.*

For Instance: You might make that code specific to your campaign.

ish button, and then we are through with this stage.

Note

You will retain this single **Event Automated Process** in your repertoire of tools, however, the dialog form shown in Figure 10-12 is the one that you will need to change each time you want to utilize this Automated Process to create a different series of historical events for a Filter/Group.

Each time that you want to utilize this process to create a historical activity for a group of individuals you will first need to change the contents of the Events Completed Activity. Okay then, let's wrap this project up. We have our single Event process, so now what do we do with this? First I would ask you to **Activate** a small Filter/Group that you may have lying around. If you do not have one then simply **Tag** a couple of records, and that will suffice. With your Filter/Group activated select:

Tools
Automated Processes ►
Execute Process...

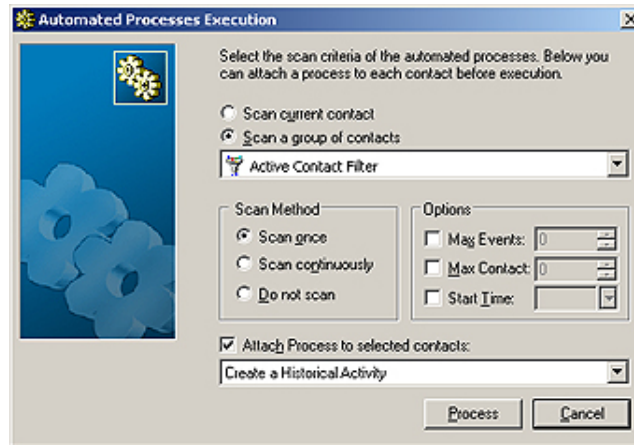


Figure 10-13

records, and because this process has been set to fire immediately upon attachment, the Event will process immediately, creating a new historical activity for each contact in the Filter/Group. Once the historical activity has been created, the Process will remove itself from the record as there is only a single Event in the process. When the **Process Monitor** indicates that you have finished with this activity, protocol dictates that you should release your Filter/Group.

For any future campaigns where you would like to repeat this action, you simply change the historical activity in the Event, and then attach that Process to the new Filter/Group. Could it get any simpler than that?

What I thought I would do now would be to look at each **Trigger** and, later, each **Action**, and I would discuss them in more detail. Although the Triggers have not changed in this edition of GoldMine Premium, the Actions list has had a new entity added, **Create Case**. Here I show you all of the possible Triggers/Actions for this edition of GoldMine Premium:

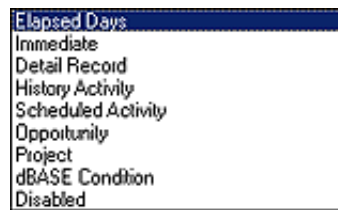


Figure 10-14a

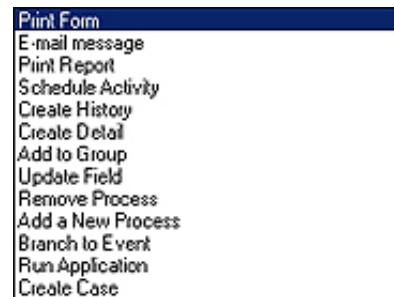


Figure 10-14b

You will notice that the **Scan a group of contacts** is selected in Figure 10-13, and that the field will display **Active Contact Filter** by default when a Filter/Group is activated. What I have selected, that is not selected in the default state, is to **Attach Process to selected contacts**, and with that I have chosen the process which we have just created called **Create a Historical Activity** using the drop-down menu. When you click on the **Process** button this Automated Process will be attached, to all of your Filtered/Grouped

Triggers

Elapsed Days: This trigger can be a bit tricky. I have seen users believe that this is the elapsed days since the process was added when, in fact, it is x number of elapsed days since the last Sequential Event had been triggered. You would probably never use this trigger in a Preemptive Event.

Immediate: This trigger is self evident. It is how one would turn a Sequential Event into a single use Preemptive Event, so to speak. You could use this trigger in your Preemptive or Sequential Events, however, in a Preemptive Event this would trigger every time that your processes are scanned, and you may want that to be the case yet I would doubt it. In my opinion, this is not a condition that you are likely to ever want to use in a Preemptive Event. That being said, I have used this when populating fields with default values, when there were too many default field values for me to employee the Lookup.ini for my client.

Detail Record: As can be seen in Figure 10-15, it is possible to trigger on **Details**, but you could trigger on other ContSupp RecTypes (yes the RecTypes field is what GoldMine uses to differentiate

between the different ContSupp records as you read previously in The Tables chapter). You can also trigger off of:

- Document Links
- Additional Contact
- Referrals

That is, you can trigger off of the creation of one of these RecTypes into your ContSupp table. Therefore, you must also tell the trigger properties how far back you are willing to look for the creation of one of these RecTypes from the day upon which the Event is being scanned.

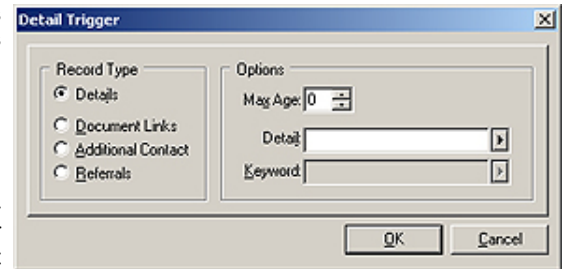


Figure 10-15

Note

Setting **Max Age**: to 0 days means if that Contact record has that RecType existing on any of its ContSupp records, regardless of the day that they were created, well that would be enough to trigger this Event. The 0 value is rarely used except in a sequential event, and never used in a preemptive event without the additional use of a **Flag** in the **Trigger Filter**.

Think about this now. If you make the **Max Age**: 5 days, and you scan your processes daily, then this trigger would be pulled every day for the next 5 consecutive days following the creation of this RecType if this were the trigger for a Preemptive Event. Not, probably, what you were trying to achieve, hence, I refer you back to the use of Flags in the Trigger Filter to prevent this type of Event from executing more than once. On the other hand, if you enter a Max Age: of 1 day, and the RecType is added on Friday, and you don't scan your processes until Tuesday next, because of a Monday holiday, you will not trigger this Event either. What a conundrum. Personally, I would rather use the **Max Age**: of 5 days with a Trigger Filter on a Flag field for my Preemptive Events than to risk the chance of not pulling the trigger on this Event at all. Naturally, this is a moot point for Sequential Events as a Trigger can only be pulled once per Sequential Event before the process moves forward to the next Sequential Event.

You can further refine your criterion, when **Details** is selected, by picking a **Detail**: from the F2 Lookup List such that you are only looking at the new P RecTypes that contain that value in the **Contact** field (you see I know this because I've read The Tables chapter in this book). Also, you can further refine your trigger by using a **Keyword**:. This field is available for refining your trigger regardless of the RecType that you wish to trigger against, whereas, the **Detail**: field is only available when the **Details** radio button option is selected.

History Activity: As can be seen in Figure 10-16, this type of trigger is similar to the Details trigger. You first have a frame for the **Activity Type** (aka RecTypes), with a frame for **Options**, and a frame that I have never had the opportunity to utilize labeled **Outcome**.

In the Activity Type listing, you have most of the possible RecTypes that the ContHist table could possibly contain. Refer to The Tables chapter for the various RecTypes that are utilized here. You can only trigger on a specific ContHist RecType, and there is no option for Any History Activity.



Figure 10-16

I don't think that I need to rehash **Max Age: 0** day for you, however, you are being introduced to three new trigger refinements, **Activity**: (ContHist.ActvCode), **Result**: (ContHist.ResultCode), and **User**: (ContHist.UserID). These three should be self explanatory. And, even though the **Ref.:** field is a different nomenclature, it acts, and reacts similar to the previously discussed **Details Keyword**: field.

Although a History trigger allows you to refine further for **Any Outcome**, a **Successful**, outcome or an **Unsuccessful** outcome, in my 20 years of working with GoldMine, I have never seen anyone use this flag when Completing an activity within GoldMine. That being said, it has been a useless refinement to the triggers for my clients, however, that does not mean that your organization may not want to use this trigger refinement.

Scheduled Activity: This brings us to the Scheduled Activity Trigger, and, as with the History Activity, we can see that the Trigger dialog form is pretty much the same, Figure 10-17. Naturally there are fewer

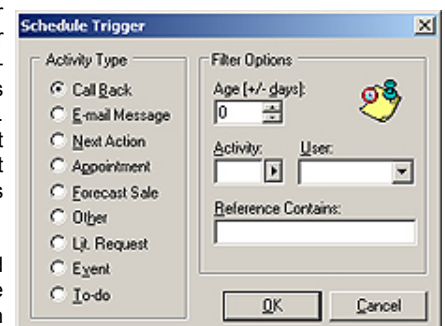


Figure 10-17

Note

The default, **Max Age: 0**, tells GoldMine that any History record of the specified RecType that was ever created against this Contact record is satisfactory to Trigger this Event.

Be weary of leaving the default setting unless you have an absolutely good understanding of what, and why you are doing so.

You would, most likely, not want to utilize this setting in a Preemptive Event.

Note

The default, **Age [+/- days]: 0**, tells GoldMine that any Scheduled activity record of the specified RecType that was ever created against this Contact record is satisfactory to Trigger this Event.

As I'm interpreting this, a + indicates Pending activities scheduled for a previous date, while a - indicates Pending activities scheduled in the future.

Note

My testing has shown me that a -3 day Trigger option will be pulled for any Cal.OnDate that is within 3 days into the future from the date of the scan.

I say this, not so much because it is an issue, as it is just a functionality that you should be aware of when creating your Events.

RecTypes to use as triggers, and, as there can be no outcome with a scheduled activity, there is no outcome refinement area. This holds true also for the fact that you see no **Result:** field on this trigger dialog form.

Having said all of that, you may have noticed that there is yet another difference. Instead of **Max Age:**, you will notice **Age [+/- days]:**. As this may be throwing a curve ball at you, I think that I will examine it in a little more detail for your benefit. Normally, one would not schedule an activity for a past date. As a test, I scheduled a Call for tomorrow. I set the trigger Age [+/- days]: to +3 days. I attached the process which was set to scan immediately upon attachment, and absolutely nothing happened. I then changed the date of the Pending activity to yesterday, and scanned the Process which did Trigger on this pass. To further test this out I changed the Age to -3 days, and I changed the Pending activity OnDate ahead to tomorrow again. I scanned the Process again, and this time it Triggered again, and it Triggered every scan as long as the OnDate was 3 or less days into the future. When I moved the OnDate to 4 days into the future and did the scan the Trigger was not pulled. Also, and just to complete my testing, I set the OnDate to yesterday, and when the age was set at -3 days, and the AP's were scanned this Trigger was again not pulled.

Here is what the GoldMine Help files have to say about this field:

Age (+/- days): Specify, in days, how recently the Scheduled Activity must have been scheduled on the calendar. Use the plus sign (+) to indicate past days, and the minus sign (-) to indicate upcoming days. Example: if you specify +3, it triggers the event only if a calendar activity was scheduled for completion within the last 3 days. If you specify -3, it triggers the event only if a calendar activity is scheduled for completion within the next 3 days. The default value of 0 = an activity on any date triggers the event.

Well, that certainly seems contrary to my way of thinking. A + equals the past, while a - equals future. I guess that I am from the old school, and that I do not understand this logic. In my school days, don't ask when, you went back in time (-), and you went forward (+) to the future. However, as long as you understand the developers logic here, then you should be able to create a working Trigger to meet your constraints.

Figure 10-18

The next Trigger that I want to examine is an **Opportunity Trigger**. Clicking the **Options** button on this Trigger brings up the dialog form shown here in Figure 10-18. In the **Max Age:** field, you would want to select the age of the Opportunity in days. You may use the spinner to increment or decrement this number, however, I am not sure why you would decrement the number past 0, although you can. You could cull your requirements down further by selecting a UserID for the Opportunity from the drop list associated with the **Managers:** field. You might also want to limit Opportunities that were of a specific type (name) in the **Opportunity:** field. In the Options frame there is one more filterable option, and that is for the **Stage:** of the Opportunity. Lastly, you see a radio button frame, **Status**, where you may further limit your criterion to either an ☒ **Open**, ☐ **Won**, or ☐ **Lost** Status for the Opportunity.

Figure 10-19

There is very little difference between the Opportunity trigger, and the **Project** trigger as you can see here in Figure 10-19. In the **Max Age:** field, you would want to select the age of the Project in days. You may use the spinner to increment or decrement this number. Again, you have the **Manager:** field, this time the **Project:** field, and again, the **Stage:** field to further refine your trigger criterion. This time, in the **Status** frame, you only have the two possible radio choices of either: ☒ **Open** or ☐ **Closed**. You understand, that either a Project is on going or it has been completed, hence, only a need for one of two options. Pretty much, everything that applied to the **Opportunity** trigger is applicable to the **Projects** trigger.

Note

*I believe that the **Disabled** trigger was intended for developers use to turn off certain known functioning events so as to not have to process them each, and every test scan. This is purely speculation, however, I can think of no other use for this trigger.*

The next trigger, **dBASE Condition**, I have already covered in this chapter a few times, and do not choose to elaborate any further on it in this section. However, the last trigger, **Disabled**, interests me in as much as I have no clue if it even functions, and if it does function, what exactly is its function. I will investigate that now for you.

The GoldMine Help file for GoldMine Premium states:

“Turns off the trigger condition setting for the selected event.”

But I would have to ask myself, Why? Is this so, you can leave the Event in the process, but not have it active? If this is the last sequential event will it be tested, and the process be removed? Give me a few minutes. I want to go run some tests.

TaDa! Testing has shown me that the Trigger is not tested, and that the Event is ignored. If this is the last Sequential Event, then the Process is removed. If this is not the last Sequential Event, this Process is just by-passed, and GoldMine moves directly to the next Sequential Event. I'm baffled. This trigger works exactly as defined in the Help dialog. What is this world coming to?

Actions

Let's move on. As that was the last trigger, I can now move my discussion forward to the various **Actions**. First, and foremost, you must understand that the Event Actions can do no more than you could do within GoldMine, and sometimes less because they have to function automatically. There is no user logic in the Automated Processes area that will make human decisions for you.

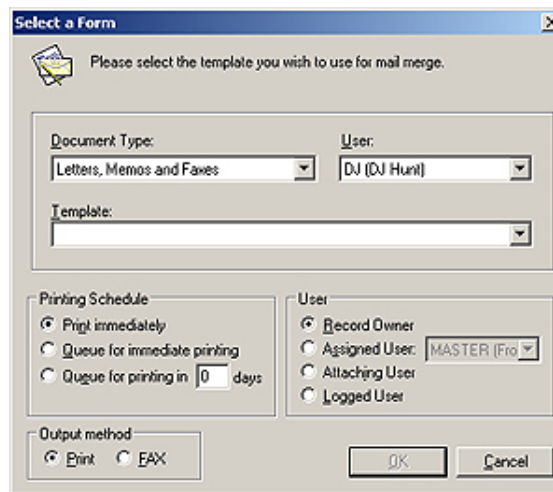


Figure 10-20

in this process.

The first **Action** on the list is **Print Form**, and, quite simply, this allows you to print a templated form. First off, let me state that I never, ever use this option in an Automated Process. First: this is an unattended and automated process, and you never know if there is paper in the printer, if the printer is on line, if the printer ink has run out, etc. I suppose, if you had the next Event trigger immediately, and it was set to schedule someone to check to see if the print job occurred, and if so to handle it, then it may be an acceptable Action. It is my thinking, however, that you should not even bother with this Event, and only have a single Event to automatically schedule someone to do the print job, as well as the handling of it after the printing. That would be the extent of automation that I would want

Having said that, you can see in Figure 10-20 that you can select any **Document Type**: for any **User**: using any of their templates, **Template**:. Once selected, you may choose a scheduling method in the **Printing Schedule** frame. You have the default ☒ **Print immediately**, or ☐ **Queue for immediate printing**, and finally, ☐ **Queue for printing in XX days**. The queue concept is the most difficult to understand so I ran some tests so that I could better explain it to you.

For instance: looking at Figure 10-20, if I had selected ☒ **Queue for immediate printing**, and as I also selected ☒ **Assigned User: Master**, I then ran the event, and this is what my debug log showed:

```
0[47] Automated Processes [10:51 am - 11/29/2008]
0[47] Scanning Contact: Computerease; DJ Hunt
0[47] Read Process: Test for Book [Next Event: 100, Process: BYCSUEV$[:] R#Y]
0[47] 100 Calendar +/-
0[47] 110 Second Event
0[47] --> Triggered.
0[47] Queue Form [3MJ2QBC.XN$/{72}
0[47] Remove ended process: BYCSUEV$[:] R#Y
4[47] Automated Processes: 1 Contacts; 1 Scanned; 1 Triggered; 1 Pass(es) [Ended 10:51 am -
10/29/2008; Dur: 0:00]
```

However, and this is where everyone gets lost, “*Where the heck was it queued?*”, and “*How do the queued print jobs get automatically printed?*”. This is where we have to look at two, yes two other areas of GoldMine. To see where the job gets queued, you need to look into the GoldMine **Literature** tab, and, in this case, under the **User: Master**. You should be able to find the job under the tree branch **Queued Documents**. You could right click on the document, and select **Fulfill** from the local menu, however, where would the automatic in that be?

Note

*In this edition of GoldMine Premium **Literature Fulfillment** has been truncated to simply **Literature**.*

Location:

GoTo
Literature

Note

Should you choose the **Merge for** option of **Selected User**, GoldMine will only permit you to select a UserID for which there is a defined E-mail Account under that UserID's Options.

Note

Selecting the ☐ **Other** option does not permit you to select another E-mail Address for the primary contact. My assumption here is that your GoldMine will automatically choose the next E-mail Address for the Primary Contact that has not been marked as the Primary E-mail Address, yet which is associated with the Primary Contact. The GoldMine Help files were not very conclusive on this definition.

Other: Sends to a different e-mail address of the primary contact.

Note

If this is an Automated Process, and it is, you should think hard and strong as to whether you really need to ☐ **Save the template text in history**. Personally, I don't believe that this is necessary.

Also, don't ask me why the default value for the **Result:** field is a 0. I have no clue. It was probably just the result of some buggy coding in that area.

This brings us to the "*How do the queued print jobs get automatically printed?*" question. For this answer, we have to look into the GoldMine **Server Agents**, under the **Print/Fax** tab. Yes, you have to set up Server Agents, and, as with GoldMine Automated Processes, GoldMine must be running, and someone or something must **Start Server Agents**. Ah ha! There goes automatic right out the window.

Figure 10-21a

Figure 10-21b

Figure 10-21c

Okay, let me wrap up this Action which, as I stated previously, I never use anyway. You can see that there is a frame called **Output method** from which you can choose to ☒ **Print** or to ☐ **Fax**, and that there is another frame called **User** from where you may select the user to be the ☒ **Record Owner**, ☐ **Assigned User:**, ☐ **Attaching User**, or the ☐ **Logged User**. I believe that all of these are self explanatory requiring no further explanation on my part.

And, moving on to the next Action of **E-mail message**, and click on the **Options** button for this action will make available the dialog form shown here in Figure 10-21a. As you can see there is a lot that has to be defined for this type of action for it to be utilized properly.

On this dialog form, you must select the template to be used, and, as you know, each user could have their own set of templates. The Action doesn't care whose template you use, but you must specify the **User**: as well which **Template** of that UserID to be utilize. You can then tell GoldMine from whom the merge should appear to be coming. You will have 4 choices (not shown) under the **Merge for** option, and they are:

Record Owner
Selected User
User who attaches this Process
User who scans this Process

The **User** field will only be active if you select to merge for the selected user. If you have chosen this option, you must select a UserID from the supplied list for this field. GoldMine will then use the Option settings, as defined for this UserID, when sending an E-mail via this particular Action.

In Figure 10-21b, you can see that you must define some **Recipient** information. Whether this e-mail is to go to the ☒ **Primary address** only, or whether to send it to ☐ **All addresses linked to this record**. The next grouping is inactive, and remains so, unless you choose ☐ **All addresses linked to this record**. Selecting this will allow you to further clarify if you want this e-mail sent to the ☒ **Primary**, ☐ **Ad-**

ditional contacts, or ☐ **Other** E-mail Addresses for the primary contact. You may further ☐ **Limit to E-mail merge code**: from where you can enter, or select from your F2 Lookup list, a merge code. If you decide to use this option, you should consider using the F2 Lookup list of Merge Codes.

Lastly, you have your **Delivery options** (although plural, you may only select one method for transport):

Queue for delivery
Send immediately
Save as draft

Note

In this build of GoldMine Premium on the **File in History** dialog form, you will notice that there are two Hot Keys that are the same:

Reference:
Result:

I have reported this to FrontRange, and I expect that it will be fixed before the build goes Gold.

Figure 10-21c, **File in History**, is the last dialog tab form. Here you determine if you want the Event Action to ☒ **Create a history record** for you. Once selected, all of the other options on this tab become enabled. You may now ☐ **Save the template text in history**, and you may override the **Reference**: with your own information instead of using the template subject in this field. As well, you may select, or add, your own **Activity**: and **Result**: codes which I highly recommend if you plan to use this feature.

This, then, concludes the E-mail message action for your events, and allows us to move forward to the action of **Print Reports**. First, and foremost, let me state that I hold the same reservations against the use of this action as I did for using the **Print Form** action. The printer must be on line, someone must know that the report was supposed to be there, etcetera, etcetera. Again, my first choice would be to schedule someone to perform this action, and to not choose this Action ever at all. With that in mind, you are offered this dialog form from which you may select the report (any report, belonging to any UserID within GoldMine). You must also determine if the report is to be printed against ☒ **All** contacts, or only just the ☐ **Current** contact. Previously you had the option of using a ☐ **Filter or Group** of records, however, this option has been removed with this release of GoldMine Premium. You cannot predefine the printer to which the report is to be printed. This is pretty much another WYSIWYG, and there is very little else that you can do with this Action. Please consider my suggestion of not utilizing this Action, but instead, using the next Action that I will be discussing with you, the **Schedule Activity** Action.

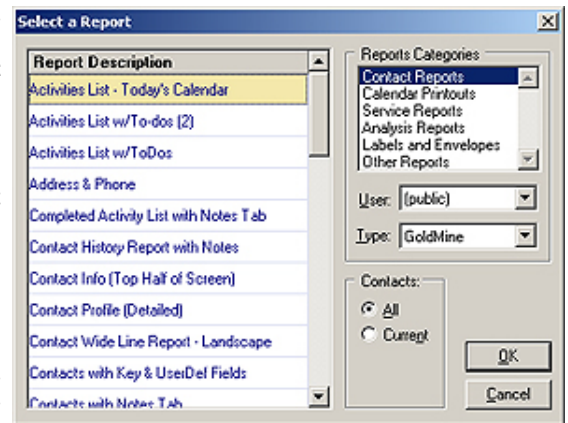


Figure 10-22

So it comes as no surprise that the next Action that we encounter is the **Schedule Activity** Action. Well now, that seems like something that I would like to know more about as I've already mentioned it a couple of times in lieu of directly doing other Actions. Here you have it, anything that you can schedule by hand within GoldMine can be scheduled via an Event Action.

Look at Figure 10-23a which is a result of clicking on the **Options** button, and Figure 10-23b, on the next page, which is a result of clicking on the **Activity Details** button in Figure 10-23a. I will describe Figure 10-23a first before moving on to Figure 10-23b. You know, the natural order of progression, a before b. Ah, let's just move on.

In the **Schedule Activity** frame, you have radio buttons, each representing a different RecType in the GoldMine Calendar (refer to The Tables chapter of this book). As with all radio buttons, you may only choose one of the available RecTypes.



Figure 10-23a

The next frame allows you to identify the **User**. The first, and default option is ☒ **Record owner**. If you choose this option, and you have a single UserID identified as the Owner of a record, then the UserID from the Owner field will be used when scheduling an activity. Should you choose to select the option of ☐ **Assigned user**, then the UserID that you have specified under **Activity Details**, discussed later, will be employed. Then we have the third option, ☐ **Attaching user**, which I have

Tip

Should you follow my lead on this one, you would be wise to modify your **Lookup.ini** to auto-populate these fields when a new record is created in GoldMine. Remember that GoldMine cannot schedule an activity if there is no **UserID** in a field. I use the **Lookup.ini** as you can only use fields from the **Contact1/Contact2** tables for this activity, and, coincidentally, the **Lookup.ini** can only populate fields in the **Contact1/Contact2** tables.

never had the opportunity to use, however, this applies to the UserID of the person who will be attaching the process to the record in the first place. Holding the fourth position is **Logged user**, which is another option that I have never had the opportunity to employ, however, the obvious here is that this means that the UserID of the Workstation that is currently scanning the processes will be used when scheduling an activity. Which brings up our final contender, and the one that I prefer to use most often, **User from field**. Yes, I make it a habit of assigning each record a series of fields, Sales Representative, Account Manager, Manager of Account Manager, etcetera, and I always select this radio button when scheduling an activity. I then proceed to point to the appropriate field that will contain the UserID for this or any scheduling Event.

I'm going to pass on the **Activity Details** button for the moment, and finish with the remainder of the **Options** frame, **Schedule x days into the future: 0**. Well this is almost obvious, however, when you throw in the radio selections this becomes a little confusing, at least to me it does. You have the choice of selecting one of two options, **From today** or **From trigger date**. Choosing the second radio option produces a crystal clear consequence, the activity is schedule x days after the trigger date, on the other hand, selecting the default option is not quite as clear. From the GoldMine Help files:

Select one:

From Today: Starts counting days from the current date.

From Trigger Date: Starts counting days from the date the event is triggered.

Well then, you have to ask yourself, exactly what is the current date? The date that I'm developing the Event? The date that I attach the process? The date the process is triggered (I think not or why have the second option)? I really can't figure this one out, and obviously, the GoldMine Help files were, well, or no help at all. Please let me know if you've figured this one out.

Next is a single checkbox option which is self descriptive, ☐ **Skip Weekends**.

Figure 10-23b

I see no reason to change this from the default at all.

In the **Reference:** field you could get a little creative if you so choose by utilize dBase syntax functionality. For Instance:

[Call]+&FirstName+[to follow up on order]

You are reminded that the **Reference:** field does have a character limitation, but within that limitation, you may use any valid dBase expression.

Notes: would probably not be a good option for an Automated Process, however, you could add standardized notes should you so desire.

This now brings us full circle, and back to the **Activity Details** button, refer to Figures 10-23a & 23b. It is here that we may define some specifics of the activity to be scheduled. This is a normal **Schedule a Call** dialog form except that some of the fields have been disabled because we are Scheduling an Activity via an Event Action.

First, let me say, that even though you can select a different contact, don't do it as it is a wasted effort. This field is populated automatically as a result of the triggering of this Event, and the Action defined. You may, however, choose to accept the default of having the Action scheduled call ☒ **Link to selected Contact:**, in fact,

Utilizing an Activity Code, **Code:**, is always a recommendation from my mouth. Personally, I use AP so as to designate that the activity was scheduled by the GoldMine Automated Processes, and not by an individual, however, this will only work properly if you have **Force valid input** turned off for this Codes F2 Lookup List.

Personally, I would not select an **Opportunity / Project:** for an Automated Process entry, yet that is strictly up to you as to whether you would utilize this or not.

At this point, you would want to define your **Primary User:** if you had selected ☐ **Assigned user** when on the previous dialog form, otherwise, just leave this field alone.

You may designate a **Time:**, however, as this is an Automated Process, you could get a lot of activities scheduled via the AP for the same user, on the same day, at the same time, for the same duration. Personally, I enter a character here, **A**, forcing this activity to the **Task** (untimed) area of the calendar as an untimed activity. Your end users will be much happier with this resulting activity than they would have been with the default timed activity.

Lastly, I always select the ☐ **Alarm:** checkbox. Did I say always? Yes siree, I certainly did. I prefer the in your face approach on activities that are scheduled automatically for a user. Do you really believe that your end users will always look at their calendars? I think not unless you somehow tie their paycheck to the calendar. However, they can't miss an alarm that pops up in their face each morning when they log into GoldMine.

I have never selected anything from the **Actions** frame, although it is certainly possible. I won't go through the three options that you have in this frame as they are all self descriptive. If you do utilize these in your normal schedule paradigm, then you may want to consider employing them via your AP's as well. For instance: ☐ **Mark as Private** may be something that your organization employees for scheduled activities. This may be a Security paradigm that your organization chooses to utilize.

Note

This type of Action is often used in GoldMine where no history was created when doing an eBlast for a campaign.

People would create a single Event process that would be processed immediately upon attachment, that would trigger immediately, and would create a history record to identify this record as having been a member of this eBlast. Attaching this Process to the Filtered group would then create a history activity for each member of the Filter group.

*Refer back to the section of this chapter titled: **Create a Filter/Group Historical Activity**.*

In list order, we now have another Action to look at, **Create History**. This is another two dialog form Action as you would expect because what is a History activity after all, but the completion of a Scheduled Activity, hence, one would expect these two Actions to be similar in nature. As an Automated Process cannot complete a scheduled activity, the Actions of the AP Events permit you to create a history activity. In point of fact, I utilized this very Action earlier in this chapter with you in the **Create a Filter/Group Historical Activity** example.

Figure 10-24a is the first of the two dialog forms that one will run across. Again, GoldMine has the **Activity Type** frame that this history record will be associated to, and, again, the list conforms to those RecTypes that are available within GoldMine. There is no difference in the **User** frame selection options from those that I have previously mentioned for a Scheduling Action. Refer back to that if you need a refresher on these options. Again, we have our **Activity Details** button (with a different Hot Key this time), however, this time GoldMine produces a variant of the **Complete a Phone Call** (if you had selected the RecType of ☒ **Phone Calls**) dialog form, Figure 10-24b. There is not much more that I can explain about the filling in of this form, however, I would remind you that you could get inventive with your **Reference:** field information. Additionally, I always recommend a **Code:**, and a **Result:** be input.



Figure 10-24a

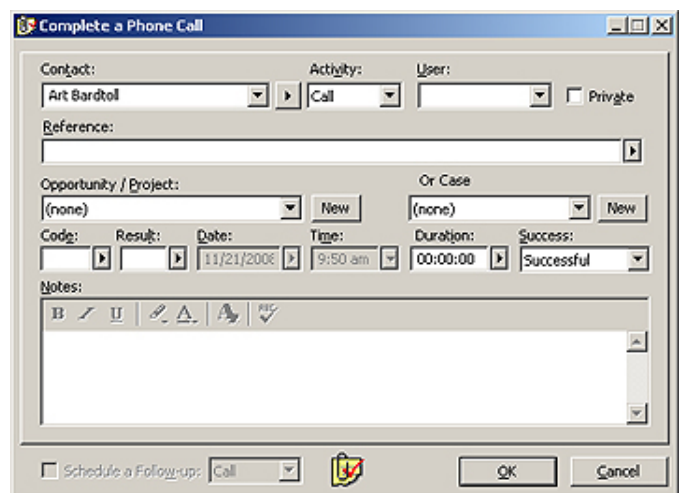


Figure 10-24b

Figures 10-25a & 10-25b show us the Action for **Create Detail**. You are allowed to select from any of your pre defined Details, and/or create a New Detail. If you have set up the **Info** tab (and you can't

Note

The extended Details have now exposed 12 fields:

Field 1: [35 characters, Title]
Field 2: [20 characters, LinkAcct]
Field 3: [20 characters, Country]
Field 4: [20 characters, Dear]
Field 5: [20 characters, Fax]
Field 6: [10 characters, Zip]
Field 7: [6 characters, Ext]
Field 8: [20 characters, State]
Field 9: [20 characters, MergeCode]
Field 10: [40 characters, Address1]
Field 11: [40 characters, Address2]
Field 12: [40 characters, Address3]

Note

As you cannot create Groups via the Actions of an Event, you must select a **User**: that has Groups available. You will then be afforded the opportunity of selecting one of that users' Groups from the **Group**: drop list.

You may have noticed that the **OK** button is disabled. GoldMine will not even allow you to accept this Action until you have select a **Group**:

Note

The **Look-up a replacement value in lookup.ini** will only activate if you choose a field that appears on the left hand side of the equation under the AutoUpdate section of your Lookup.ini. i.e.

[AutoUpdate]
Key1 = Key3

This works counter intuitive to the norm. If you actually wanted to update the Key1 field with the instruction set in the Lookup.ini for the Key1 field, you would need to have something similar to this in your AutoUpdate section:

[AutoUpdate]
Key1 = Key1

This basically says, if the Key1 field changes, then to apply the instruction set for Key1 to control how the field changes.

You can read more on this in the chapter which covers the **Lookup.ini**.

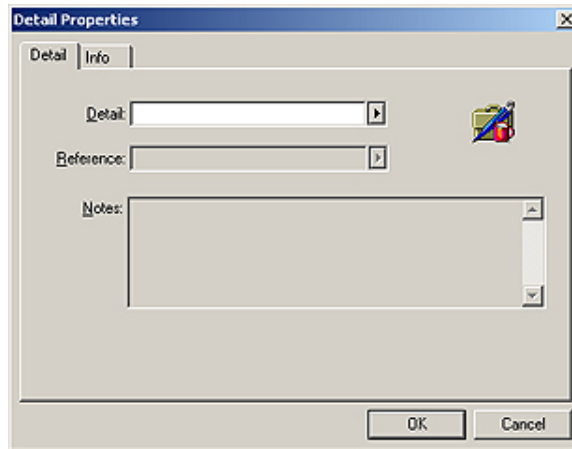


Figure 10-25a

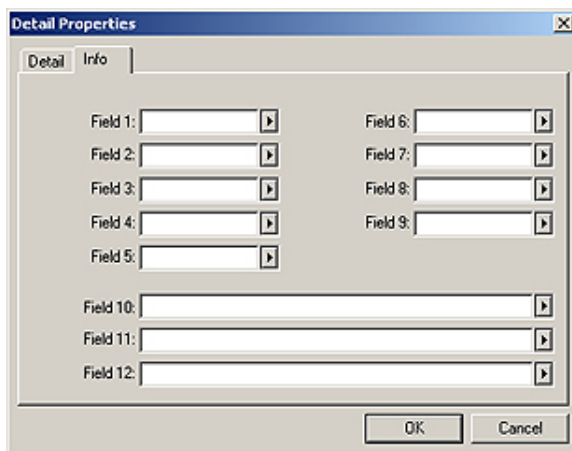


Figure 10-25b

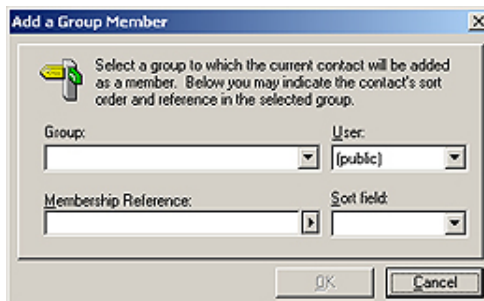


Figure 10-26

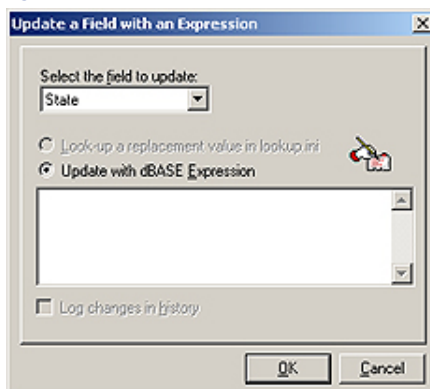


Figure 10-27

do that here) then you would be allowed to fill in any of these exposed extended fields as well. There really isn't anything more that I can explain when it comes to creating a Detail via an Action.

You may have noticed, in Figure 10-25b, that GoldMine Premium now has exposed 12 fields from the Cont-Supp table for Details as opposed to the previous 8 fields in the GoldMine Premium 8.0 series. See the sidebar for the fields that can be utilized for a new Detail record in today's GoldMine Premium.

The Action of **Add to Group** is even simpler. There is but a single dialog form to complete as shown on this page in Figure 10-26. If the selected **User**: has any Groups available, they will be listed in the **Group**: field (you may not create Groups here). You may then add a **Membership Reference**:, or select one from the F2 Lookup list, and select a **Sort field**: from the drop list. Once you have selected a **Group**: the **OK** button will be enabled for you to accept this Action.

Your next available Action, from the drop list, would be the **Update Field** option. I use this one very often, especially when I have the need to use Flags in my AP's to prevent the same action from occurring more than once during a given period. You will notice, Figure 10-27, that I have selected the **State** field from the drop list associated with the **Select the field to update**:, so that the Lookup radio option is enabled (see sidebar Note). I'm keeping this in here as it functioned properly in previous builds of GoldMine Premium, however, in the build that I am currently using, and as you can see in Figure 10-27, this feature is disabled despite the fact that I have the proper section in my Lookup.ini.

So now, instead of entering an **Update with dBASE Expression**, you can simply select the **Look-up a replacement value in lookup.ini** option, and let your Lookup.ini perform the appropriate task for you.

Barring that, you will have to create your own dBase expression, and remember, it is asking for an expression, and not for a value. For Instance: the Key1 field is a character field, and if we wanted to update this field with **Client**, we would need to enter "**Client**" or [**Client**] in the edit box. This edit box accepts any valid dBase expression, and will throw an error if you click on the **OK** button when there is an invalid dBase expression in the edit box. One reminder, always keep in mind the field data type that you are working with, and field characteristics of the field that you are updating. I mean don't try to stuff 50 characters into a 20 character field as it just ain't

gonna happen.

Another new option, that has been added to this version of GoldMine Premium for this Action, is the ability to ☐ **Log changes in History**. If the field selected to be updated has its Security feature set to Log changes to History, then this option, as shown in Figure 10-27 on the previous page, will be enabled for selection. Sometimes you may not want a field that is updated via an AP Event to create a History record, and then again, sometimes you will. Hence, FrontRange has added this as an option for which you are capable of making the final determination.

That's enough said on the Update Field Action. Let's look at the next Action in the drop list, **Remove Process**. The first thing that you may have noticed, no figure supplied, is that the **Options** button is disabled. That's right, this Action does one thing, and one thing only. It removes this process, and only this process from the triggering Contact record. This Action cannot be set up to remove any other process other than itself, and that's all that I have to say on the Remove Process Action.

Well, if we can remove a Process, you'd think that we could add a Process as well, and you'd be 100% correct. The next Action is to do just that, **Add a New Process**. Notice in Figure 10-28, produced by clicking on the **Options** button, that every Process that was ever created is brought up in the **Attach an Automated Process** dialog form (see sidebar Tip).

So, you see that you can add any Process as your Event Action that was ever created under any UserID as long as the process still exists within your GoldMine database, specifically the **Tracks** table. That's right. Although FrontRange may have changed Tracks to Processes on the dialog forms, the Table names have not been changed. Please notice the ☐ **Process immediately** option at the bottom of the dialog form. You must select this option if you want the selected Process to be processed immediately at the time that it is attached. As well, the Process selected must have been set to ☐ **Execute this process immediately upon attachment** (refer Figure 10-1). Other than that, there is nothing spectacular to say about the Action of attaching an Automated Process. So let's move forward.



Figure 10-28

Now we come to one of the biggies, **Branch to Event**, see Figure 10-29. Here you can get into some programming capability called branching script. Basically, this Action will permit you to branch to any other Event within this Process. Let me restate that, "...any other Event within the same Process.". You may not step outside the confines of the present Process using the Branch to Event Action.

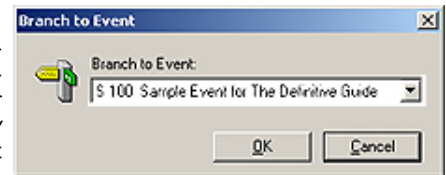


Figure 10-29

You must be very careful with your branching within the Process. You see, many times, a programmer will inadvertently program themselves into an infinite loop. This in itself is not bad. It is when they then turn around, and claim that the AP's don't work properly. Be very careful that you do not make this mistake. Adhere to the Tip in the sidebar, and follow the path of your branches to their logical conclusion to assure that you haven't placed them into an infinite loop. If you do happen to find yourself in this situation, you will need to shut down GoldMine to break out of the loop. Don't be embarrassed as you won't be the first to have done this, and I assure you that you won't be the last to do this. Where do you think the term, **Infinite Loop**, came from to begin with? Many have trod down this path before you including me.

And that brings us to the final Action on the list, **Run Application**. I have already clicked upon the **Options** button to bring up the dialog form shown here in Figure 10-30. With this Action, your Event may ☒ **Run Applications** or ☐ **Execute a DDE Command**. That is to say, launch an external application or execute a Dynamic Database Exchange (DDE) command.

If you choose the default, ☒ **Run Application**, then in the **Application executable file:** field, you must point to the executable. I find it easier to browse to the executable by clicking on the ellipses

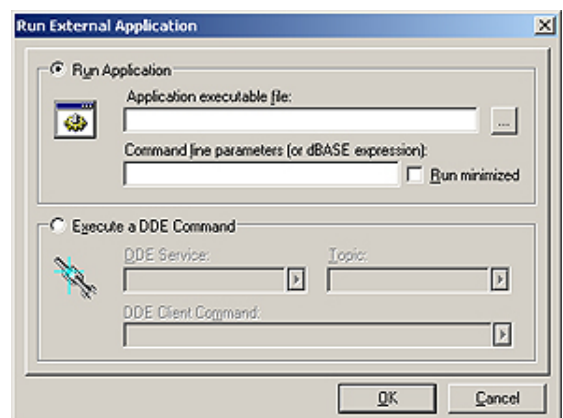


Figure 10-30

Tip

The **Attach an Automated Process** dialog form is brought up in **Code** order. Every UserID that has an Automated Process has their process displayed here. If you are trying to find a process that DJ created, it is extremely difficult.

Assign a single letter or single digit as the 1st character of the **Code** field representing a single UserID in your organization. That will give you 62 different groups of processes that you can have, and you will be able to find those created by DJ because you know that all of his processes begin with a lower case d.

Tip

Even the best of programmers occasionally programs themselves into an infinite loop. To assist in helping you avoid this situation, you should always **Branch to Events** that are Sequential Events, and that follow the Triggering Event.

Tip

Use your head. These are Automated Processes with the emphasis on them being **Automated**. Now don't go and ask your Event to run an external application that requires manual intervention. That is just a contradiction to automated.

(...) button at the end of the field. This way I'm certain that the resulting path is correct, and formatted exactly as GoldMine would want it to be. It might look something like this:

Y:\GoldMine\ClipDate\ClipDate.exe

Be very careful. Know from where your Automated Processes will be scanned. For Instance: Just because the mapped drive Y:\ is visible from my development computer does not mean that it will be mapped or visible from the computer that is performing the Automated Process scans. You should make an effort to have any executables, that will be run via Automated Processes, installed under the networked GoldMine folder. This way, you will be certain that the executable is visible no matter which networked computer is scanning the Processes.

Some external applications will accept **Command line parameters (or dBASE expressions)**; and, if the particular external application will accept them, they are to be entered via this field. For Instance: Let's say that you had a link ID to another database in the GoldMine Key5 field, and you wanted to pass that as a parameter:

[/u:] + trim(upper(Contact1->Key5))

That might just do the trick for you.

I'm really hesitant to talk about the **Execute a DDE Command** option as I have never had the opportunity to test this out myself. I will give you the information as supplied from the GoldMine Help files, and maybe you can make more out of it than I could.

To send DDE commands:

1. Select **Execute a DDE Command**.
2. Select a **DDE Service**.
3. Select a **Topic**.
4. Select or type a **DDE Client Command**.

Note: Each product application uses a unique format for DDE requests. Consult that product's documentation for details.

5. Click **OK**.

Now, based on my programming experience, I can tell you that a DDE Service might be GoldMine. While a Topic might be Data. As to what a DDE Client Command is, I am totally at a loss, and hoping that one of you may supply the correct answer. What I do know, from my programming experience, against the GoldMine API, is that you must establish a channel first. You may then use that channel to pull other information from GoldMine.

For Instance: Here is the Visual FoxPro code to do just that:

```
* Open DDE conversation
ThisForm.nCh = ddeInitiate([GoldMine],[Data])

* Retrieve Data
ThisForm.AccountNo = ddeRequest(ThisForm.nCh, [AccountNo])
ThisForm.Company = trim(ddeRequest(ThisForm.nCh, [Company]))
ThisForm.Contact = trim(ddeRequest(ThisForm.nCh, [Contact]))
ThisForm.SysDir = ddeRequest(ThisForm.nCh, [&SysDir])
ThisForm.GMSysDir = upper(left(ThisForm.GMSysDir, len(ThisForm.GMSysDir)-1))
ThisForm.GMBase = ddeRequest(ThisForm.nCh, [&GoldDir])
ThisForm.Run_Date = ddeRequest(ThisForm.nCh, [Contact2->udRun])
ThisForm.Run_Time = ddeRequest(ThisForm.nCh, [Contact2->ucRunTime])
ThisForm.Contract_Time = val(ddeRequest(ThisForm.nCh, [Contact2->unCntTime]))

* Close DDE conversation
ddeTerminate(ThisForm.nCh)
```

Now, as to how to translate that to the dialog form in Figure 10-30, I am at a complete loss. It is now time for you to step up to the bat. If you know how to correctly use this dialog form, then send me an e-mail to DJ Hunt <DJ@DJHunt.US>.

Now we have a new Action to discuss for this, the latest build of GoldMine Premium, and it is called **Create Case**. If you were to click on the **Options** button, you would bring up the dialog form shown on the next page in Figure 10-31.

You only have 3 possible options here:

☐ Use Existing Case Template:

Create A New Case Template

☒ Create an empty case.

As I have no Case Templates available, the default on this dialog form would to ☒ **Create an empty case**.

We have reached the end, and there aren't any more Actions to be covered for this build of GoldMine Premium. I think that I have been able to cover all of the available Triggers, and the the available Actions in enough detail to show you just how intricate a GoldMine Automated Processes could become.

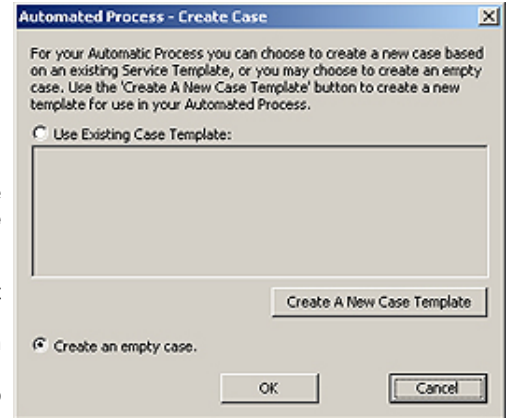


Figure 10-31

Tid Bits

I would like to continue on by looking at some of the subcomponents of events, and some related things that you could do that may help you achieve your goals through the use of Automated Processes.

Tid Bit 1:

As a rule, when creating events, I personally make heavy use of **dBase Conditions** both as a **Trigger**, and as a **Trigger Filter**. I have found, more often than not, that the dBase expressions allow me much more flexibility. In this light, I find the Age (+/-days): for Scheduled Activity to be very confusing to most GoldMine Administrators. I, therefore, always leave this set to 0, and make heavy use of a Flag to indicate to me whether a Process was or was not triggered. Witness the example that was shown to you earlier in this chapter. The **Contact2.Comments** field is seldom used, and affords you 65 characters for use as 65 individual Flags. Additionally, in GoldMine, you can hide from display, any field that you want to, with the use of the -2 expression as I discussed with you in another chapter in this book. Therefore, you could use your Comments field as your Flags area, and you could hide the Comments field, displayed on the Summary tab by default, from view. Alternatively, you could leave the field displayed, but make sure that only the Master user has Update Rights under the Security area for that field.

Tid Bit 2:

Another item, that I always advocate, is the use of user defined fields to hold the UserID for Events that will be creating a scheduled or a historical activity. In either of these activities there are radio button selections for the User. One of those options is to take a ☒ **User from field**. I may even employ a number of user defined fields as different events may warrant the creation of activities for different users such as a Manager, an Account Representative, etcetera. I have one client with 12 user defined UserID fields for all of the Processes that they utilize. I usually place these fields on an Administration screen that is visible only to those users possessing Master Rights.

Tip

Where possible program the Lookup.
ini to auto populate these field upon
the creation of a new contact record.

Tid Bit 3:

Did you know, when creating a scheduled activity, or a historical activity, that you can use an expression in the **Reference:** field that will be processed at the time the activity is created? Yes, I think that I mentioned this earlier. I've worked with a client creating the example that I will use here. Create an Event or modify an existing Event. Move to the Action page of the Wizard, and set **Perform Action:** to **Schedule Activity**. Click on the **Options** button, and then again on the **Activity Details** button. In the **Reference:** field you should enter:

[New Appointment on:]+ Wdate(Contact2->MeetDateOn,3)

That's it. When this activity is scheduled through Automated Processes, the **Reference:** field will be populated, and displayed where appropriate. Here is an example of the results:

New Appointment on: Wednesday, October 3, 2007

This will add more detail to the created activities, viewable on the users Day Calendar, as created via an Automated Processes.

Tid Bit 4:

Okay, if you've understood nothing else in this chapter, pay particular attention to this. Whether you are a GoldMine Consultant, a GoldMine Administrator, or even a GoldMine Enduser, you should adhere to these steps when developing your Automated Processes.

- Step 1:** Type up the process steps in Microsoft Word.
- Step 2:** Relate those process steps to GoldMine Event Actions.
- Step 3:** Determine what Trigger/Trigger Filter is required to cause the Action to be applied for each step, and insert it into Word preceeding the Action.
- Step 4:** Leave a space or two between the Action, and the following Event.
- Step 5:** Determine if there are any Preemptive Events that are required for things such as possibly, removing the process before it has completed all of its Sequential Events.
- Step 6:** Create your Process in the Automated Process Manager.
- Step 7:** Test your Process thoroughly against a single record.
- Step 8:** Create a Preemptive Event in your Observer Process that will cause this Event to be attached to the record at the appropriate time.
- Step 9:** Test the Observer Process/Your Process functionality against a single record again.
- Step 10:** Match your Word document to the final Process that you have created, fixing anything in the document that may have changed while actually building the process. Your goal is a detailed document that describes all of the Triggers, and Actions completely that you can refer to a year from now when you have to make changes to this Process.
- Step 11:** Copy the text of your completed Word document into your GoldMine KnowledgeBase for easy reference in the future.

Tid Bit 5:

In Chapter 3, I discussed the ini setting to display the Automated Process flow in the GoldMine Process Monitor. It is so important in the Automated Processes development phase that I will reprint it here:

■ Programmers - Display AP Flow within the GoldMine Process Monitor

There has been a need to debug an Automated Process for a long time now. To facilitate this, FrontRange had added the ability to switch On/Off the flow of the processes through the GoldMine Process Monitor. This is done individually within the UserID.ini or as a Corporate Override in the GM.ini (not recommended).

This is the switch statement:

```
[GoldMine]
APDebugLog=1
```

This is a typical result of said action:

```
0[1] Automated Processes [1:12 pm - 7/24/2007]
0[1] Scanning Contact: Computereze; DJ Hunt
0[1] Read Process: Update Department Field [Next Event: 100, Process: 9G4RA45$W=98R#Y]
0[1] 100 Test Update Field
0[1] --> Triggered.
0[1] Update Field: DEPARTMENT with DJ Hunt
0[1] Remove ended process: 9G4RA45$W=98R#Y
4[1] Automated Processes: 1 Contacts; 1 Scanned; 1 Triggered; 1 Pass(es) [Ended 1:12 pm -
7/24/2007; Dur: 0:00]
```

As you can see this could be valuable information when trying to debug an Automated Process. Of course, if you are perfect, and never make a mistake when developing your AP's this is, naturally, not required.

This concludes my discussion of Automated Processes. There are a couple of things that I would like to reiterate. Again, I would like to state that you should take baby steps with Automated Processes. Solve one Event at a time, and then put it into action using the Observer Process as the launching point for all other Processes.

Lastly, and most importantly, you must have a process in order to automate a process. That only sounds logical doesn't it? GoldMine does provide you with some examples under the **(public)** User Group. All of these examples must be attached to the appropriate record by hand. They do not make

Note

Should you set this option to display Automated Process flow, you want to make certain that you remove the option when you are done with your AP testing. This has been known to cause processing issues in the past.

Wrap Up

use of the Observer Process paradigm as I have discussed it in this chapter. They are, however, good examples of Processes, and expose you to a typical Process flow. These GoldMine Processes expose a wider range of the use of the available options than I may have exposed here.

It was my aim to expose you to a technique that I have employed over the years, as have many readers of my prior book "**Sales Force Automation with GoldMine 4.0**", which has been employed with success over the years. Sales Force Automation with GoldMine 4.0 was called the blueprint for GoldMine Automated Processes. The techniques discussed are still applicable to the GoldMine Premium product. You may still purchase Sales Force Automation with GoldMine 4.0, as an eBook only, and only from DJ Hunt <DJ@DJHunt.US>.

