

In This Chapter

The Basics

Referential Lookup List

Code-based

Text-based

Products

Multicast Gold

GoldBook

The Basics

Tip

Think through all of the possible choices for each and every field in your GoldMine installation, and add those to your F2 Lookup List. However, remember that a lookup list that has too many choices will probably be used as much as one that has no choices. There is a delicate balance that you must achieve between no list at all, and too large a list.

GoldMine Premium Edition 8.5s release is causing me to have to rewrite the entire F2 Lookup List chapter for the first time as FrontRange has finally added the often requested **Referential Lookup List**. I will still cover the basics as I have in the past, however, you can see in the sidebar that I have also added the new Referential Lookup List. I, as are we all, am extremely excited to finally see the inclusion of the Referential Lookup List into the GoldMine product line.

Consistency is the name of the game when it comes to database systems of any kind. I'm sure you will remember the acronym **GIGO** for **Garbage In, Garbage Out**. This is true in all database systems where one would require report output from the data entered into the database. It would be rather difficult to filter a database on a field unless that field contained a modicum of consistency that one could apply a filter against. Is it possible that you could know all of the possible variants for a piece of data such that you could even conceive of a possible filter condition to extract that data for exports or reports? I think not.

One of your first tasks should be to establish a vehicle which will provide for as much consistency of data which is entered into the GoldMine fields as is possible. GoldMine provides, for each and every field in its database an F2 Lookup List from which the pop-up list draws its choices. The name F2 Lookup List is a carry over from older versions, although it still functions today, such that, when the field is in the edit mode, the user simply hits the **F2** key, to pop-up a **Field Lookup** list of possible choices for that particular field. Having a well developed lookup list system in your GoldMine, could eliminate as much as 90% of the end user typing that would usually be required in a Customer Relationship Management system such as GoldMine. When you combine that with the Lookup.ini instruction set that I will be discussing later in this book, well, you can have a very clean and consistent data entry system.

I recommend that you establish just such a system, and that you further develop a common process methodology that advocates the usage of the F2 Lookup List whenever possible. To this end, you will find **Auto-Fill** to be a valuable friend, however, I will cover that later in this chapter. I just wanted to plant the seed in your mind early on in the chapter.

GoldMine, out of the box, is set up for use by FrontRange Solutions as a solution for their own internal needs. All of the field lookups relate to GoldMine in one manner or another, and as a GoldMine Administrator, it is your responsibility to make your GoldMine function efficiently for your organization.

Let's think about when it would be appropriate to supply a lookup list for a field, and when one should not be supplied for a field. Some fields are to contain unique values each and every time data is entered into these fields. Some of these types of fields could be the **Company**, **Contact**, **Address1**, **Address2**, and **Address3** fields. Each of these fields requires unique information each and every time that data is entered into the field 98% of the time for most organizations. Some might argue that the **Company** field could contain repetitive information, and one should, therefore, create a list for those company names that are repeated frequently. I might agree with that argument under certain constraints only. If your organization dealt with a very limited number of companies, and each of these companies contained multiple sites, then, yes, I might see a requirement for a lookup list associated with this field. However, few organizations deal with such a finite number of large companies that would have multiple sites, at least in my observations.

The key word, in the previous paragraph, is repetitive. If a field can contain information that would be entered repetitively on different records, then you, most probably, have a good candidate for a lookup list entry. Such a field might be the **Source** field. Let's look at the **Source** field as our example. To display just the lookup list, while the field is in the edit mode, just click on the arrow (►) button to the right of the field. From the edit mode there are only

Note

After 20 years of consistency in this area, FrontRange has finally decided to follow the **Windows Standards**. No longer can you **Right-Click** when a field is in the **Edit Mode** and expect to bring up the **F2 Lookup List**.

This is going to take some getting used to for us GoldMine old-timers. Even after eight months of using GoldMine Premium 8.5, I still have a tendency of right-clicking in the field. After a **Stupid Shit** mumbled comment about myself, I then go ahead and click on the arrow (▶) icon at the end of the field.

Trust me, eventually you will get the hang of it.

Note

Items in a lookup list can only be removed one at a time. If you have a substantial list of entries, you may want to consider using a 3rd party product like **GoldBox** which allows you to delete entire lists in a **Sync Aware** manner.

Alternatively, you may want to delete this list en masse through the **SQL Server Management Tools**, but be forewarned, doing so in this manner will cause your deletions to be **Sync Unaware**.

Note

You could use the **Field Name:** field as a mini instruction set to your users. As this is independent of the GoldMine field name, it is pretty much free form.

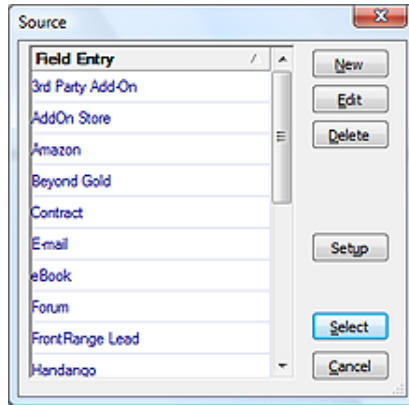


Figure 5-1

two ways in GoldMine Premium 8.5 that you can enter the F2 Lookup List window once the field has been placed into the edit mode. The first is to simply click on the **F2** key. The second method would be click the ▶ icon to the right of each field that has been placed into the edit mode. Either of these methods would bring up the dialog window shown here in Figure 5-1.

As you can see, there are a few entries in my Source F2 Lookup list, however, I only want to examine the buttons that are available to us at this time. As you will have your own Source item list. I would add that these items are only meaningful to Computerese Incorporated, and that they are only here as examples as to how this lookup list may be utilized. In other words, you may delete all of the default entries as they are probably meaningless to your organizational needs.

It is important for you to understand that all of the buttons shown in Figure 5-1 pertain to the lookup list dialog form only. This is a very important distinction, and one that you must come to understand. Clicking on the **New** button will allow you to add a new entry to your lookup list. There are a couple of points to remember here, the first being that this entry will hold up to 40 characters. In this lookup window, however, only about 28 characters of the possible 40 will be displayed.

Clicking on the **Delete** button will simply delete the selected lookup list entry from the list. It will not delete any information from the Contact record itself. You will receive the standard warning message: "<I<UserID>>, Delete the selected F2 record?" to which you must select **Yes** if you really intend to delete this entry, or **No** if you did not mean to delete the selected entry.

Next we come to the **Edit** button. Clicking on this button will allow you to edit the selected lookup list entry. This is just in case an entry should have gotten onto the lookup list that is misspelled or was entered by someone else, and that may not conform to your corporate standard. This just eliminates the need to delete the entry, and then re-add it as a new entry.

I'll discuss the **Setup** button in a minute, but first let's look at the **Select** button which simply selects the highlighted entry, and enters it into the field currently being edited on the Contact record. You may bypass the need to use the **Select** button by just double-clicking on the appropriate entry from within the lookup list. The selected lookup list entry will then be sent directly to the field that is currently being edited.

Next we have the Windows obligatory **Cancel** button which, when clicked upon, will close the lookup list dialog without entering any value from the list into the field being edited.

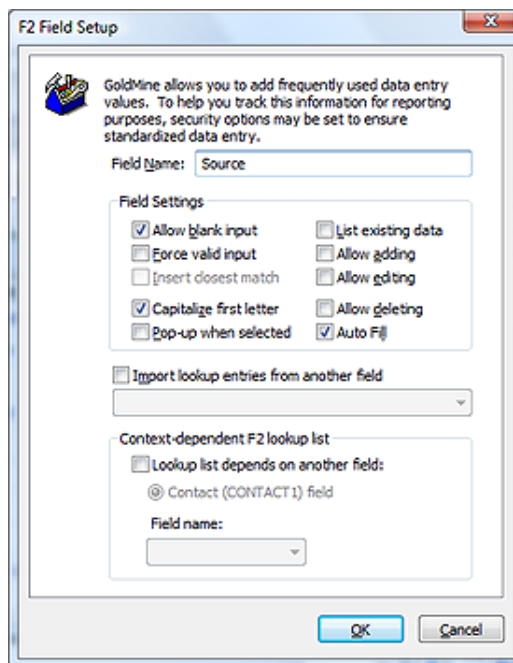


Figure 5-2

Now let's step back, and examine that **Setup** button. Clicking on the **Setup** button will bring forth the **F2 Field Setup** dialog screen shown here in Figure 5-2. The first item that we see in this dialog screen is the **Field Name:** field. Please don't confuse this with the field name that is being displayed on the GoldMine screen. The name that you enter into this field is the name that is displayed in the title bar of the lookup list, as shown, previously, in Figure 5-1. In that case, the name **Source** was the entry as it was the name that appeared in the title bar of Figure 5-1. It is always a good practice to have the name of the lookup list match that of the field to which the lookup list is associated. This way the user will always know which field they are editing. There is an exception to this rule.

Exception: In those cases where you are employing Record Typing for a particular field, in which case, you may want to leave this field blank.

Now we can look at the frame labeled **Field Settings**. **Allow blank input** is the first

Note

You may remember, in Chapter 4, when I discussed **Form Level Validation** which allows you to set a field(s) for required input. In addition to that, once they have placed the field into the edit mode, you may then prevent them from leaving said field if they have not entered information and/or appropriate information into that field.

Tip

If you uncheck the box for **Allow blank input**, and check the box for **Force valid input**, you must make certain that there is an out for the user. An out might be the use of **Other**, or **Miscellaneous**. Do not leave the user stuck in the field because there is no entry on your list that meets their particular condition. If you don't supply an out, then the user, in this case, would have to select an entry from your list which may be totally inappropriate for the field for this contact record..

option that one is faced with. Notice that I use the term option. Because this is a checkbox, you have the option of setting it to **True, checked**, or of setting it to **False, unchecked**. This option is checked in its default state. In its most simple terms, the default setting means that GoldMine should allow the user to leave the field empty when they are leaving the edit mode of the field. You must remember that GoldMine performs its valid checking as one is leaving a field that has been placed into the edit mode. For you developers, this is done through the **Valid** method in the code usually, however, there are other methods that could have just as easily been utilized. If you should choose the opposite, then a user would be required to enter something into the field that has been placed in the edit mode. This option setting, unchecked, by itself, only means that something must be entered into the field, and GoldMine really doesn't care what that something is.

However, should you combine the first option, unchecked, with the **Force valid input** option, then you would have begun to take control over the information that is allowed into the field. By default this is unchecked, and, in this state, GoldMine would allow any entry into the field that is currently being edited. Check this box, and GoldMine will require that whatever is entered into the field must also currently reside on the F2 Lookup list. If this is too stringent, then you may want to check the next option, **Insert closest match**, which will become available as an option once you have selected to **Force valid input**. This is an out, so to speak, in that, if one begins typing, and nothing on the list matches the entry, then the closest item, alphabetically, from the list will be inserted into the field. I have discovered that selecting this option could produce some very odd field data entry. I would rather see you supply your own out, and not use the insert closest match option at all as someone may enter the closest match, which is not at all correct, and easily not even realize that they have done so. End users, in my opinion, tend to not look at what is actually being entered or, for that matter, look at and read warning messages that are thrown at them.

Capitalize first letter is the next option, and may or may not be appropriate for your usage. You need to understand that this is not capitalizing the first letter in the first word, it is capitalizing the first letter of every word in the field. If you can configure your GoldMine so that the user will be allowed to pick entries from the lookup list most of the time, then this option will only control those few times that the user types data into a field by hand. As long as you understand that the first letter of each word in the field will end up capitalized, and this is the result that you are striving to achieve, then go ahead and check this option.

Pop-up when selected is an option that is both good, and annoying at the same time. If you should select this option, which is by default unchecked, then every time the user places the associated field into the edit mode, and, as importantly, the field is blank, GoldMine will pop-up the lookup list. You may wish to do this to force the user to realize that there is a set of standardized entries for this field. The annoyance factor comes into play when the user wants to type into the field. If this option is selected, they would be required to click on the **Cancel** button on the lookup list before they could begin typing. I would think that you would want to listen to a few complaints about this in order to achieve the consistency factor that I discussed earlier. If there are too many complaints, then you may opt for an alternative solution such as the **Auto Fill** option that I will discuss shortly.

List existing data is an option that was Added new to GoldMine Premium 8.5. First off, let me state that selecting this option will disable quite a few of the options on the **F2 Field Setup** screen. Now let me explain what it actually does do. Have you ever typed in a Source field value that was not contained within your F2 Lookup List? Selecting this option executes a SQL Query that will pull all of your unique values from the field in question, and populate your F2 Lookup List with the results of this query. So that everything that you have ever typed in by hand will now appear as a possible choice from your F2 Lookup List. Personally, I'm not sure how much value this adds to the F2 Lookup List.

The next three options I will discuss as a group. These three options do not, in any way, have anything to do with how the field is handled, but, instead, determine if you will **Allow adding**, **Allow editing**, or **Allow deleting** of the lookup list items by a user not processing Master Rights. It is important that you understand that these affect the lookup list and its entries, and do not at all affect the field that is being edited. I can't tell you how many times I have heard someone exclaim, *"We have discovered a bug, we had unchecked allow deleting and the users were still able to delete the information in this field."* In trying to achieve as much consistency as is possible, I always recommend that the GoldMine Administrator manage the lookup list in a corporate and consistent manner. In that vein, I do recommend that all three of these options be unchecked such that the users will not be allowed to add, edit or delete items from your lookup list. I further recommend that you establish a standard operating procedure by which the end user is able to notify the GoldMine Administrator any time that they would like an item added, edited or removed from a lookup list.

Auto Fill is the next option that we will want to take a look at. If you have received too many complaints about the pop-up feature, and are truly tired of listening to them, then this may be the option for you. Once the option is selected, whenever a user begins typing, the closest matching entry from the lookup list is inserted into the field. If the user continues typing, and you have not selected to force a valid input, then the users entry will be accepted in lieu of the suggestion from the lookup list. If, however, the user likes the suggestion from the lookup list, they simply have to stop typing, and hit

Tip

Auto Fill is especially useful if you have excessively large F2 Lookup lists.

Referential Lookup List

enter to select the suggested entry. This option is a more subtle approach to consistency. One point that I would like to make, either select the pop-up option or select the auto fill option, but never both options. For our clientele, and when appropriate, I always opt for the auto fill option over the pop-up option. One way to assist in the usage of the auto fill option, is to have the user make sure that their **Tools | Options | Record** tab option of **Select contents of fields** is checked or, better yet, by setting this for your organization via a User Override. By default this option is not selected. Please review Chapter 3 of this book for the individual Option settings. And that, believe it or not, was the last option for the **Fields Settings** frame.

Now, mid screen, there is a stand alone item to **Import lookup entries from another field**. This feature was often requested by end users in earlier versions of GoldMine where it did not exist. Let's just say that you were creating a lookup list for a user defined field of **Contact2.uBillState**. You already have all of the entries for this lookup list contained in the lookup list for the field of **Contact1.State**. You really would not want to have to type in that entire list all over again for the new field. That's exactly where this option comes into play. By selecting this option, you will activate the drop down combo list selection below this option, from which you may choose the field that contains the lookup list that you wish to have copied into the current lookup list. The lookup list will be populated from the selected field once you click on the **OK** button for this dialog screen. It should go without saying that clicking on the **Cancel** button will negate any changes that you have made in the **F2 Field Setup** dialog screen.

Now that we are familiar with the F2 Lookup List **Basics**, I can move on into **Code-based** F2 Lookups.

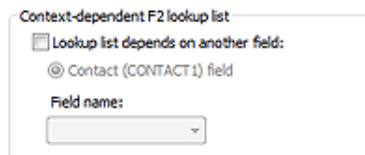


Figure 5-3

Wait, what's that we see in the dialog form? **Contact-dependent F2 lookup list** is new to GoldMine Premium 8.5, and refers to the GoldMine referential lookup list that we have all been asking for since 1988 when GoldMine was first released. Can you believe that we finally have it? I'm thinking that we'd better talk about Contact-dependent F2 lookup list before we move onto the Code-based F2 Lookups. Criminy, we've been waiting for this long enough.

Notice, Figure 5-3, that I have pulled out this frame independently for a closer look. One of the first things that I must say is that this frame will look different depending upon the area of GoldMine from where the **F2 Field Setup** screen is rendered. Currently we are on the **Source** field, and this is the frame that is presented to us from the F2 Field Setup screen for that field.

In order for us to use this frame, we must first enable the frame, and we do that by selecting the **Lookup list depends on another field:** option. The next option, in this case, is not really an option at all in this particular rendering of the F2 Field Setup screen. Use it as a radio button selection, however, there is only one value that can be selected for the field, hence, there is no need to enable the feature as there is nothing that the user can select other than that which is already selected. For the Source field, your only possible table for dependency is the **Contact (CONTACT1) field**, and the **Field name:** list should now be enabled as well. For this discussion, I'm going to select the **Key1** field in the drop list for Field name:. After having done this, you correctly observe that absolutely nothing has happened. Now isn't that strange? Well, actually no, as this is a multistep process.

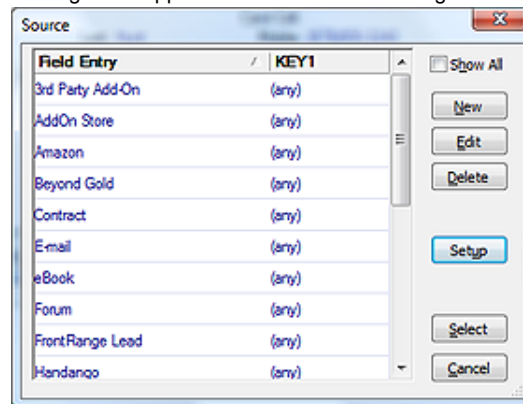


Figure 5-4

Let's click on the **OK** button, and look at the resulting F2 Lookup List dialog form for the Source field now as shown in here Figure 5-4 and compare it to the F2 Lookup List dialog form shown to you previously in Figure 5-1. There is quite a bit of difference now, isn't there?

Instead of having a single list column, you now notice that there is two columns available with the second column being the column that we selected previously on the Setup dialog form which, in this case, is the Key1 field. Currently this column shows the same value, **(any)**, for all list items. So is this any different really than what we had before? No it is not, at least

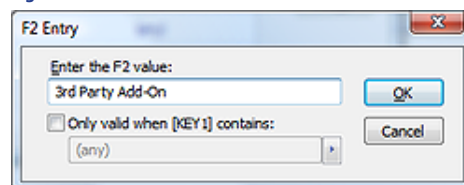


Figure 5-5

not without further editing. Go ahead, Highlight the first list item and click on the **Edit** button to produce the dialog form shown here in Figure 5-5. From the **F2 Entry** dialog form, one now has the ability to relate this list item to a value contained in the Key1 field. All that one needs to do would be to select the **Only valid when [KEY1] contains:** option, and then select one of the fields from the Key1 F2 Lookup List or by sim-

WARNING

While in the development mode, you want to make sure that you keep the **Show All** option selected as this will keep all of the list items displayed regardless of their relationship. Once you have completed development, you will probably want to reverse that option by unselecting the **Show All** option.

Code-based

Tip

I cannot emphasize enough the importance of the **Code** and the **Result** fields. Their usage is most noticeable when attempting to create reports. Each report has an **Options** button associated with it. In the options dialog area, you can filter your data on a number of things such as activity type, date range, and, what I am interested in at this point, the **ActvCode** and the **ResultCode** codes. Plan your codes well, and assure that your users are implementing them to assist in your reporting granularity.

Note

If you are a support based organization, you may want to use the very same **Contact Updater** application that we use in house.

ply typing in a value. Once you have done this for all of the items in the current F2 Lookup List, they will then be referentially related and displayed based on the value contained within the chosen field. You asked for it, and you've got it, albeit 20 years later.

As long as the now visible **Show All** option is not selected, your F2 Lookup List will now display only list items that are related to the Contact1 field name that you have selected. If you ever have a need to display all list items, as when in development mode, make sure that you select the **Show All** option to regain the entire list.

Let's send out a round of applause to the FrontRange Developers for finally including **Contact-dependent F2 lookup list** in the GoldMine product which I have chosen to rename to **Referential Lookup List**.

All right then, now we can catch up with the other books, and move on to the **Code-based** F2 Lookup List. These are the lookups that are associated with the scheduling of and/or the completing of an activity. GoldMine labels the activity code field (**Cal.ActvCode**), simply **Code:**, while it labels the result code field (**ContHist.ResultCode**), **Result:**. The Result field is only available when you are completing an activity, while the Code field is available when you are either scheduling or completing an activity.

Let's start by looking at the **Code:** field, and it's entries. In fact, let's look specifically at the **Next Action Activity Codes**. Here's an example right from my GoldMine, Figure 5-6. I show you this example only so that you may understand that what I say, and what I do, are not necessarily the same. In other words, I don't always listen to that which I preach. You'll see this as I continue on.

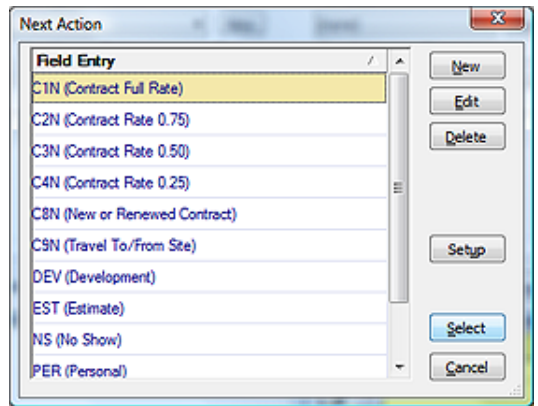


Figure 5-6

One must understand that **Code-based** fields are limited to **only** accepting a maximum of 3 characters. I want to emphasize this fact. Everything you see in Figure 5-6, beyond the first 3 characters, is only there for the purpose of code usage clarity to the end users. Therefore, any note indicators (//) would be clearly extraneous, and do not need to be used in these types of code-based lookup lists, however, not using them in GoldMine Premium 8.5 will throw a Warning message as shown here in Figure 5-7. Clicking on the **OK** button will still add the entry to the list however. I will discuss those note indicators in detail later on when I discuss **Text-based** F2 Lookups.

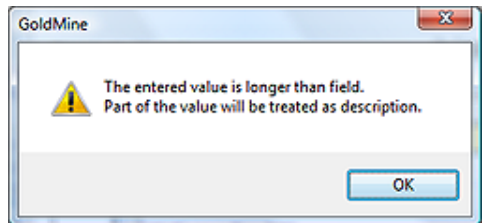


Figure 5-7

I cannot stress enough, if **Code-based** F2 Lookups are going to be used, as they most often are to aide in reporting, then they must be unique per activity. Emphasis is on the *“unique per activity”* part of that statement. You may use the code over again in a different type of activity as these are differentiated further by **RecType**, however, never use a code more than once in any single activity.

So, you might ask, *“What are appropriate codes?”*, and that is a very suitable question to ask. Figure 5-6 above, shows my employees choices for Next Actions as we use our Next Actions to track our clients **GoldMine Support Contract** time. Specifically, the first 6 items on this list are unique to my add-on application which calculates Contract Support time plus or minus based on an ActvCode beginning with C, and having a ResultCode of COM. If you are going to be reporting on how many support activities were made for a specific customer these may be appropriate. May I suggest that you would be interested in reporting on the various marketing campaigns that you have on going. Therefore, an appropriate use would be codes for the various campaigns. That way, you could easily report on how many activities were made for any given campaign, or how many sales were directly a result of a given campaign. This seems to me to be a more appropriate use of the activity code field. Using the codes shown in Figure 5-6, I could easily run reports that can be used to invoice a client for billable work accomplished. I could easily run a report to show our clients how their contract time was used, and, I might add, this is often requested by our clients. *“We couldn't have used that much time. Where did all our time go?”* I can't tell you how often I have avoided arguments just by printing and sending these targeted reports, and it was through the careful use of coding that made these reports easy and even possible at all.

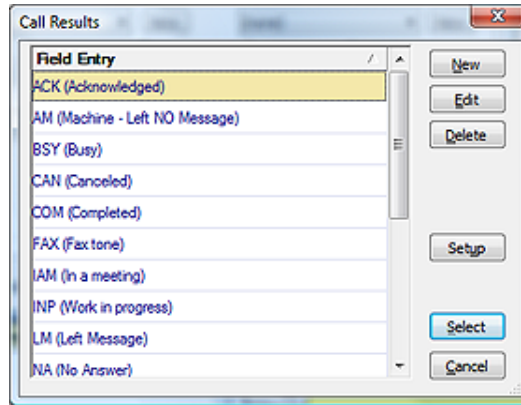


Figure 5-8

I won't look too closely at the **Result** field from the completed activities. They certainly don't hold the same significance as the **Code** field, but they do hold their importance when running reports. For example, I may not want to know all the calls for a given campaign as much as I would want to know all of the calls for a given campaign that were completed (**COM**) successfully. A call that was made where the caller left a message (**LM**) may not be as useful in your report. Using the **Result** field will further help you quantify your reports.

I show you Figure 5-8 for an important reason. This is the F2 lookup from the Complete Unscheduled Incoming Call activity.

Tip

For a multi use F2 Lookup List it may be helpful to annotate the codes to assist with your end user training:

- ACK //Acknowledged OC
- AM //Machine Left No Message OC
- BSY //Busy OC
- CAN //Canceled OC
- COM //Completed SC
- FAX //Fax Tone OC

Where:

- IC = Incoming Call Code
- OC = Outgoing Call Code
- SC = Shared Call Code

You might ask, "How can an incoming call be busy or have no answer?"

You might think that these are inappropriate codes to have in this type of activity, and you would be correct except that a few of the F2 lookups in GoldMine are shared among multiple activity types. This particular F2 lookup is shared between Complete Scheduled Call, Complete Unscheduled Outgoing Call, and Complete Unscheduled Incoming Call. This means, to maintain consistency, that you will have to be extra careful in your end user training sessions. You must make it clear when it is appropriate to use a code from a shared list, and when it is not. You may even find that explaining which activities the codes are appropriate for, as a note on the list item, to be more than beneficial when using this type of shared lookup list.

I thought it only appropriate that I should give you a little insight as to how this information is stored. All of the lookups are stored in the shared table **Lookup** in your database. Each field has its own lookup (with exceptions as discussed above) and, hence, its own lookup value. In the **Lookup** table, there are three fields of concern, **FieldName**, **LookupSupp**, and **Entry**. Let's examine the **Call Results Codes** data.

The header record for **Call Result Code** is in:

```
Lookup.FieldName = 'MCALLRSTCDH'
```

Notice that the 11th position contains the letter **H** for Header. This record contains all of the **Setup** settings for this Lookup in the **Lookup.LookupSupp** field. A typical setup might be:

```
Lookup.LookupSupp = 'NNYNNNNNN'
```

While the **Entry** field contains the Lookup window title.

```
Lookup.Entry = 'Call Results'
```

New to GoldMine Premium 8.5.x is the **Lookup.MasterValue** field which is used to support the new **Referential Lookup List**. In its default state it would look as:

```
Lookup.MasterValue = '(any)'
```

So where are the actual values that are displayed in the list? They are stored in subsequent records that can be located by:

```
Lookup.FieldName = 'MCALLRSTCDV'
```

Notice, again, the 11th position is a V for Value. To pull all of this together, one would simply need to execute this SQL Query statement from the query window:

```
select *
from Lookup
where FieldName like 'MCALLRSTCD%'
order by FieldName,
Entry
```

You programmers will be especially interested in this. Whenever I want to read tables to populate my own forms, I **never** use the API query. I always use a select query via an ODBC connection. I have populated the lookup list of many a form directly from the **Lookup** tables in GoldMine.

WARNING

If you are utilizing the **Merge:** field, you **must** be certain that only unique characters are placed into each byte. If a character/digit/symbol has been utilized once in the field, then be certain that it can't be entered a second time as I have done with my expressions.

```
~trim(strtran(MergeCodes,[E],[I]))+[E]
```

This expression first removes the **E** in any byte in the **Contact1.MergeCode** field before replacing it with another **E**.

Before I continue on and discuss Text-based lookup lists, I want to discuss a lookup list for a field that would normally be a text-based list, but that I feel should be utilized as a code-based field. That would be the lookup list for the **Merge:** field. You see this is a very important field in GoldMine, yet it only contains 20 character bytes. Hence, I'm going to suggest that you utilize each of those 20 characters as a code for a particular type of merge. Take a look here at Figure 5-9. This is my **Contact1.MergeCode** field lookup list. You'll notice that I have used special characters that I will discuss with you shortly. The significance here is, if I select all 3 of the expressions on the list, that the resulting value in the **Merge:** field will be EMN which only utilizes 3 of the 20 available character bytes in my **Contact1.MergeCode** field. This then means that I have 17 other possible character codes that I can enter, and that can be utilized in GoldMine when the **Merge:** field is used as a filtering condition. You may not see the significance today, however, as you add more and more campaigns, you may find this approach extremely useful.

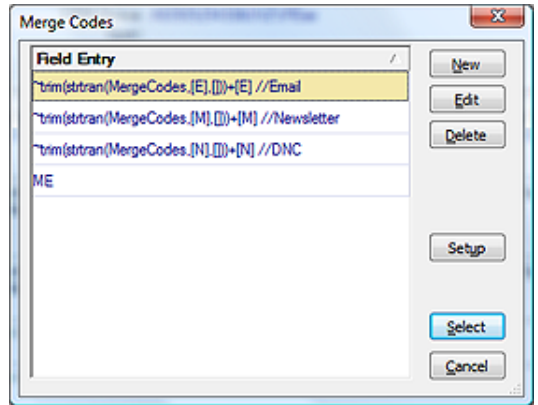


Figure 5-9

Text-based

I discuss the **Text-based** F2 Lookups separately from the code-based lookups although the text-based lookups could certainly be used as code-based, if properly set up. The difference is that I would need to define the coding paradigm for text-based lookups as I just did above for the **Merge:** field. Each field in GoldMine has a **Text-based F2 Lookup**. If the field is a user defined field, the field itself can be whatever length you want it to be, but the lookup will still be bound by the constraints of the lookup table structure. I have discussed these constraints earlier in this chapter for you.

Let's look at some of the special characters that can be used in a text-based lookup. There are now only three symbols that hold special significance to GoldMine when incorporated in a text-based lookup. They are the semicolon (;), paired forward slashes (//), and the tilde (~). Old time GoldMine users will remember the percent (%) sign functionality. In GoldMine Premium that functionality has been removed at the behest of the end users who had a need to select a percentage (i.e. 35%) from the F2 Lookup list, and have the selected item entered into the field.

Note

The semi-colon (;) is only useful for field presentation. If you are using the field as a code-based field, you will probably not want to waste 2 bytes of the field on a space and a comma.

Tip

Many administrators have wished to have their F2 Lookup List display in numerical order as opposed to alphabetical order. Whether you are working with **Code-based** lookups, or **Text-based** lookups, this is not currently possible. The list will always display sorted in alphabetical order, and there is no facility to change this order.

To have numbers sort properly, you must use all possible characters. i.e. 001, 010, 100, etc.

Note

New in GoldMine Premium 8.5 is that the paired slashes (//) no longer display in the F2 Lookup List.

Pre 8.5 Display Example:

Waldo Emerson //Owner

Post 8.5 Display Example:

Waldo Emerson (Owner)

When you are creating your own code-based system using a text-based lookup, you might, at first, be happy for the functionality supplied by the semicolon (;). GoldMine reads the text that you select from the text-based lookup, and if it encounters a semicolon at the end of the string, it interprets this as an append instruction. If there is nothing in the field to begin with, the append instruction is ignored. If something exists in the field, and the user selects another item from the text-based lookup with an append instruction, the second selection is appended to the first, separated by a comma and a space.

Let's start with the field as shown in Figure 5-10, and the associated text-based lookup as shown in Figure 5-11. If you were to select all of the items shown in Figure 5-10, the resulting value in the field would appear as shown in Figure 5-11. You could continue on indefinitely, or, at the very least, until you had reached the end length of the field as had been defined by the user or default by GoldMine.

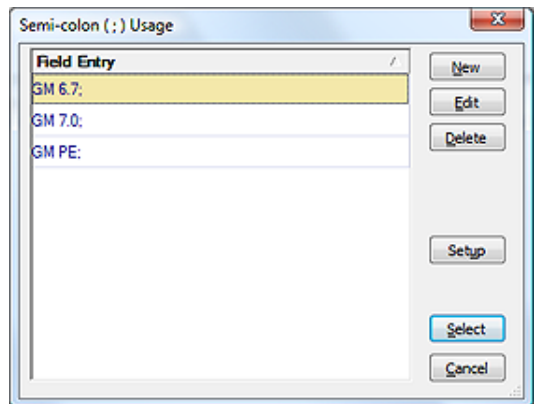


Figure 5-10

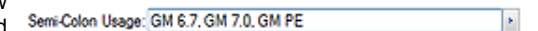


Figure 5-11

As I have already employed them, Figure 5-9 on the previous page, I suppose that I should discuss the paired forward slashes (//) next. You may have noticed that, upon selection of an item from Figure 5-9, that only the information up to (xxx) was entered into my field. We already know that the special characters do not get added to the field, but instead, they are translated to an appropriate function. *What then happened to the rest of the information?* The paired forward slashes in the string, indicates to GoldMine that anything that follows is to be considered as informational only, a Note. GoldMine will not enter either the special character, the paired forward slashes, nor any trailing text, contained in the same string, into a field value. There is an exception to this rule that I will discuss shortly, however, this

is quite a nice feature as it stands. Now you are able to create our own code-based information in a text field using a Text-based F2 Lookup list, and you can annotate the meaning of each code to assist your end users, refer to Figure 5-9. Again, you must be mindful of the 28 character display limitation of the lookup list itself. Additionally, you should never use the paired forward slashes in a Code-based F2 List. They are not necessary anyway as those fields will only hold the 3 character maximum field length of the code.

% The percentage sign possessed a similar functionality to that of the semicolon (;). However, alas, its functionality has been removed from your GoldMine Premium product. Too many individuals complained of their inability to use the % in the F2 Lookup list as a % for the value field like **35%**. So one has to ask them self: If the functionality was good, and it was, then why did the developers simply not utilize another special character? Possibly the carrot (^) symbol. I wonder if we will ever get an answer to that question?

Note
With GoldMine Premium, you no longer have to include the table alias when using the Contact1/Contact2 fields in your expression. This affords you more character bytes for your expression.

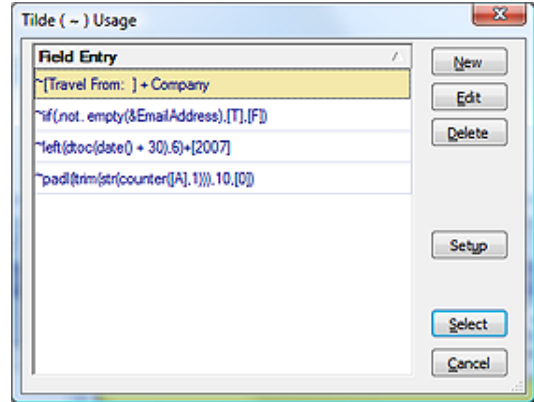


Figure 5-12
In this, my first example, as depicted in Figure 5-12, you can see:

The tilde sign instructs GoldMine to evaluate what follows as it would a function, and to place the resulting value into the associated field. I have supplied four examples of functions that may be incorporated into a lookup in Figure 5-12, as well, there are examples shown in Figure 5-9. I would emphasize, however, that these expressions could just as easily be used via your Lookup.ini, your reports, your templates, and other locations within the GoldMine environment. Your only limitation, when using a function via the F2 Lookup list, is the 40 character limitation of the field itself.

WARNING
I have found, in the GoldMine Premium version, that this macro does not always pull the expected information:
~[Travel From:] + Contact
I have found, if scheduling from the record using the menu that the &Contact macro works appropriately. However, if you are scheduling from the Calendar, where the Contact record is not then active, that macro pulls the first alphabetically positioned contact name. In my case it pulls Andrea Dominquez many times when I schedule using this macro from the Calendar.

~[Travel From:] + Company

This might be used in the **F2 Lookup** list for the **Reference** field of an **Appointment** activity for instance. This is a good example of combining character string population with the information contained in a GoldMine field. As I used to charge half rate for travel to and from clients, I always documented that time using **Appointment** activities, and, yes I always utilized the Code field. **C9S** is, in fact, the Code that I utilized. But I digress. This expression was often used to save typing, and worked well when reporting on said activities as well as when Publishing GoldMine calendars to the web that were meaningful. The result, as it might look, of selecting this expression is shown here in Figure 5-13.

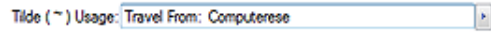


Figure 5-13

My second example will stuff a character based field with a **T** for **True** if there is a Primary E-mail Address for the contact record, and with an **F** for **False** if there is no Primary E-mail Address for the contact record. This function would best be employed in the Lookup.ini upon NewRecord creation. The expression is:

~iif(.not.empty(&EmailAddress),[T],[F])

The results of the evaluation are displayed in Figure 5-14. This is a good example of the **iif(logical expression, true, false)** function (refer to Appendix A) which can be most useful, as I've shown you, in the GoldMine field attributes.



Figure 5-14

My third example stuffs a character based field with the character representation of the computer system clocks date plus 30 days. The results of this expression:

~left(dtoc(date) + 30),6)+[2009]

generated today are shown to you here in Figure 5-15.



Figure 5-15

My last equation generates a 10 digit sequential number (refer to Appendix A). This can not be used as a unique number, but is the first step to creating a unique number. See the Lookup.ini chapter of this book for a more detailed explanation of use of this function to generate a unique number when in a synchronization environment. The function is:

```
~padl(ltrim(str(counter([A],1))),10,[0])
```

The results are displayed in Figure 5-16. However, I want you to notice that I have no spaces in the expression. Why is that do you think? Why is it that I used a single character **A** as the variable name? The limitations of the F2 Lookup list field for this value is 40 characters, and, if you'll count the characters utilized in my example expression, you will notice that I have employed each and every available space for this expression. There is absolutely no room available for extraneous characters like spaces, and longer variable names.



Figure 5-16

I have completed my description of the various implementations for your F2 Lookups, but I would like to take the time to reemphasize a point. The effective use of the F2 Lookups is your best vehicle to consistency of data input into your GoldMine tables. Make sure that you take the time to develop as many of the lookups as thoroughly as possible. I can not stress enough how much this will help you with any database activities such as filtering, or reporting. Time well spent now, will result in time not being wasted later.

Further, the mere developing of adequate F2 Lookups does not ensure consistent input, only the capability to have consistent data input. You must assure that your end users are well schooled in the use of the F2 Lookups, and it must be clearly understood that the use of the F2 Lookup option should be employed before the option of typing in the data.

Products

Even though FrontRange has removed the **Invoice** capability of a **Forecasted Sale** as of the GoldMine 7 product release, they have maintained the **Product Pricing** that Elan Susser incorporated into the code many years ago for a special client of his. As it is still functioning in GoldMine Premium, I thought that I should like to include it along with the F2 Lookup List capabilities presented here in this chapter.

I now need to discuss that special usage of the paired forward slashes as was previously discussed. Many GoldMine versions ago, a special feature was added to one particular F2 Lookup List. This functionality may disappear from future builds of

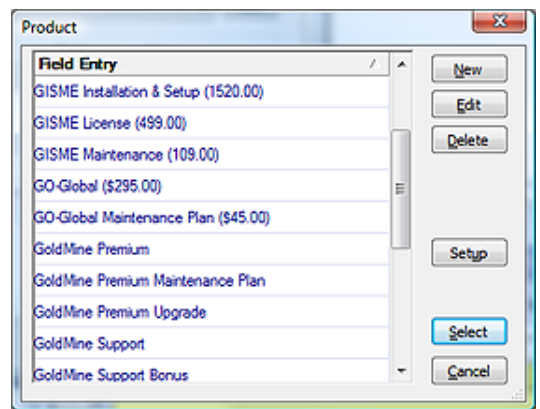


Figure 5-17

GoldMine, however, for now at least, that functionality still exists in the **Product:** field when you **Schedule** a **Forecasted Sale**. New with the GoldMine Premium release, this feature is now extended into the F2 Lookup List associated with the Product field when one is **Completing a Sale...** as well.

The functionality was intended to allow a GoldMine Administrator to supply a list of products along with their associated per unit cost. Let's say that I selected, and highlighted the first list entry in Figure 5-17. Not only would the **Product:** field contain the text, **GISME Installation & Setup**, but the **Price:** and **Amount:** field of this forecast sale would also have been populated with **1,525.00** and **1,525.00** respectively. This proved to be highly useful. Instead of supplying an associated price list that the end user could easily lose, and would have required updating for all users, the GoldMine Administrator need only update the pricing in two places (remember that the Forecast Sales and Completed Sales each have separate a F2 Lookup List). The prices would immediately be available to those on the network, and it would also be synchronized to those remote users when next they synchronized. As useful as this feature is, it was, and is, not a highly publicized feature of GoldMine. In fact, I don't know that it is publicized anywhere except in The Hacker's Guide series of books and now this, The Definite Guide, series of books.

Unfortunately, GoldMine Premium does break the compounding capability of the quoting functionality. In previous builds you could have used the semicolon (;) to compound the **Product:**, **Price:** and **Amount:** values. You'll remember that semicolon (;) was the append function special character. Alas, this capability is gone, and in today's GoldMine, the compounding will only work against the **Product:** field properly, and will not work properly against the **Price:** or **Amount:** fields.

MultiCast Gold

Select a user to populate the sales list on the left. Once the list is loaded, pick any sale to have the fields on the right populated. Add/review/modify the associated products below the global sale fields.

Save

User: DN

Clear Fields

Global Values

Contact: [dropdown] Reference: <-GM Complete

Probability: [dropdown] Notes: [text area]

Expected Sale Date: [text field with >]

Product List:

- 1 [Delete] Product: [dropdown] Code: [dropdown]
Units: [text field] Price: [text field] Amount: [text field]
- 2 [Delete] Product: [dropdown] Code: [dropdown]
Units: [text field] Price: [text field] Amount: [text field]
- 3 [Delete] Product: [dropdown] Code: [dropdown]
Units: [text field] Price: [text field] Amount: [text field]
- 4 [Delete] Product: [dropdown] Code: [dropdown]
Units: [text field] Price: [text field] Amount: [text field]

Figure 5-18

Many users had requested the capability of Product line items, and this functionality, which used to exist within GoldMine, is now no longer available. To bring back that functionality while still utilizing the capability of the Product F2 Lookup list functionality, I have designed a product called **MultiCast**. This is a product which I and others sell to end users, that you can purchase and that will work nicely with the GoldMine product. MultiCast will be your Sales Forecasting tool once instituted so you would want to remote **Schedule | Sale** and **Complete | Sale...** from the users menus in GoldMine. All forecast sales created via this instrument will be shown for each UserID in the column directly below their UserID in MultiCast. They may select any of these by double-clicking to modify or complete the activity. The paradigm for this application is that there is one frame for the Global Values, and then there are up to 10 line item products that may be entered. Each field in MultiCast has a lookup list that is pulled directly from your GoldMine F2 Lookup list for the related field. The Product field in MultiCast utilizes the same functionality that exist within the GoldMine F2 Lookup list for that product, and will populate the Price & Amount field in MultiCast with any related values. When the end user Saves this Forecast Sale, MultiCast will create one Master Forecast Sale with a specific ActvCode containing the total value of the Forecast (this is for easy Forecast reporting within GoldMine). As well, MultiCast will create up to 10 related Forecast Sales, each having their own user entered Actv-Code that will represent each of the possible 10 line items.

MultiCast can then be utilized as your Forecast Sale Management Center. We have many corporations utilizing our MultiCast application, and I think, if you have the need for it, that you will certainly like the ease of use that MultiCast offers. By using all of your GoldMines' configured information there is nothing to configure in MultiCast. It can't get any simpler than that. For more information on MultiCast visit www.DJHunt.US.

GoldBook

While we're on the subject of Invoices, and for those of you who are using QuickBooks for creating your Invoices etc., Computerese Incorporated has introduced yet another new add-on for GoldMine, that being GoldBook. Personally, I am very excited about this application, and I use it extensively myself. GoldBook permit me to use my QuickBooks for my day to day business operations, and will push any of that information into GoldMine for any records that have been linked via the GoldBook application. Additionally, GoldBook permits us to synchronize information between GoldMine and QuickBooks eliminating any need for double entry, hence, typos.

The GoldBook product was designed specifically to replace the defunct QBLink that used to be available from FrontRange, and was specifically designed to surpass the capabilities of QBLink. To that end, I am certain that we have surpassed the capabilities of the QBLink as witnessed by all of the reviews of the GoldBook product.

Specifically, my design goals were to:

- Eliminate the need to have the User Access Control turned on in any of the Vista operating system environments.
- Allow for the easy Linking of existing GoldMine/QuickBooks Customers/Vendors
- Allow for the easy Creation/Linking of New GoldMine records as QuickBooks Customers/Vendors
- Allow Accounts Perceivable Aging to be automatically brought over from QuickBooks into GoldMine fields.
- Allow QuickBooks Estimates/Sales Orders to be automatically brought over to GoldMine Pending/History which will display nicely in the GoldMine Preview windows.
- Allows QuickBooks Invoices/Sales Receipts/Purchase Orders to be brought over to GoldMine as History activities which will display nicely in the GoldMine Preview windows.

I think that I have met all of these design specifications in GoldBook, and here is a sample screenshot showing mismatched linked records between GoldMine and QuickBooks. From this screen you could easily update GoldMine or QuickBooks or both with the simple click of the **Synchronize Contact Fields** button.

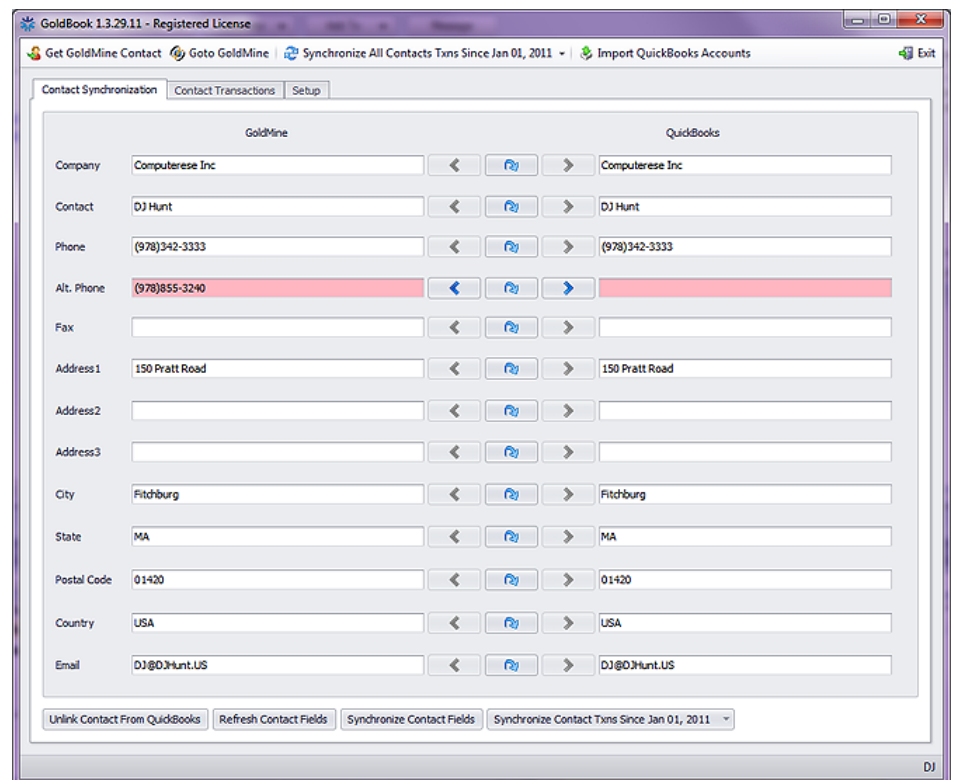


Figure 5-19

Since the original writing of this chapter, I have sold the source code for GoldBook to Chris Wettre of W-Systems. The new screenshot above, in Figure 5-19, is actually from the latest version of GoldBook as now is distributed by W-Systems. You can see that Chris has taken GoldBook to an entirely new and better level of integration between GoldMine QuickBooks, while making GoldBook more user friendly. Chris has taken my original GoldBook application to fruition, and for that I am most grateful.

