

In This Chapter

Contact1

Contact2

ContUDef

ContHist

ContSupp

ContGrps

Cal

Mailbox

Cases

GoldMine is comprised of a number of related tables. These tables are, for the most part, referentially linked in some way, although there are quite a few independent tables as well.

FrontRange has assured me that the table structures for this, the second release of GoldMine Premium 8.5.x, has significantly changed the data structure, and that I would need to rewrite this chapter in my books. FrontRange now ships SQL Server 2008 for Workgroups with GoldMine Premium 8.5.x. After review the tables covered in this chapter, I saw very few changes to the schema, although those that are there are certainly significant.

On the next few pages I will give you the table structures, their default indexes, and any relationships as they exist as of this writing against GoldMine Premium 8.5.1.12. FrontRange is constantly updating GoldMine, and although they try not to make structural changes, it occasionally becomes necessary for the inclusion of some new functionality (aka Notes). As I give you these structures, if I have any insight about the particular table, I will also impart that information to you.

In this chapter I will examine what I feel are the most prominent tables used within the GoldMine Premium product. At a later date or edition, I may decide to add more tables to this chapter.

Contact1

Contact1 Table Indexes

CN1RECID	RecID (Unique, Non-Clustered)
CONTACC	AccountNo, RecID (Non-Unique, Non-Clustered)
CONTCITY	u_City, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTCNTY	u_Country, u_State, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTCOMP	u_Company, u_Contact, RecID (Non-Unique, Non-Clustered)
CONTCURTAINING	Owner, Status, AccountNo (Non-Unique, Non-Clustered)
CONTKEY1	u_Key1, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTKEY2	u_Key2, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTKEY3	u_Key3, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTKEY4	u_Key4, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTKEY5	u_Key5, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTLAST	u_LastName, u_Contact, RecID (Non-Unique, Non-Clustered)
CONTNAME	u_Contact, u_Company, RecID (Non-Unique, Non-Clustered)
CONTPHON	Phone1, u_Contact, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTSTAT	u_State, u_City, AccountNo, RecID (Non-Unique, Non-Clustered)
CONTZIP	Zip, AccountNo, RecID (Non-Unique, Non-Clustered)

Relationships

Contact2.AccountNo	One-to-One	Contact1.AccountNo
ContHist.AccountNo	Many-to-One	Contact1.AccountNo
ContSupp.AccountNo	Many-to-One	Contact1.AccountNo
Cal.AccountNo	Many-to-One	Contact1.AccountNo
Mailbox.AccountNo	Many-to-One	Contact1.AccountNo
Cases.AccountNo	Many-to-One	Contact1.AccountNo
Notes.AccountNo	Many-to-One	Contact1.AccountNo

Structure

AccountNo	VarChar	20	Account Number - not Null*
Address1	VarChar	40	1 st Address Line
Address2	VarChar	40	2 nd Address Line
Address3	VarChar	40	3 rd Address Line
City	VarChar	30	City
Company	VarChar	40	Company Name
Contact	VarChar	40	Contact Name
Country	VarChar	20	Country
CreateAt	VarChar	5	Creation Time
CreateBy	VarChar	8	Created by UserID
CreateOn	DateTime	8	Creation Date
Dear	VarChar	20	Dear/Salutation
Department	VarChar	35	Contacts Department
Ext1	VarChar	6	1 st Phone Extension
Ext2	VarChar	6	2 nd Phone Extension
Ext3	VarChar	6	Fax Extension
Ext4	VarChar	6	3rd Phone Extension
Fax	VarChar	25	Fax Number
Key1	VarChar	20	Key 1
Key2	VarChar	20	Key 2
Key3	VarChar	20	Key 3
Key4	VarChar	20	Key 4
Key5	VarChar	20	Key 5
LastDate	DateTime	8	Last Modified Date
LastName	VarChar	15	Contacts Last Name
LastTime	VarChar	5	Last Modified Time
LastUser	VarChar	8	Last Modified by UserID
MergeCodes	VarChar	20	Record Merge Codes
Notes	Text	16	Legacy Notes
Owner	VarChar	8	Record Owner
Phone1	VarChar	25	1 st Phone Number - not Null
Phone2	VarChar	25	2 nd Phone Number
Phone3	VarChar	25	3 rd Phone Number
RecID	VarChar	15	Record ID - not Null
Secr	VarChar	20	Secretary
Source	VarChar	20	Lead Source
State	VarChar	20	State
Status	VarChar	3	GoldMine Status**
Title	VarChar	35	Contacts Title
U_City	VarChar	30	Upper Case City - not Null
U_Company	VarChar	40	Upper Case Company Name - not Null***
U_Contact	VarChar	40	Upper Case Contact Name - not Null
U_Country	VarChar	20	Upper Case Country - not Null
U_Key1	VarChar	20	Upper Case Key1 - not Null
U_Key2	VarChar	20	Upper Case Key2 - not Null
U_Key3	VarChar	20	Upper Case Key3 - not Null
U_Key4	VarChar	20	Upper Case Key4 - not Null
U_Key5	VarChar	20	Upper Case Key5 - not Null
U_LastName	VarChar	15	Upper Case Last Name - not Null
U_State	VarChar	20	Upper Case State - not Null
Zip	VarChar	10	Zip Code/Postal Code

*** The AccountNo field is comprised of**

Position 1 thru 6	Creation date in YYMMDD format (For Y2K compliance, positions 1 and 2 would store A9 for the year 2009)
Position 7 thru 11	Seconds since midnight
Position 12 thru 17	Randomly generated number
Position 18 thru 20	First three characters of Contact or Company name if Contact empty

**** The Status field is comprised of**

Position 1	U (US phone format) or I (International phone format)
Position 2	Curtain level (0 = None 1 = Partial 2 = Full 3 = Semi-Partial)
Position 3	1 indicates a record alert is active for the contact record

*****GoldMine employs a mirror field for indexed fields**

These mirrored fields are included in the Indexes which helps to speed up searches against a SQL backend, be it Microsoft or Firebird. These fields are auto-populated via a SQL Trigger with the upper case version of the contents in their parent field.

The **Contact1** table has one record for each primary contact in GoldMine. Both the **AccountNo** and the **RecID** are unique identifiers in the table with the **AccountNo** being a **Primary Key**, and the **RecID** being a **Candidate Key**. The **AccountNo** is used as the **Primary Key** in all of the relationships in other tables, while the **RecID** uniquely identifies each record within any table.

The **Contact1** table maintains 16 separate indices at all times in its default state. Each index is employed when the user is performing a lookup for a particular piece of information making that lookup, even against a large database, extremely fast. FrontRange has been asked, and often, to include more indices (see sidebar Note). They have determined that these 16 are an optimal number of indexes. Any more, and performance could take a hit. **CN1RecID** was the last index added.

There are seven primary relationships which have been identified for the **Contact1** table. There is the one-to-one relationship that exists between the **Contact1** table and the **Contact2** table. There should never be more than one record in the **Contact2** table for each record in the **Contact1** table. If the GoldMine administrator notices that there are multiple records in the **Contact2** table having the same **AccountNo**, then it behooves them to rectify the matter. This must be accomplished using 3rd party software to maintain synchronization awareness. Remember, in an earlier chapter, I gave you this SQL select statement to identify any **Duplicate Contact2** records:

```
select count(*), AccountNo
from Contact2
group by AccountNo having Count(*) > 1
```

Additionally, there exists a one-to-many relationship between the **Contact1** table, the **ContHist**, **ContSupp**, **Cal**, **Mailbox**, **Cases**, and the **Notes** tables. This means that for each **AccountNo** in the **Contact1** table there can be multiple records in these other tables possessing the same **AccountNo** which relates the records in both tables. You can have many historical activities against one contact record, as you could have many scheduled activities against one contact record. The **ContSupp** is a unique table that serves multiple purposes. I'll discuss that momentarily.

One thing that I would like to discuss here is the **Contact1.MergeCodes** field. The information contained in this field is employed when merging to document templates, and should not be confused with the merge codes that I will discuss later when I talk about the **ContSupp** table, and it's E-mail Address records.

Note

FrontRange has informed me that, unlike previous versions of GoldMine, users will be able to add their own indices through a **Firebird Database Manager**, or **SQL Server Management Studio**, and that GoldMine will pick up, and honor/use the new indices.

I would advise that you try not to add any user defined index, unless it is absolutely required, as any increase in the index count is bound to have an adverse affect on performance. You may want to add the index one at a time, and have your staff work with them for a while to assure that their GoldMine performance is still at an acceptable level. If acceptable, then add the next index that you need, and test the performance again for another period of time, and so on.

Contact2

Contact2 Table Indexes

CONTACT2	AccountNo (Non-Unique, Non-Clustered)
CN2RECID	RecID (Unique, Non-Clustered)

Relationships

Contact2.AccountNo	One-to-One	Contact1.AccountNo
---------------------------	------------	--------------------

Structure

AccountNo	VarChar	20	Account number - not Null
ActionOn	DateTime	8	Next Action on Date
CallBackAt	VarChar	8	Call back Time (unused)
CallBackOn	DateTime	8	Call back Date
CallBkFreq	SmallInt	2	Call back frequency
CloseDate	DateTime	8	Forecast Sale expected Close
Comments	VarChar	65	Comments
LastAtmpAt	VarChar	8	Last Attempt at Time
LastAtmpOn	DateTime	8	Last Attempt on Date
LastContAt	VarChar	8	Last Contacted at Time
LastContOn	DateTime	8	Last Contacted on Date
MeetDateOn	DateTime	8	Meeting on Date
MeetTimeAt	VarChar	8	Meeting at Time
NextAction	VarChar	65	Next Action
PrevResult	VarChar	65	Previous Results
RecID	VarChar	15	Record ID - not Null
UserDef01	VarChar	10	User defined field 1*
UserDef02	VarChar	12	User defined field 2*
UserDef03	VarChar	15	User defined field 3*
UserDef04	VarChar	12	User defined field 4*
UserDef05	VarChar	10	User defined field 5*
UserDef06	VarChar	10	User defined field 6*
UserDef07	VarChar	3	User defined field 7*
UserDef08	VarChar	10	User defined field 8*
UserDef09	VarChar	10	User defined field 9*
UserDef10	VarChar	10	User defined field 10*
UserDef11	VarChar	10	User defined field 11*
UserDef12	VarChar	10	User defined field 12*
UserDef13	VarChar	25	User defined field 13*
UserDef14	VarChar	10	User defined field 14*
UserDef15	VarChar	25	User defined field 15*
UserDef16	VarChar	10	User defined field 16*

*GoldMine User Defined Fields

GoldMine **UserDef01** thru **UserDef10** can never be deleted, whereas, **UserDef11** thru **UserDef16** you are permitted to delete. In all User Defined field cases you may not change the VarChar type, however, you are permitted to change the data size.

The **Contact2** table has only two indexes. The first, **Contact2**, is used to maintain the relationship between the **Contact1**, and the **Contact2** tables, however, because of its Non-Unique status it is possible to have two **AccountNos** exactly the same in the **Contact2** table (see WARNING in sidebar).

This table should contain one record for each record in the **Contact1** table, however, the reverse does not hold true. You could have fewer **Contact2** records than you have **Contact1** records. When one created a new **Contact1** record, a new **Contact2** record was not created automatically in past versions of GoldMine. The **Contact2** record was only created when there was a need to store information in fields contained in the **Contact2** table. This should not occur in new GoldMine Premium installations, but is possible if you upgraded to GoldMine Premium from an older build of GoldMine. Occasionally it is possible, although unlikely, to have a **Contact2** record without having a matching **AccountNo** in the **Contact1** table. These are called **Orphans**. When this occurs the GoldMine Administrator is advised to use a 3rd party software application to find, and remove these orphaned records. I did supply you with a SQL Query to identify these orphans, but here it is just in case:

■ Contact2 - Orphans

```
select AccountNo
from Contact2
where AccountNo not in
(select Accountno
from Contact1)
```

As you add user defined fields to your GoldMine, these new user defined fields will be contained in the **Contact2** table, while their structural definitions will be maintained in the **ContUDef** (see next page) table. The **ContUDef** table will reside in the same database as the **Contact2** table.

WARNING

The number of records in the **Contact2** table should never exceed the number of records in the **Contact1** table. You are advised to check the state of these two tables regularly using the queries:

```
select count(*)
from Contact1
```

```
select count(*)
from Contact2
```

ContUDef

ContUDef Table

Indexes

CONTUDEF	DBFName, Field_Name (Non-Unique, Non-Clustered)
CNURECID	RecID (Unique, Non-Clustered)

Structure

DBFName	VarChar	8	CONTACT1 or CONTACT2 - not Null
Field_Dec	SmallInt	2	Number of decimal places contained in the field
Field_Len	SmallInt	2	Field Length
Field_Name	VarChar	10	Field Name - not Null
Field_Pict	VarChar	20	Field Picture
Field_Type	VarChar	1	Field Type*
FieldDesc	VarChar	25	Field Description (Global Label)
FieldNo	VarChar	3	Field Number (Tab Order User Defined Fields)
FldOpts	VarChar	8	Field Options**
LocalLabel	VarChar	25	Local Label for the field
RAccess	VarChar	8	Read Access User or User Group
RecID	VarChar	15	Record ID - not Null
Status	VarChar	3	Field Status (Currently Unemployed)
WAccess	VarChar	8	Write Access User or User Group

* Field Type

- C - VarChar
- N - Numeric
- D - DateTime
- M - Text (Legacy Contact1.Notes)

** FldOpts

- 0 - Do not log changes
- 1 - Log changes

There is one record in the **ContUDef** table for each field in the **Contact1/Contact2** tables in any given contact database. The primary reason for including the **Contact1** fields in the **ContUDef** table is not to maintain the field definitions, as these are maintained in the data dictionary, but to allow for **Local Labeling** of the fields as well as the capability of maintaining the **Read/Write Access** properties which are also not maintained in the data dictionary.

ContHist

ContHist Table Indexes

CONTHIST	AccountNo, OnDate, RecID (Non-Unique, Non-Clustered)
CONTHUSR	UserID, sRecType, OnDate, RecID (Non-Unique, Non-Clustered)
CNHRLINK	LopRecID, OnDate (Non-Unique, Non-Clustered)
CNHLSDT	LastDate (Non-Unique, Non-Clustered)
CNHRECID	RecID (Unique, Non-Clustered)

Relationships

ContHist.AccountNo	Many-to-One	Contact1.AccountNo
ContHist.LOpRecID	Many-to-One	OpMgr.RecID
ContHist.LOpRecID	Many-to-One	Cases.RecID
ContHist.RecID	One-to-One	MailBox.LinkRecID
ContHist.LinkRecID	One-to-One	MailBox.RecID

Structure

AccountNo	VarChar	20	Linked Contact AccountNo - not Null
ActvCode	VarChar	3	Activity Code
CreateAt	VarChar	5	Created at Time
CreateBy	VarChar	8	Creation UserID
CreateOn	DateTime	8	Created on Date
Duration	VarChar	14	Duration***
Ext	VarChar	5	Notes characteristics html or plain
LastDate	DateTime	8	Last Modified on Date
LastTime	VarChar	5	Last Modified at Time
LastUser	VarChar	8	Last Modified by UserID
LinkRecID	VarChar	15	Linked MailBox Record ID
LOpRecID	VarChar	15	Linked Opportunity Manager Record ID/ Case Manager Record ID - not Null
Notes	Image	16	Notes
OnDate	DateTime	8	Activity on Date
OnTime	VarChar	5	Activity on Time
RecID	VarChar	15	Record ID - not Null
RecType	VarChar	10	Record Type*
Ref	VarChar	80	Reference****
ResultCode	VarChar	3	Result Code
SRecType	VarChar	1	1 st character of RecType - not Null
Status	VarChar	2	GoldMine field**
Units	VarChar	20	Units of a Forecasted Sale
UserID	VarChar	8	User - not Null

Note

Notice that the **Notes** field takes on new characteristics in GoldMine Premium as of 8.5.1.12.

Note

The **3rd byte** of the **RecType** field is used to store the **Success** (**space**) or **Unsuccessful** (**U**) for the activity.

The **4th byte** of the **RecType** field is used to store the **privacy** (**P**) value if the activity is marked as **Private**. The 4th byte may also be used when the **Completed** activity was the result of an **Auto Generate RSVP** (**R**) selection on the scheduled activity.

The **5th byte** of the **RecType** field is used to store any **color coding** for the activity as carried over when completing a **Scheduled Activity**.

* The following are possible values for the RecType field:

A Appointment	O Other	CO Outgoing call
C Phone call	S Sale	MG E-mail message
D To-do	T Next Action	MI Received e-mail
E Event	U Unknown	MO Sent e-mail
F Literature fulfillment	CC Call back	RO Audit Override
L Form	CI Incoming call	RS Audit New
M Sent message	CM Returned message	

** Status

1st character **Flag**
2nd character 1 if there are notes in the **Notes** field.

*** Duration

This takes the form of **hh:mm:ss** for most activities or it may be the string form of a number such as **[1.5]**.

**** Ref

The contact linked to the activity is stored at the end of the Ref field preceded by (oc:).
i.e. Sample for eBook (oc: DJ Hunt)

An additional note that should be conveyed, pertains to the **LOpRecID** field. Programmers should be aware that this field must contain a value and is set to **Not Null**. If there is no actual link to the **OpMgr/Cases** table, then the programmer is required to stuff this field with a space, plus the first **14** characters of a newly generated **RecID**.

Also, for the programmers among us, the **OnTime**, **CreateAt**, and **LastTime** fields all contain values based upon the 24 hour clock. Endusers may use the **FmtTime()** function (Appendix A) to display this value based on the 12 hour clock.

There may be many history records for any given **AccountNo** number, as well, there may be history activities that have no **AccountNo** value. History activities without an **AccountNo** value in the field are unlinked activities and may be considered orphans.

The **Duration** field may be **.null/empty**, may contain a character based numeric value (**1.25**), or may contain a time designation (**01:30:47**). All of these are acceptable values in this field. One could expect to see **357.25** in the Duration field if the **sRecType** field contained an **S**. On the other hand, if the **sRecType** field contained an **A** value, then the value in the **Duration** field could be, either, **1.25**, or **01:15:00**. as both are acceptable.

ContSupp

ContSupp Table Indexes

CONTSUPP	AccountNo, RecType, u_Contact, RecID (Non-Unique, Non-Clustered)
CONTSPFD	RecType, u_Contact, u_ContSupRef (Non-Unique, Non-Clustered)
CNTSUPADDR	RecType, u_Address1 (Non-Unique, Non-Clustered)
CNSRECID	RecID (Unique, Non-Clustered)
CSUPREFIDX	u_ContSupRef (Non-Unique, Non-Clustered)

Relationships

ContSupp.AccountNo	Many-to-One	Contact1.AccountNo
ContSupp.AccountNo	Many-to-Many	ContSupp.LinkAcct

Structure

AccountNo	VarChar	20	Linked Contact AccountNo - not Null
Address1	VarChar	40	Additional Contact Address 1
Address2	VarChar	40	Additional Contact Address 2
Address3	VarChar	40	Additional Contact Address 3
City	VarChar	30	Additional Contact City
Contact	VarChar	40	Contact Name/Detail
ContSupRef	VarChar	35	Reference
Country	VarChar	20	Additional Contact Country
Dear	VarChar	20	Dear/Salutation
Ext	VarChar	6	Phone Extension
Fax	VarChar	20	Fax Number
LastDate	Date/Time	8	Last Modified on Date
LastTime	VarChar	5	Last Modified at Time
LastUser	VarChar	8	Last Modified by UserID
LinkAcct	VarChar	20	Linked AccountNo
LinkedDoc	Text	16	Linked Document
MergeCodes	VarChar	20	Additional Contact or E-mail Merge Codes
Notes	Text	16	Notes
Phone	VarChar	20	Phone Number
RecID	VarChar	15	Record ID - not Null
RecType	VarChar	1	Record Type* - not Null
State	VarChar	20	Additional Contact State
Status	VarChar	4	GoldMine field**
Title	VarChar	35	Contact Title/Referrals account number
U_Address1	VarChar	40	Upper Case Address1*** - not Null
U_Contact	VarChar	40	Upper Case Contact Name*** - not Null
U_ContSupRef	VarChar	35	Upper Case Reference*** - not Null
Zip	VarChar	10	Additional Contact Zip/Postal Code

* The following are possible values for the RecType field:

A	Record Alerts	L	Linked document
C	Additional contact record	O	Relationship Tree
E	Automated process attached event	P	Profile record/extended profile record
H	Extended profile header	R	Referral record
V	CS_Version		

** Status

- 1st byte **HTML On/Off**
- 2nd byte 1 notes in the **Notes** field
- 3rd byte **Wrap On/Off**
- 4th byte **MIME On/Off**

***GoldMine employs a mirror field for indexed fields

These mirrored fields are included in the Indexes which helps to speed up searches against a SQL backend, be it Microsoft or Firebird. These fields are auto-populated via a SQL Trigger with the upper case version of the contents in their parent field.

Now we have the motherload of tables. Not only can there be a 1 to many relationship between the **Contact1** table and this, the **ContSupp** table, but it can be a 1 to many relationship many times based on the **ContSupp.RecType** value. You see, the **ContSupp** table is used for many different record types which each have their own display area within GoldMine. Over the following pages I will discuss the various **ContSupp** record settings based on the various **ContSupp.RecType** values as shown above. You will see that the same field may be used differently based on the **ContSupp.RecType** value.

Record Alert

Each field in the **ContSupp** table is capable of holding a variety of information, and again, it is all dependant upon the **ContSupp.RecType** field value. The first **ContSupp.RecType** that I will examine is that of **A**. When a **Record Alert** is attached to a **Contact** record, the actual alert for that contact is stored in the **ContSupp** table, while the characteristics of the alerts are stored in the **InfoMine** table (not to be covered in this book), and better known to most as the **Knowledge Base**. In the **ContSupp** table itself, the record pointer to the alert stores very little information. These represent the fields that were populated for a typical alert:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **A** for this type of record.

Add'l Contacts

Recommendation

*This author does not recommend that any organization utilize the **Additional Contacts** option within GoldMine. I always ask my clients to create separate records for each **Contact**, and **Contact Location** (i.e. **Home** or **Office**). If there is any relationship between these records I ask that one use the **Relationship Tree**, in GoldMine Premium, to graphically represent these relationships.*

Note

*Please remember that the **Contact1.MergeCodes** field is for merge codes that pertain to **Word Document Template** merges, and should not be confused with the **ContSupp.MergeCodes** field that is associated with, and used for the **E-mail Template** merges (discussed later in this chapter).*

Automated Process Tracks

Notes - contains the **Alert Code**, the **UserID** and any triggers. In example:

~TST~DJ+chr(9)+1+chr(10)

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

These **Additional Contacts** should not, in any way, be confused with the main or **Primary Contacts**, but, instead, are related to the **Primary Contact** as **Additional Contacts**. These records contain a **ContSupp.RecType** of **C**. For this record type all of the fields are, pretty much, used as the field labels would indicate. The few exceptions are:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **C** for this type of record.

ContSupRef - will hold any information supplied through the **Ref** field in the dialog form when **Adding/Editing** an **Additional Contact** record. GoldMine also utilizes this field when it is creating an **Additional Contact** from a deletion of a merged **Contact** record. In this case, GoldMine would populate this field with the value **Duplicate Record**.

Status - 1st character = **1** if there is an associated/related **E-mail Address** which would also be contained within this table as another record with a **ContSupp.RecType** of **P**. Once populated, the **ContSupp.Status** field will remain populated, and the bytes will be turned on (**1**) or off (**0**) as necessary. However, the **ContSupp.Status** field will remain **.null**, until such time as either an **E-mail Address** is added, or a note is added in the **ContSupp.Notes** field. The 2nd character = **1** if there is an associated note or it could have a **0**, once the field has been populated, if there are no notes associated with the particular record.

MergeCodes - This field is used for merge codes that are to be applied against Additional Contacts when merging to word document templates.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

Continuing on, and in alphabetical order, the **RecType** of **E** would be next. Whenever an **Automated Process Track** is attached to a **Contact** record, a **ContSupp** record of this type is created to maintain the current state of any attached Track. There are a few fields in the record that maintain information.

AccountNo - stores the relational link back to the appropriate record in the Contact1 table.

RecType - will always be **E** for this type of record.

Contact - maintains the Automated Process attachment date & time information, and takes the form of:

YYYYMMDDHH:MM:SS (2009091509:35:19)

Title - stores the **Tracks.TrackNo** field information of the related track.

ContSupRef - will maintain the current position of the track within the process, or, probably better understood as the next **Sequential Event** that is to be tested. The information for this value is drawn from the **Tracks.Name** field.

Phone - this field maintains the **Tracks.Options** flags for the specified track. The information is a set of 5 digits, in a character based field, representing the state of the flags. In example:

10001

Ext - maintains the **Track.EventNo** for the **Next Event**.

LinkAcct - maintains the **Last Event** date in the form of:

YYYYMMDD (20090915)

Address1 - is used to maintain the **Tracks.Name** from the parent track.

City - maintains the **Next Event** (**Tracks.Name**), in English, for the **Next Event**.

State - maintains the **Track.EventNo** for the **Next Event**.

Zip - maintains the **UserID** of the individual who attached this **Track** to the **Contact** record.

Headers

Note

The double asterisk (**) Instructs GoldMine to not display the field on the dialog form when displayed for this **Detail** type. Anything else following the equal sign (=) Indicates that the field is to be displayed and that this information is to be displayed as the label for the equivalent field.

I would add that **Details Plus** exposes more **ContSupp** fields to be used for **Details**, as well as being capable of allowing you to design the layout of those fields on the screens. Additional information can be found at:

<http://www.Solica.com>

Linked Documents

Country - is used to maintain the **Trigger** basis (**Tracks.TriggerTyp**), in English, for the **Next Event**.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

Next we should look at the **RecType** of **H**. This record type is where the **Setup** information is maintained for the various **Details** for which a user creates, and for which extended information has been established. GoldMine looks at this record to ascertain which fields, of the extended fields, to display, and what is to be displayed as their label.

AccountNo - stores the relational link that is used by **ContSupp P RecType** to relate the setup criterion for the **Detail**. This might take the form of:

HxD5UZ1F8+#

RecType - will always be **H** for this type of record.

ContSupRef - will contain the name of the **Detail**, up to 20 characters along with the final 15 characters being allocated to the tab name if one has been established for this particular **Detail**. In example:

Partner	Part&ner
Handango	MyDiabetes
Credit Card	Credit Card
GoldMine	GoldMine

Phone - this field appears to always carry a **0**, and I have yet to determine its significance.

Notes - is where the extend field label association is maintained, and contains the true field name along with its local label for all eight of the extended detail fields utilized by GoldMine Premium. In example from the GoldMine **Detail** shown above:

```
Title =HDA#
LinkAcct=PoR
Country =**
Dear =**
Fax =**
Zip =**
Ext =**
State =Expiration
MergeCodes=**
Address1=Serial #
Address2:
Address3=**
SaveCols=1
```

All information preceding the equal sign (=) consists of exactly 8 characters except for the **MergeCodes** label which is, obviously, 10 characters. Prior to GoldMine Premium 8.5.1.12 only the **Title**, **LinkAcct**, **Country**, **Zip**, **Ext**, **State**, **Address1** & **Address2** fields could be utilized in the **Extended Details** for a total of eight extended fields. Today we have the capability of 12 extended fields within a GoldMine **Detail** with the last entry telling GoldMine whether or not to save the user defined column settings for that **Detail** type.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

I can now look at the **RecType** of **L**, and this record type is utilized for the **Linked Documents**.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **L** for this type of record.

Contact - this field contains 10 spaces, the document extension type, 8 spaces, and the link date. In example:

```
Expression: padr(UserID, [ ], 10) + document extension + space(8) + dtos(date())
Value: [DJ .dot 20090930]
```

Title - holds the document type. Based on the information in the **Contact** field example above, this document type might be **Microsoft Office Word 97 - 2003** as this is the association that my computer holds for a .dot extension.

ContSupRef - contains the descriptive **Document Name**: entered when the link was created or it might be the **Subject**: of the E-mail from which the linked document was derived.

Dear - this field maintains the link document creation date & time information, and takes the form of:

Expression: **YYMMDDHH:MM:SS**
Value: **2007031610:43:03**

LinkAcct - is used when the linked document is the result of a link to an e-mail received, and contains the **MailBox.ReclD** of the linked e-mail message, otherwise, this field will probably remain **.Null..**

Address1/Address2/Address3 - these fields carry the physical document name, and the path to that document. Remembering that each field is 40 characters, **Address1** takes characters **1** through **40** of the path while **Address2** takes characters **41** through **80** of the path, with any remaining characters being dumped into the **Address3** field.

City - this field is broken up into 3 concatenated sections. The first section is the eight character UserID. If the UserID is DJ, 2 characters, then this first section will be DJ plus six spaces. The next section contains the string representation of the date that this record was created. This might look like 20070327. The next section, 7 characters long, is the creation time of this record. It is based on a 12 hour clock, and it includes a space if it is not a 2 digit hour. Here are a couple of possible representations of a value from this field:

DJ 20090927 9:31am
DJ 2009092710:49am

Status - this is, again, a flag field. This first byte may indicate that this is an attachment (**A**). The second byte is set to **1** to indicate that there are notes associated with this record or set to **0** when there are no notes associated with this record.

LinkedDoc - this field contains most of the sync information about the file, plus the actual file that is launched when instructed to do so by a user. The information in this field could look like:

~~SYNC=1
~~CREATETIME=200803120721
~~FILENAME=Y:\goldmine\mail box\ldhunt\DJ\2008\03\image001.jpg
~~SYNCSTAMP=200803120721

LastUser, LastDate, LastTime, and ReclD - are all employed in the normal manner.

Note
Please note that each of the lines in the **ContSupp.LinkedDoc** field is followed by **chr(10)**, and not the normal **chr(13)+chr(10)**. Additionally, and most importantly for you programmers out there, the last line must end with a **chr(10)**.

Relationship Tree

Note
Except in rare circumstances, I always remove the link between the **Books** (Top Level), and **Folders** (Sections) to any given contact record. Usually, I only link the **Contact** (Page) to a **Contact** record. In the sample **Relationship Tree** that ships with GoldMine Premium, I noticed that FrontRange follows this paradigm as well.

Based on the **Relationship Tree** shown to the right in Figure 8-1, I think that you can easily see why I do not link the **Books** and **Folders**.

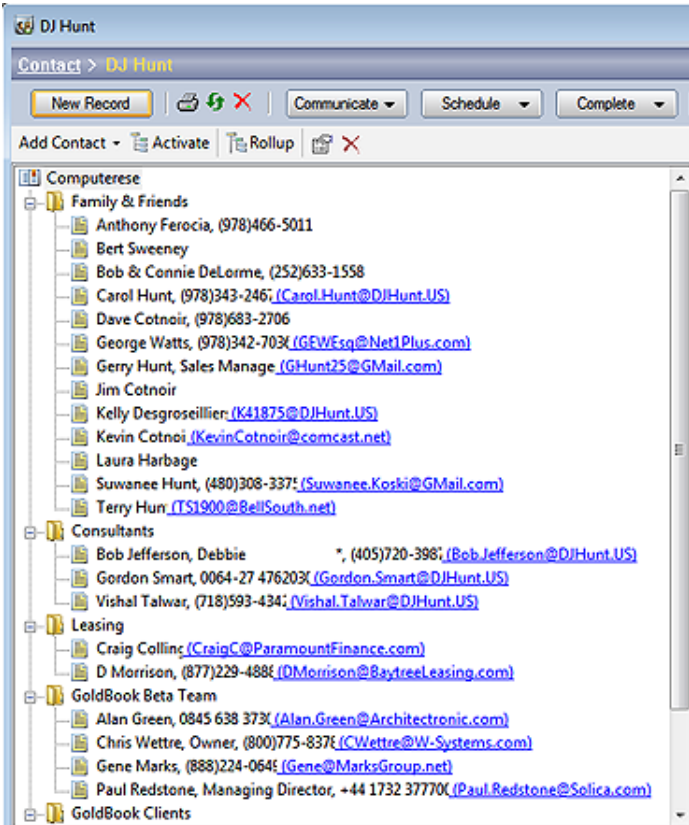


Figure 8-1

The next **RecType** that I will examine with you is that of **O** for the **Relationships Tree** record. This represents yet another totally different usage of each of the fields in the **ContSupp** table. Before I examine the fields for this type of **RecType**, and what they contain, I would refer you to Figure 8-1. Figure 8-1 is a **Relationship Tree** that I created against the my GoldMine database. Using this **Relationship Tree** as the basis for the information contained in the various **ContSupp** fields, let's examine it in detail.

AccountNo - if this field is populated, it represents a link between the **Relationship Tree** entity, and a particular contact record.

RecType - these will always have a type of **O**.

Note

*It is because of the 35 character limitation of the **ContSupRef** field that FrontRange imposes a 5 **SubSection** limitation for the **Relationship Tree**. Based on the schema that they have developed this is the maximum that the schema will permit.*

Contact - will contain a uniquely generated **AccountNo** for the **Book** (top level), and all entities contained within this book will carry this value in the **ContSupp.Contact** field.

ContSupRef - contains the sort order for the **Relationship Tree**. The data in this field will be a numeric value representing a **Book** (top level), or **Folder** (section), or a numeric value, and as much of the contact name as the 35 character limit of this field will permit. This will vary with the number of SubSections that may be involved.

For the **Relationship Tree** shown in Figure 8-1 on the previous page, these are the values contained within the **ContSupRef** field sorted as they would be for the **Relationship Tree** ordering:

```

10000
1000001000
1000001000 Anthony Ferocia
1000001000 Bert Sweeney
1000001000 Bob & Connie DeLorme
1000001000 Carol Hunt
1000001000 Dave Cotnoir
1000001000 George Watts
1000001000 Gerry Hunt
1000001000 Jim Cotnoir
1000001000 Kelly Desgroseilliers
1000001000 Kevin Cotnoir
1000001000 Laura Harbage
1000001000 Suwanee Hunt
1000001000 Terry Hunt
1000001500
1000001500 Bob Jefferson
1000001500 Gordon Smart
1000001500 Vishal Talwar
1000002000
1000002000 Craig Colling
1000002000 D Morrison
1000002500
1000002500 Alan Green
1000002500 Chris Wettre
1000002500 Gene Marks
1000002500 Paul Redstone
1000003000
1000003000 Damian Schwarz
1000003000 Debbie Green
1000003000 Eric Soloff
1000003000 Gary Zimmerman
1000003000 Rick Kabra
    
```

Ext - this field holds the icon type to be displayed in the **Relationship Tree**. If this is the record for the **Book** level, then this field will contain a 1. If this record is for a **Section** level, then this field will hold a 3. Whereas, if this record is for a **Contact** level it will contain a 6.

Address1 - contains the **Contact** name of the record as the **Contact** name appears in the **Relationship Tree**. There is one critical addition to this statement. The name is entered into this field preceded by 2 spaces for each level of the tree. **Contact** records under a section are considered to be at the same level as the section. Let's look more closely at this. Specifically I will look at the information contained in this field for the record pertaining to **Consultants**, and **Bob Jefferson**. Both of these would be considered to be at level 2, therefore, in this field both **Consultants**, and **Bob Jefferson** will be preceded by 4 spaces (2 spaces * 2 levels = 4 spaces).

City - this field is broken up into 3 concatenated sections. The first section is the eight character **UserID**. If the **UserID** is **DJ**, 2 characters, then this first section will be **DJ** plus six spaces. The next section contains the string representation of the date that this record was created. This might look like **20091027**. The next section, 7 characters long, is the creation time of this record. It is based on a 12 hour clock, and it includes a space if it is not a 2 digit hour. Here are a couple of possible representations of a value from this field:

```

DJ   20091027 9:31am
DJ   2009102710:49am
    
```

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

Next I will examine the **ContSupp** table fields when the **RecType** is **P**. This single record type contains three subsets of possible record types. Let's start by assuming that the first record type that I am examining is one that is an **E-mail Address** for the main contact record. For this type of record most of the fields in this table are not employed.

Note

*Please note that additional information displayed is the **Relationship Tree** is not stored nor maintained in the record, and is, however, generated by GoldMine for the graphical display of the tree at each change of the **Contact** record.*

E-mail Address

Those fields that are employed, and their usage are:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **P** for this type of record.

Contact - will always contain **E-mail Address**, upgrades from previous versions might still have **Internet Address** as a value in this field.

ContSupRef - will contain the first **35** characters of the actual **E-mail Address**.

Notes - this text field will contain any characters that were entered into the **Notes** are of the dialog form for the **E-mail Address** at the time that said **E-mail Address** was created or after it was been modified.

Address1 - this field will contain any characters from the actual **E-mail Address** that exceeded the **35** character limitation of the **ContSupp.ContSupRef** field. As this field is **40** characters long, the longest **E-mail Address** that GoldMine is capable of handling is **75** characters.

Address2 - this field contains the **Contact** name associated with the **E-mail Address**.

City - this field is broken up into **3** concatenated sections. The first section is the eight character **UserID**. If the **UserID** is **DJ**, **2** characters, then this first section will be **DJ** plus six spaces. The next section contains the string representation of the date that this record was created. This might look like **20091027**. The next section, **7** characters long, is the creation time of this record. It is based on a **12** hour clock, and it includes a space if it is not a **2** digit hour. Here are a couple of possible representations of a value from this field:

```
DJ 20091027 9:31am
DJ 2009102710:49am
```

Zip - this is a flag field, and there are four possible flags. The first byte indicates whether this **E-mail Address** can use **Rich Text (HTML)**. The second byte indicates whether this is a **Primary E-mail Address** or not. In theory only one **E-mail Address** per **AccountNo** may have this flag set to **1**, indicating **Primary E-mail Address**. The third byte indicates the **MIME** capability of this **E-mail Address**, while the fourth, and the final byte, indicates a users decision to **Wrap Lines** or not.

MergeCodes - this field contains merge codes that are employed when doing an E-mail merge based on codes.

Status - I only mention this field here as it had relavence in previous versions of GoldMine. It remains **.null**. in this, the GoldMine Premium, version of GoldMine. In fact, of my **7814 E-mail Address** records, only **31** have a value in the **ContSupp.Status** field.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

As an addition to the last record type discussed, I will examine the fields when the **RecType** is still **P**, but this is an E-mail Address for an Additional Contact. All of the fields mentioned above pertain to this grouping with the addition of one field.

LinkAcct - this field contains the **RecID** from the linked **ContSupp** having a **RecType** of **C** (**Additional Contact**) record. This value will be something like:

```
4A7Y8PC,2+,$FJ:
```

Now let's examine the fields when the **RecType** is **P**, but this record type is a **Web Site** associated with the **Primary Contact**. The fields employed are:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **P** for this type of record.

Contact - will always contain the value **Web Site**

ContSupRef - this field will contain the URL to the web site.

Phone - could contain a **10** character link ID that is used in linking the **ContSupp H RecType** record, as well as for linking to lookup values in the **Lookup** table.

Notes - this text field will contain any characters that were entered into the **Notes** of the **Web Site** at the time that the **Web Site** was created or after it was edited. In addition, the **Notes** field takes

Tip

Many users confuse the **ContSupp.MergeCodes** field with the **Contact1.MergeCodes** field. The two are separate and distinct, and have totally different functionality. For instance, one can create a **Filter** using the **Contact1.MergeCodes** field, whereas, this is not possible using the **ContSupp.MergeCodes** field.

WebSite

Note

In some older versions of GoldMine, one used to be able to enter extra long Web Site addresses into the Notes field of the ContSupp record, and the URL could navigate to it from there. In GoldMine Premium, however, you are again allowed to enter exceedingly long Web Sites. Refer to the Notes definition.

on special characteristics if the **Web Site** exceeds the **35** characters of the **ContSupp.ContSupRef** field.

You may expect it to look something like this:

~~REF=www.Long_Web_Site.com"~~NOTES=These are the actual Notes"

City - this field is broken up into **3** concatenated sections. The first section is the eight character **UserID**. If the **UserID** is **DJ**, **2** characters, then this first section will be **DJ** plus six spaces. The next section contains the string representation of the date that this record was created. This might look like **20091027**. The next section, **7** characters long, is the creation time of this record. It is based on a **12** hour clock, and it includes a space if it is not a **2** digit hour. Here are a couple of possible representations of a value from this field:

DJ 20091027 9:31am
DJ 2009102710:49am

Zip - this is a flag field, and there are four possible flags. The first byte is set to **0**. The second byte indicates whether this is a **Primary Web Site** or not. In theory only one **Web Site** per **AccountNo** may have this flag set to **1**, indicating the **Primary Web Site** for the related **Contact** record. The third byte, and the fourth byte is set to **0**. This field may or may not be populated.

Status - I only mention this field here as it had relevance in previous versions of GoldMine. It remains **.null** in this, the GoldMine Premium, version of GoldMine. In fact, of my **7814 E-mail Address** records, only **31** have a value in the **ContSupp.Status** field.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

And, finally, we examine the fields when the **RecType** is **P**, and this record is a default or user defined **Detail** type. The fields employed, and their usage are:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **P** for this type of record.

Contact - will contain the **Detail** name. i.e.:

Computer
Serial Number
Credit Card

ContSupRef - this field will contain the **Reference** information as supplied by the user creating this specific **Detail**, and, should the field contain a lot of repetitive entries, it is best to use the **F2 Lookup List** with the **Auto-Fill** option turned on.

Title - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Dear - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Phone - will contain a **10** character link ID that is used in linking the **ContSupp H RecType** record, as well as for linking to lookup values in the **Lookup** table.

Ext - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Fax - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

LinkAcct - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Notes - this field contains any notes associated with this **Detail**.

Address1 - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Address2 - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Detail

Note

All Detail records were formally known as Profile records in earlier versions of GoldMine. For legacy purposes, a Detail record maintains the RecType of P in all versions of GoldMine.

Note

As of GoldMine Premium 8.5.1.12 GoldMine, itself, exposes 12 Cont-Supp fields for Extended Detail field usage. This is better than the previous 8 that were being exposed within, but still not as good as Details Plus from Solica which exposes 13 fields, and additionally permits you to distinctly design a dialog form for each Detail that you define.

Address3 - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

City - this field is broken up into 3 concatenated sections. The first section is the eight character **UserID**. If the **UserID** is **DJ**, 2 characters, then this first section will be **DJ** plus six spaces. The next section contains the string representation of the date that this record was created. This might look like **20091027**. The next section, 7 characters long, is the creation time of this record. It is based on a 12 hour clock, and it includes a space if it is not a 2 digit hour. Here are a couple of possible representations of a value from this field:

DJ 20091027 9:31am
DJ 2009102710:49am

State - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Zip - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Country - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

MergeCodes - this field can be defined by the user as an extended field for this **Detail** within GoldMine under the **Info** tab.

Status - the 1st byte could be either 0 or 1, and for a **Detail** record other than **E-mail Address** or **Web Site**, would appear to have no relevance. While the 2nd byte, which could also be either 0 or 1, indicates that there are not or are notes respectively in the **Notes** field.

LinkedDoc - this field is not used for this type of detail within GoldMine, but is exposed for usage by add-on products like **Details Plus** by **Solica**. This adds another memo field to the users repertoire when exposed by **Details Plus**.

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

The next to the last record type that I will examine for the **ContSupp** table is the **RecType** is **R**, and this record type is for a **Referral** type of record. One must remember that two records are created for each **Referral** created. The fields employed and their usage are:

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table.

RecType - will always be **R** for this type of record.

Contact - Referral record A: "For: "+ **Contact1.Company**
- Referral record B: "To: "+ **Contact1.Company**

Title - Referral record A: **Contact1.AccountNo** of the referred **For:** company
- Referral record B: **Contact1.AccountNo** of the referred **To:** company

ContSuppRef - this field will contain the Reference information as supplied by the user creating the Referral.

Ext - Referral record A: **R**
- Referral record B: **T**

LinkAcct - Referral record A: **Contact1.RecID** of the referred **For:** company
- Referral record B: **Contact1.RecID** of the referred **To:** company

Notes - if applicable, whatever notes were applied to both of the **Referral** records.

City - this field is broken up into 3 concatenated sections. The first section is the eight character **UserID**. If the **UserID** is **DJ**, 2 characters, then this first section will be **DJ** plus six spaces. The next section contains the string representation of the date that this record was created. This might look like **20091027**. The next section, 7 characters long, is the creation time of this record. It is based on a 12 hour clock, and it includes a space if it is not a 2 digit hour. Here are a couple of possible representations of a value from this field:

DJ 20091027 9:31am
DJ 2009102710:49am

Referral

Recommendation

*In my opinion, this type of **ContSupp** record is maintained for legacy purposes only. I always recommend that my clients do no use the **Referrals** tab, but, instead, remove it from view using the **User Override** as I discussed in Chapter 3 of this book.*

*I always recommend the creation of one **Contact** record for each contact, and that you set up any referential links using the **Relationship Tree**. The power of the **Relationship Tree** is usually under utilized by the end user. While the lack of power encompassed by the information under the **Referrals** tab is usually embraced by the end user.*

As my wife would say: "Go Figure".

Version

State - this field is used, in this case, as a flags field. There are **13** possible flags. When the user creates a **Referral** there is a **Status** tab, see Figure 8-2, and on this tab are **13** options. Each byte in the **State** field represents one of these options. The **State** field should be identical for both **Referral** records.

Status - this is a flag field as well. The 1st byte is set to **0**, but only if a note is added to trip the 2nd byte. The **0** is not removed from the 1st byte once it is established. When the 2nd byte is set to **1** this indicates that there are notes associated with these records.

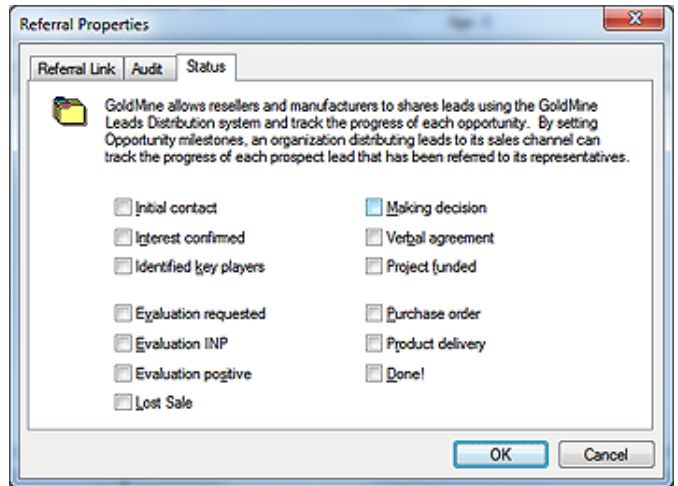


Figure 8-2

LastUser, **LastDate**, **LastTime**, and **RecID** - are all employed in the normal manner.

Now let's examine the fields when the **RecType** is **V** which appears to be a new **RecType** as of GoldMine Premium 8.5.1.12 or I simply missed it for my previous books. This record type appears to be a **Version** control, and there is but a single record for this **RecType**. The fields employed are:

AccountNo - contains a fixed value **CS_Version** which I can only assume stands for **ContSupp Version**. That does seem a reasonable assumption, does it not?

RecType - will always be **V** for this type of record.

Contact - will always contain the version number which, in my case, was **8.10.90514**.

All of the rest of the **ContSupp** fields are either **.null**. or empty except for **U_Contact** and, of course the **RecID** which can never be **.null**. or empty.

ContGrps

ContGrps Table Indexes

GROUPACC	AccountNo, UserID (Non-Unique, Non-Clustered)
GROUPNO	UserID, U_Code, RecID (Non-Unique, Non-Clustered)
GRPRECID	RecID - Unique (Unique, Non-Clustered)

Relationships

ContGrps.Accountno	Many-to-One	Contact1.AccountNo
---------------------------	-------------	---------------------------

Structure

AccountNo	VarChar	20	Linked Contact AccountNo - not Null
Code	VarChar	8	Code Assigned to the Group
RecID	VarChar	15	Record ID - not Null
Ref	VarChar	24	Reference
U_Code	VarChar	8	Upper Code - not Null
UserID	VarChar	15	User - not Null

The **CONTGRPS** table stores the names of groups, and the individual group members, so records have two formats.

For **Group Header** records the contents are:

UserID – the **UserID** of the owner of the **Group** or blank for a **(public) Group**.

Code - the code given to the group at the time of creation or when modifying a **Group**.

AccountNo – for **Header** records is always ***M** or ***MS** followed by the number of members in the group, right justified to utilize the full **20** character space. The **S**, if present, shows this is a group that is permitted to synchronize. For Example: a value of ***M 282** shows a non-synchronizing group possessing 282 members.

Ref – the name of the **Group** assigned at the time of the creation of the **Group**.

RecID – normal meaning.

For **Group Member** records the contents are:

UserID – the **RecID** of the associated **Group Header** record. It is **ContGrps.UserID** which links all the members to a particular **Group**.

Code – the sort order of the **Group** as selected (1st eight characters) when the **Group** was created or the **Member** added.

AccountNo – the **Accountno** of the **Contact** record of the **Group Member**.

Ref – the **Reference** of the **Group** as selected when the **Group** was created or the **Member** added.

It is helpful to know the **ContGrps** structure when writing SQL queries. For example the query:

```
select *
from ContGrps
where UserID='DJ'
and Ref = 'Customers'
```

Could return the **Group Header** record for this group as:

Row	UserID	Code	Accountno	Ref	Recid
1	DJ	DJH	*M 3024	Customers	CIN12R9)<1+IR#Y

where as this query would show the individual members:

```
select *
from ContGrps
where UserID in (select RecID
from ContGrps
where UserID='DJ'
and Ref = 'Customers')
```

This could be useful for more sophisticated queries, for example combining group membership with other criteria.

Note
Occasionally the **Group Header** record could possess the wrong count. This can be fixed by cloning the group. The new group should have the correct count. Cloning is also useful to change a Non-Syncing Group into a Syncing Group and vice versa.

Cal

CAL Table Indexes

CAL	RecType, UserID, OnDate, OnTime, RecID (Non-Unique, Non-Clustered)
CALALARM	AlarmFlag, UserID, AlarmDate, AlarmTime (Non-Unique, Non-Clustered)
CALCDCINST	CalDef_ID, AccountNo, Is_Exception, Orig_Date, Orig_Time (Non-Unique, Non-Clustered)
CALCDEX	CalDefEx_ID (Non-Unique, Non-Clustered)
CALCDINST	CalDef_ID, Orig_Date, Orig_Time (Non-Unique, Non-Clustered)
CALCDUINST	CalDef_ID, UserID, Is_Exception, Orig_Date, Orig_Time (Non-Unique, Non-Clustered)
CALCONT	AccountNo, RecType, OnDate, OnTime, RecID (Non-Unique, Non-Clustered)
CALDATE	UserID, OnDate, OnTime, RecID (Non-Unique, Non-Clustered)
CALENDDATE	EndDate (Non-Unique, Non-Clustered)
CALLINKRECID	LinkRecID (Non-Unique, Non-Clustered)
CALPROB	RecType, UserID (Non-Unique, Non-Clustered)
CALRECID	RecID (Unique, Non-Clustered)
CALRLINK	LOPRecID, RecType, OnDate, OnTime (Non-Unique, Non-Clustered)

Relationships

Cal.AccountNo	Many-to-One	Contact1.AccountNo
Cal.RecID	One-to-One	MailBox.LinkRecID
Cal.LinkRecID	One-to-One	MailBox.RecID
Cal.LOpRecID	Many-to-One	OpMgr.RecID
Cal.LOpRecID	Many-to-One	Cases.RecID
Cal.Def_ID	One-to-One	CalDef.RecID

Structure

AccountNo	VarChar 20	Linked Contact AccountNo - not Null
AConfirm	VarChar 3	Meeting Confirmation Flag
ActvCode	VarChar 3	Activity Code
AlarmDate	DateTime 8	Alarm on Date
AlarmFlag	VarChar 1	Alarm Flag - not Null
AlarmTime	VarChar 5	Alarm on Time - not Null
ApptUser	VarChar 10	Meeting Confirmation User/Color Flag Byte 10
Attendees_Ex	VarChar 100	Refer to sidebar Note
CalDef_ID	VarChar 15	Refer to sidebar Note
CalDefEx_ID	VarChar 15	Refer to sidebar Note
Company	VarChar 60	Company/Contact Name associated w/activity
CreateAt	VarChar 5	Created at Time
CreateBy	VarChar 8	Creation User
CreateOn	DateTime 8	Created on Date
DirCode	VarChar 10	DirCode for the Contact Set
Duration	SmallInt 2	Duration/Probability
EndDate	DateTime 8	Activity Scheduled End Date
Ext	VarChar 5	Used to code format (html or plain)
Flags	Ineger 16	Refer to sidebar Note
Is_Exception	SmallInt 2	Refer to sidebar Note
LastDate	DateTime 8	Last modified on date
LastTime	VarChar 5	Last modified at time
LastUser	VarChar 8	Last modified by user
LDocRecID	VarChar 15	Reserved for possible future use
LinkRecID	VarChar 15	Linked Record ID - not Null
LOpRecID	VarChar 15	Linked Opportunity Record ID - not Null
Notes	Image 16	Notes
Number1	Float 8	Sales Potential Amount
Number2	Float 8	Unit in a Forecasted Sale
OnDate	DateTime 8	Activity on Date
OnTime	VarChar 5	Activity on Time - not Null
Orig_Date	DateTime 8	Refer to sidebar Note
Orig_Time	VarChar 5	Refer to sidebar Note
RecID	VarChar 15	Record ID - not Null
RecType	VarChar 1	Record Type - not Null*
Ref	VarChar 80	Reference
RSVP	VarChar 1	RSVP Notification Flag
Service	Text 16	Notes
Status	VarChar 4	GoldMine field**
UserID	VarChar 8	User - not Null

Note

In a sampling of 500 Calendar Activities from my production database these fields were not utilized as of this writing:

```
Cal.Attendees_Ex
CalDef_ID
CalDefEx_ID
Cal.Flags
Cal.Is_Exception
Cal.Orig_Date
Cal.Orig_Time
```

Note

Please notice that **Cal.Notes** is now an **Image** based field, and in the normal SQL Query, unreadable. Use this as part of your select statement to make these notes readable:

```
select cast(cast(NOTES as var-
binary(max)) as varchar(max)) as
Notes
```

WARNING

The accuracy of the **Peg Board** is dubious, and should never be relied upon for user accountability. If GoldMine is closed improperly or closed using the **X**, then the **Peg Board** is not updated. There are many other instances that could cause incorrect data in the **Peg Board**.

**Appointments,
Calls, Other
Actions, Next
Actions**

Note

This has been asked many times over, so I will answer it for you here, the **Activity** field contains a hardcoded (fixed) list to which you may neither add to nor remove from list items. This field is the string representation of the **RecType** for the particular activity being scheduled.

* The following are possible values for the **RecType** field:

A Appointment	E Event	O Other Activity
B Occasions	F Literature Fulfillment	Q Queued E-mail/Quota
C Phone call	H Holidays	S Sales Potential
D To-do	M Message	T Next Action

** Status

- 1st character **Flag**
- 2nd character 1 if there are notes in the **Notes** field.

This table holds all of the **Pending** activities for **Contacts**, and **Users** alike. **Cal** table records can be broken up by **RecType** as we have seen in the other GoldMine tables. The same holds true for this table as in the **ContHist** table, activities with the **AccountNo** field populated are linked activities. Some of the activities with nothing in the **AccountNo** field are unlinked activities, while others are there by default. For example, there are a number of records with an **AccountNo** like **PB:DJ**, **PB:MASTER**, etc. These records are the **Peg Board** (**PB:**) records for the various users. It is the information from these records that is utilized in the **Peg Board** tab display on the GoldMine graphical calendar.

In this type of record the **Cal.Notes** field for a given user maintains the users to be displayed on the **Peg Board**, while **Cal.Company** field for each **PB:UserID** holds the monitored information. This information is from the **PB:DJ** record in my **Cal** table:

Cal.Company = 20091009 09:54 20091009 09:54 20091009 10:03 20091009 10:04
Cal.Notes = DJ;MASTER;

You will notice that I am monitoring the GoldMine **Peg Board** activity for the **DJ**, and the **MASTER** logins.

I will examine four of the **RecType** values at the same time as they all use the same dialog form for the input of the information. A typical input dialog form is displayed below in Figure 8-3 to **Schedule an Appointment**.

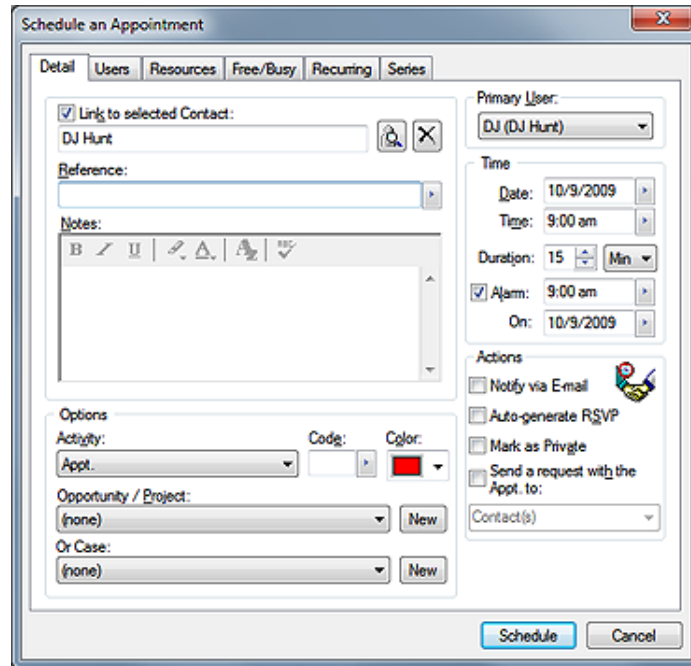


Figure 8-3

The **Activity** field is the only thing that needs to be changed to schedule a different **RecType** (**A**, **C**, **O**, and **T**). Let's begin by looking at the various fields in a record as they pertain to these record types.

UserID - (**Primary User** :) stores the GoldMine **UserID** for whom this activity is scheduled.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table if the **Link to selected Contact**. option remains selected which is it's default state.

OnDate (**Date** :) - contains the date for which this activity is to be scheduled.

Tip

The **OnTime** field can accept a character value such as **A**. By entering such the activity will be placed on the **Task** portion on your graphical calendar. It may still be a dated, and even an alarmed item on your graphical calendars **Task** listing unlike **To-Do's**.

Note

The various color coding values for the 10th byte of **ApptUser** are:

Blue (default) = space or null
 Magenta = !
 Red = " (double quote)
 Cyan = #
 Green = \$
 Yellow = %
 Dark Cyan = &
 White = ' (single quote)
 Light Gray = (
 Maroon =)
 Dark Green = *
 Dark Yellow = +
 Dark Blue = , (comma)
 Purple = -
 Dark Gray = . (period)
 Black = /

WARNING

Cal.Notes will contain **HTML** coding, and this coding will be visible when doing reports unless you, the GoldMine Administrator, specifically turn off this questionable feature or accommodate it in your reports.

You must add:

`HTML_Cal_Notes = 0`

in the **GM.ini** under the **[GoldMine]** section in order to turn off **HTML** coding. Alternatively, you may use the GUI:

Tools
 Configure ►
 System Settings
 Display tab
 Notes Format frame

WARNING

Programmers please note that the **LOpRecID** field must be populated. If there is no linked **Opportunity** or **Case**, then this field must have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.

OnTime (Time:) - contains the time, in a 24 hour format (i.e. **2:00pm = 14:00**) when entered programmatically, for which the activity is scheduled.

EndDate - in all cases of these **RecTypes** that I have examined, this field appears to hold the same date/time as the **OnDate** field. I have even tried scheduling recurring appointments, and, as each activity is schedule separately, each has the same **OnDate**, and **EndDate**.

AlarmFlag - this field must always be populated with either an **N** for no alarm, or a **Y** if the **Alarm:** option has been selected.

AlarmDate (Alarm:) - contains the date on which an alarm is to be triggered, if any.

AlarmTime (On:) - contains the time an alarm is to be triggered, if any. Again, this time is entered into the table in a 24 hour format (i.e. **2:00pm = 14:00**) when it is entered programmatically.

ActvCode (Code:) - could contain up to three characters which act as an activity code for the given activity. This helps to granularize your GoldMine data for more focused reporting and queries.

RSVP - is a one character field that must contain an **N** if the **Auto-generate RSVP** option has not been selected in the scheduling dialog form, and a **Y** if this option has been selected.

Duration (Duration:) - when scheduled through the GoldMine GUI, **Duration:** may be scheduled in minutes or hours. The resulting value, stored in the **Duration** field of the record, is always stored in minutes. (i.e. **16.2 hrs = 972**)

RecType - for this section of the chapter is, obviously I hope, either **A**, **C**, **O** or **T** depending upon the type of activity being scheduled.

AConfirm - despite what the structure table mentions, I have not found this to be used anywhere in my **Cal** table. In fact, after scheduling activities this field contains a **.null.** value.

ApptUser - despite what the structure table mentions, I have only seen that the 10th byte is used, and it is used to store the character representing the color to be displayed for the activity when the activity is being displayed in the GoldMine graphical calendar.

Status - the 1st byte in this field may contain a **B**, if this activity is one of multiple recurring activities that were scheduled at the same time. The 2nd byte will be either a **0** or a **1** if there are no notes, or if there are notes, respectively. The 2nd byte should always contain a value.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has some significance and, hence, should be populated. The **DirCode** will be found in the **SpFiles** table.

Number1 - for these rectypes, this field is used to identify whether this activity has been marked by the creator as **Private**, **Mark as Private**. If the activity has been designated as **Private** then this field will contain **16** otherwise the field will contain a **0**.

Number2 - I have not found any indication of this field being employed for any of these **RecTypes**, hence, being float type of field, a **0** will be contained in this field.

Company - this field is used to store the **Contact** name, or **Company** name when the **Contact** field is blank, of the contact record against whom the activity was scheduled (linked).

Ref (Reference:) - is up to 80 characters of a short description for this scheduled activity.

Notes (Notes:) - is employed to hold any text type **Notes** that were entered for this activity. Be mindful, in GoldMine Premium, these notes may be in HTML format, unless that option was specifically turned off in the GM.ini (see sidebar). Additionally, as of GoldMine Premium 8.5.1.12 this field is a SQL Image based field as opposed to the former SQL Text based field.

LinkRecID - will maintain the **RecID** from the **MailBox** record, if the creator of the activity has selected the option to **Send a request with the <<Activity Name>> to:**. Once entered, this information cannot be removed through GoldMine. It may only be removed through the use of an external application.

LDocRecID - for these **RecTypes**, I have never observed that this field was populated, and, in fact, is populated with a **.null.** value for these **RecTypes**.

LOpRecID - is employed when the activity is associated with an **Opportunity/Project** or a **Case** and, in those situations, this field would contain the **RecID** of said **Opportunity/Project** or **Case** as

Note

As the **LOpRecID** field is but one field, it should be obvious that one can never have an **Opportunity/Project** and a **Case** linked to the same activity.

recorded in the **OpMgr** table or the **Cases** table that the relationship is maintained against. Refer to Note in the sidebar.

If there is no relationship with an **Opportunity/Project** or a **Case** then the application must populate this field with a **space** followed by a generated **RecID** (i.e. **space(1)+LNC8TK#>(+< W<**). This bit of information is probably only relevant to you readers who are developing external applications for use against the GoldMine tables.

CreateBy, CreateOn, CreateAt, LastUser, LastDate, LastTime, Ext, Service, and RecID - are all employed in the normal manner.

Cal.Rectype = B is used for displaying an **Occasion** on the GoldMine graphical calendar. Figure 8-4 and Figure 8-5 on the next page, show the **Edit an Occasion** dialog form. This set of dialogs is accessed via menu selection of:

- Edit
- Record Properties ►
- Contact Details...
- Occasions tab

The record results from these screen entries, and/or if the record were to be added by an external application are as follows:

UserID (User:) - stores GoldMine **UserID** login for the user creating this record or the selected **User** if modified.

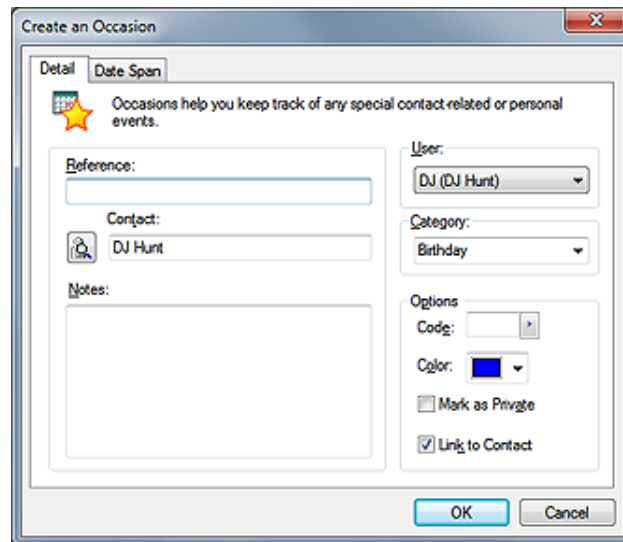


Figure 8-4

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table if the **Link to Contact** option is selected. If **Link to Contact** is not selected, or if an external application creating the record does not want to link the occasion to a contact record, then this field would remain blank.

OnDate (When frame Date:) - is the date on which the first occurrence of the occasion happens.

OnTime - occasions are not displayed in the timed section of the graphical calendar, and, therefore, do not make use of this field.

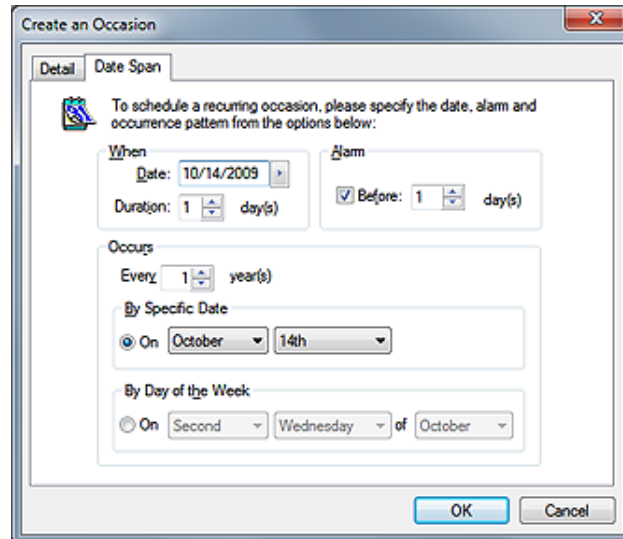


Figure 8-5

EndDate - this date is based on the **Duration**: information that was entered. In Figure 8-5, you can see that we entered a **Duration**: of **1 day(s)**, therefore, the **EndDate** field would be **1 day** from the **OnDate** of **10/14/2009** or **10/15/2009**.

AlarmFlag (Alarm frame) - this field must always be populated. For this **Rectype** there is no graphical option, but a programmer could populate this field through their application with a **Y** or an **N**. This option is simply displayed in this frame as a which is selected in its default state. There is no label for this option beyond the frame label **Alarm**.

Occasion

AlarmTime - if an alarm has been set, the **AlarmTime** is always **00:00** unless programmatically altered.

AlarmDate (Alarm frame Before: X day(s)) - if an alarm is set, it may be set to occur from **0** to any number of days before the established **OnDate**. GoldMine will calculate the date for this field based on the number set by the user in **Before: X day(s)**. External applications just need to enter a date for the alarm in this field.

ActvCode (Code:) - could contain as many as three characters which act as an activity code for the given activity. This helps to granularize your GoldMine data for more focused reporting.

RSVP - is not populated for this record type, and actually remains as a **.null** value.

Duration - when scheduled through the GoldMine GUI, **Duration**: may be scheduled only in days. External applications simply need to populate the number of days over which the occasion is to run. See related **EndDate** field.

RecType - for this section of the chapter is **B**.

AConfirm - despite what the structure table mentions, we have not found this to be used anywhere in our **Cal** table.

ApptUser (Color:) - despite what the structure table mentions, we have only seen that the 10th byte is used to store the character representing the color to be used for the activity when the activity is displayed in the graphical calendar. In Figure 8-4, if I had selected Magenta, this field would contain:

[!]

Status - this field is not used for this record type even if there are notes associated with the record. This is because **RecType B** always has notes as we'll show you in a bit. The field is set to the default **.null** value.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has greater significance, hence should be populated. The **DirCode** will be found in the **SpFiles** table or through the GUI:

[Tools](#)
[Databases](#)
[Open Databases...](#)

The **File Code** column can be seen in the **Contact Set Databases** dialog form as seen here in Figure 8-6. You will notice that the **DirCode** for my database will be **COMMON**.

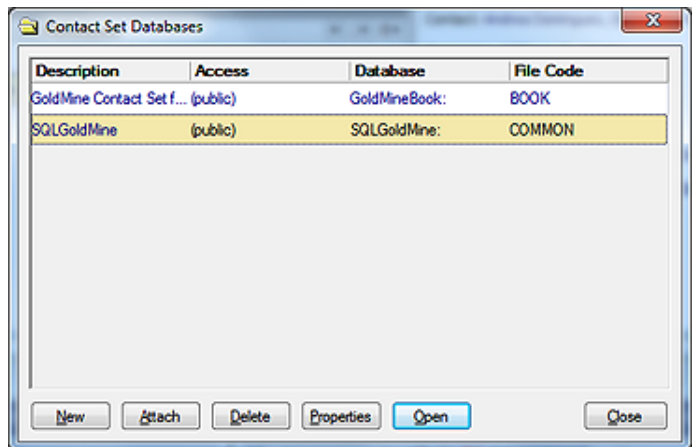


Figure 8-6

Number2 (Category:) - depending on the **Category**: selected this number could be anything from a **0** to **5**. In Example:

- 0 = Other
- 1 = Anniversary
- 2 = Birthday
- 3 = Festival
- 4 = Reunion
- 5 = Retreat

Company - this field is used to store the **Contact** name, or **Company** name, when the **Contact** field is blank, of the contact record for whom the activity was scheduled against.

Note

The various color coding values for the 10th byte of **ApptUser** are:

- Blue (default) = space or null
- Magenta = !
- Red = " (double quote)
- Cyan = #
- Green = \$
- Yellow = %
- Dark Cyan = &
- White = ' (single quote)
- Light Gray = (
- Maroon =)
- Dark Green = *
- Dark Yellow = +
- Dark Blue = , (comma)
- Purple = -
- Dark Gray = . (period)
- Black = /

Note
Indentation is for book presentation only. In the **Notes** field, no such formatting is encountered.

WARNING
Programmers please note that the **LOpRecID** field must be populated. If there is no linked opportunity, then this field **must** have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.

To-Do

Note
Programmers should be aware that in the **OnTime** field, one is not limited to values between 1 and 9. You could as easily use **A** thru **Z**.

Ref (Reference:) - is up to 80 characters of a short description for this scheduled occasion.

Notes (Notes:) - is employed to hold any memo type notes in this image type field that were entered for this activity. Be mindful, in GoldMine Premium, these notes may be in HTML format, unless that option was specifically turned off in the GM.ini as I discussed in an earlier chapter. The notes field for this record type contain very specific information. A typical setting might be:

```
BEGIN:VCALENDAR
PROPID:~/FrontRange Solutions/GoldMine 8.0/EN
VERSION:2.0
METHOD:PUBLISH
BEGIN:VEVENT
ORGANIZER;CN="DJ (DJ Hunt)":MAILTO:--X-GM-USER-- <DJ>
DTSTART;VALUE=DATE:19700902
DTEND;VALUE=DATE:19700903
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYMONTHDAY=2;BYMONTH=9;WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:4543465944334520404F2934522359
DTSTAMP:20091016T131446Z
SUMMARY:Andreas Birthday
PRIORITY:5
CLASS:PRIVATE
CATEGORIES:Birthday
BEGIN:VALARM
TRIGGER:-P1D
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
END:VEVENT
END:VCALENDAR
```

LinkRecID, LDocRecID - are not populated for this record type.

LOpRecID - for this record type there is no association with an **Opportunity/Project**, any external application must populate this field with a space followed by a generated **RecID** (i.e. **space(1)+9F6VPQ2%+IXJR#**).

CreateBy, CreateOn, CreateAt, Ext, Service, and RecID - are all employed in the normal manner.

LastUser, LastDate, and LastTime - are not populated for this record type.

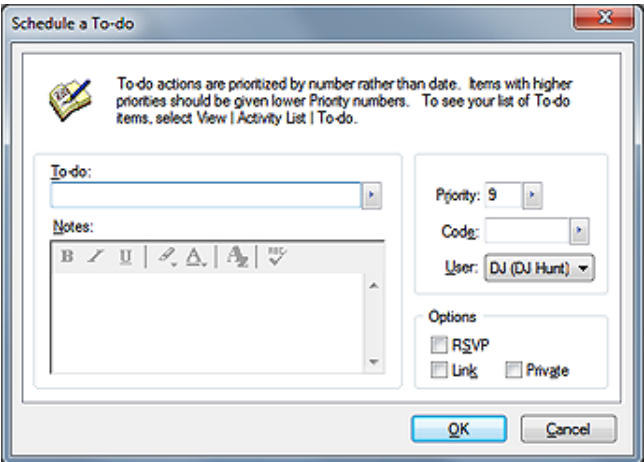


Figure 8-7

Cal.Rectype = D is used for displaying untimed **Tasks**, in GoldMine called a **To-Do**, on the graphical calendar. As one can see in Figure 8-7, not as much information is required to **Schedule a To-do**. They are untimed activities such that no clock is required, and no alarms are available via the GUI. (see **TIP for Appointments, Calls, Other Activities, and Next Actions**)

UserID (User:) - maintains the GoldMine **UserID** for whom the **To-Do** is scheduled.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. If **Link** is not selected, or if an external application creating the record does not want to link the task to a contact record, then this field would be left blank. Unlike the **Appointment, Call, Other Action** and **Next Action**, this option is not selected in the default state.

OnDate - for a task this field always remains **.null**. for this **RecType**. **To-Dos** are displayed everyday on the graphical calendar as long as they have not been completed.

OnTime (Priority:) - maintains the **Priority**: assigned to this task either by the user or programmatically by a developer. Priorities are assigned 1 through 9 normally, 1 being the highest, and 9 being the lowest priority. This information is stored in the first byte of the **OnTime** field.

Note

AlarmDate and AlarmTime may be set programmatically, and these settings will be honored by the GoldMine Calendar. Additionally, within GoldMine, specifically the UserID.ini, or with an override in the GM.ini, the OnByDefault=D may be set to have To-Dos created in GoldMine alarmed by default.

Note

Even though one could type many characters into the ActvCode field, when saved, only the 1st 3 characters will be retained.

EndDate - is populated with the creation date of the scheduled task.

AlarmFlag - for a scheduled task, will always contain an **N** if the **To-Do** was created via GoldMine. If, however, the **To-Do** was created programmatically then this flag could be assigned a **Y**, and the alarms would be activated.

AlarmDate, AlarmTime - are not populated as there is no alarm that can be set unless you have set up your **UserID.ini** or **GM.ini** to alarm this **RecType** in which case the **AlarmDate** field will contain the creation date for the activity. The **AlarmTime** will contain **00:00** (midnight).

ActvCode (Code:) - could contain up to three characters which act as an activity code, identified as **Code:** in the dialog form as shown in Figure 8-7 above. This helps to granularize your GoldMine data for more focused reporting.

RSVP (RSV P) - is a one character field that must contain an **N** if the **RSVP** option has not been selected in the scheduling dialog form, and a **Y** if this option has been selected.

Duration - for the **To-Do RecType** should always be a **.null.** value.

RecType - for this section of the chapter is **D**.

AConfirm - despite what the structure table mentions, we have not found this to be used anywhere in our **Cal** table, and for this **RecType** contains a **.null.** value.

ApptUser - color coding on the graphical calendar for a task is not employed within GoldMine, hence this field is always empty for this **RecType**.

Status - the 2nd byte will be either a **0** or a **1** if there are no notes or if there are notes, respectively. The 2nd byte must always contain a value.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has greater significance, hence should be populated. The **DirCode** will be found in the **SpFiles** table or through the GUI:

[Tools](#)
[Databases](#)
[Open Databases...](#)

Number1 - for this **RecType**, this field is used to identify whether this activity has been marked by the creator as **Private**. If the activity has been designated as **Private** then this field will contain the number **16** otherwise the number **0**.

Number2 - for the **To-Do RecType** will always be **0**. As this is a **Float** based field, the field will always contain a **0**, and there is no need for an application developer to populate this field externally.

Company - this field would normally be used to store the **Contact** name (**Company** name when the **Contact** field is blank), however, even when linked, in GoldMine 8.5.1.12, this field remains empty (not **.null.**, but **empty**).

Ref (To-do:) - contains up to 80 characters of a short description for this scheduled activity.

Notes (Notes:) - is employed to hold any text type **Notes**, in an image based field, that were entered for this activity. Be mindful, in GoldMine Premium, these notes may be in HTML format unless that option was specifically turned off in the **GM.ini** as I discussed earlier in this book.

LinkRecID - will remain empty for this record type.

LDocRecID - will remain contain a **.null.** value for this record type.

LOpRecID - is employed when the activity is associated with an **Opportunity/Project/Case**, and, as this **RecType** can not be associated with one of these, this field must contain a generated value. An external application must populate this field with a space followed by a generated **RecID** (i.e. **space(1)+9F6VPQ2%+IXJR#**).

CreateBy, CreateOn, CreateAt, LastUser, LastDate, LastTime, Ext, Service, and RecID - are all employed in the normal manner.

Cal.RecType = E is used for displaying **Events** on the GoldMine graphical calendar. As one can see in Figure 8-8 on the next page, scheduling **Events** is not that much different from scheduling an **Appointment, Next Action, Other Action**, etc. These are untimed events such that no clock is required but, in this case, alarms are available via the GUI. **Duration:** can only be given in **x Day**

WARNING

Programmers please note that the LOpRecID field must be populated. If there is no linked opportunity, then this field must have a space as the 1st character followed by an API generated RecID. This requirement was included to help with indexing, hence performance.

Event

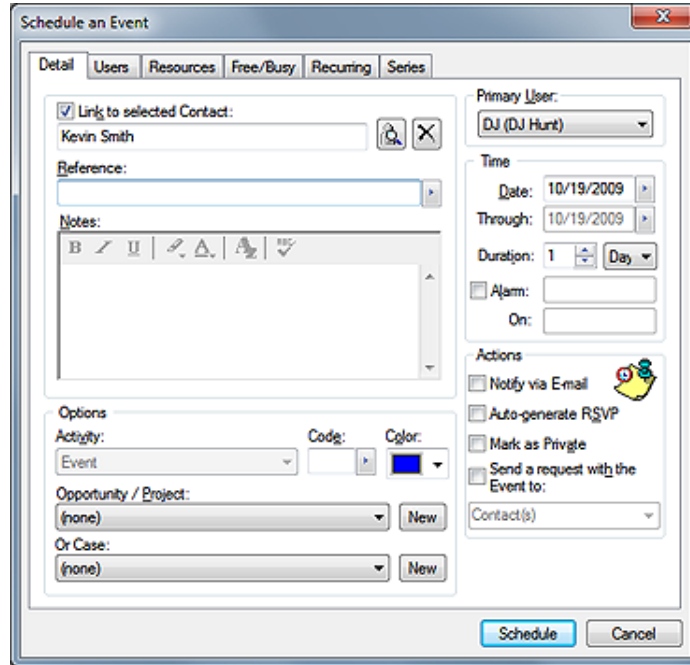


Figure 8-8

(sorry, but that is what the dialog form says) even though it is a drop list and would appear to contain more options. **Events**, as they are untimed, are displayed in the graphical calendar under **Task** listing along with all the other untimed activities such as **To-Do's**. Even though there is a **Through:** field displayed on the **Schedule an Event** GUI, this value is controlled programmatically based on the **Duration:** that one enters.

UserID (Primary User:) - maintains the GoldMine **UserID** against whom the event is being scheduled.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. If **Link to selected Contact:** is not selected, or if an external application creating the record does not want to link the **Event** to a contact record, then this field would be left blank. Like an **Appointment**, **Call**, **Other Action** and **Next Action**, this option is selected in the default state.

OnDate (Date:) - stores the date on which the event begins. Conflict checking is only performed through the GoldMine interface.

OnTime - remembering that **Events** are not timed activities, this field should always remain **empty** for this **RecType**.

EndDate (Through:) - for a change is populated appropriately. If the **Duration:** were populated with **3 Days**, then the **EndDate** would be **OnDate** plus 2 days.

AlarmFlag - this field must always be populated with either an **N** for no alarm, or a **Y** if the **Alarm:** option has been selected.

AlarmTime (On:) - contains the time an alarm is to be triggered, if any. Again, this time is entered into the table in a 24 hour format (i.e. **15:50**).

AlarmDate (Alarm:) - contains the date on which an alarm is to be triggered, if any, and is usually the same date as the **OnDate** value.

ActvCode (Code:) - could contain up to three characters which act as an activity code. This helps to granularize your GoldMine data for more focused reporting.

RSVP - is a one character field that must contain an **N** if the **Auto-generate RSVP** option has not been selected in the scheduling dialog form, and a **Y** if this option has been selected.

Duration (Duration:) - will maintain, in whole days, the length of the event, and should not be converted to minutes.

RecType - for this section of the chapter is **E**.

Note
Programmers are reminded that **empty** and **.null**. Are separate and distinct values.

Note

The various color coding values for the 10th byte of **ApptUser** are:

Blue (default) = space or null
Magenta = !
Red = " (double quote)
Cyan = #
Green = \$
Yellow = %
Dark Cyan = &
White = ' (single quote)
Light Gray = (
Maroon =)
Dark Green = *
Dark Yellow = +
Dark Blue = , (comma)
Purple = -
Dark Gray = . (period)
Black = /

AConfirm - despite what the structure table mentions, we have not found this to be used anywhere in our **Cal** table.

ApptUser (Color) - color coding on the graphical calendar for an **Event** is not employed by GoldMine, hence this field could be left empty for this record type. When scheduling an **Event**, however, you are permitted to select a color. Should the user select a color the code for that color will be stored in the 10th byte of this field.

Status - the 2nd byte will be either a **0** or a **1** if there are no notes, or if there are notes, respectively. The 2nd byte must always contain a value.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has greater significance, hence should be populated. The **DirCode** will be found in the **SpFiles** table or through the GUI:

[Tools](#)
[Databases](#)
[Open Databases...](#)

Number1 - for this **RecType**, this field is used to identify whether this activity has had **Mark as Private** selected by the creator. If the activity has been designated as **Private**, then this field will contain **16** otherwise **0**.

Number2 - for the **Event** will always be **0**. As this is a **Float** based field, the field will always contain a **0**, and there is no need for an application developer to populate this field externally.

Company - this field is used to store the **Contact** name, or **Company** name when the **Contact** field is blank, of the contact record for whom the activity was scheduled against, if the **Event** is linked, as it is in the default state, using the **Link to selected Contact:** option. Otherwise, this field will remain empty.

Ref (Reference) - is up to 80 characters of a short description for this scheduled activity.

Notes (Notes) - is employed to hold any text type **Notes**, in an image based field, that were entered for this activity. Be mindful, in GoldMine Premium, these notes may be in HTML format unless that option was specifically turned off in the **GM.ini** as I discussed earlier in this book.

LinkRecID - will maintain the **RecID** from the **MailBox** item, if the creator of the activity has selected the option to **Send a request with the Event to:**. Once entered, this information cannot be removed through GoldMine. It may only be removed through the use of an external application.

LDocRecID - for this **RecType**, we have never seen this field populated, and, in fact, is **.null.**

LOpRecID - is employed when the activity is associated with an **Opportunity/Project/Case**, and, if there is such an association, this field must contain the **RecID** of the **Opportunity/Project** record or the **Case** record. As this is only one field an **Event** can only be associated with either an **Opportunity**, a **Project** or a **Case**. One cannot associate an **Opportunity** and a **Case** for instance. An external application must populate this field with a space followed by a generated **RecID** if there is no association (i.e. **space(1)+9F6VPQ2%+IXJR#**).

CreateBy, CreateOn, CreateAt, LastUser, LastDate, LastTime, Ext, Service, and RecID - are all employee in the normal manner.

Cal.Rectype = F is used for displaying **Literature Fulfillment Requests** on the graphical calendar. **Literature Requests** are dated, but untimed activities. Hence, they appear on the graphical calendar in the untimed **Task** list on the day for which they are scheduled. Figure 8-9, on the next page, shows the input dialog form for this **RecType**.

UserID (Assigned to) - maintains the GoldMine **UserID** to whom the **Literature Fulfillment Request** activity has been delegated.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. There is no link option associated with this activity as literature requests are always linked to a contact record.

OnDate (Send Date) - stores the date on which the user is scheduled to perform this activity.

OnTime - as stated previously the **Literature Fulfillment Requests** are untimed activities, as such this field should always remain blank for this **RecType**, unless one selects the **Alarm** option. For some unknown reason, this field is populated with 09:00 when the **Alarm** option is selected.

WARNING

Programmers please note that the **LOpRecID** field must be populated. If there is no linked opportunity, then this field **must** have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.

Literature Fulfillment

Note

Programmers must understand the GoldMine paradigm for the **OnTime** and the **AlarmTime** fields, and be careful to populate this information appropriately when creating the **F RecType** records.

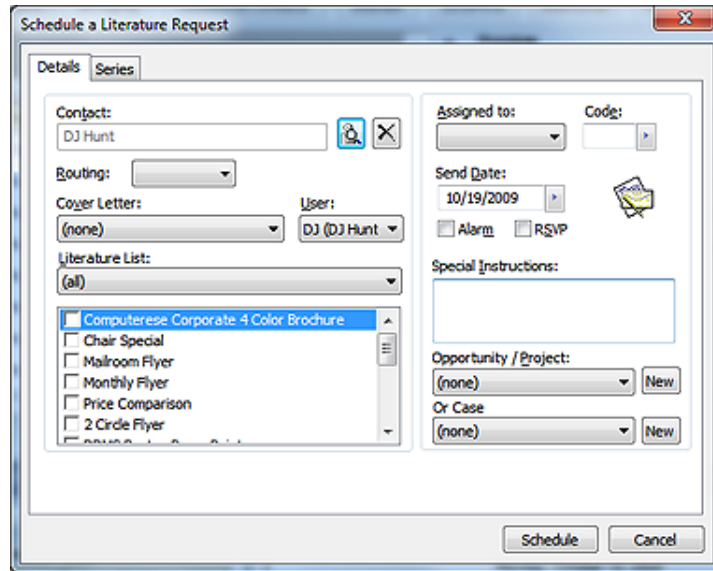


Figure 8-9

EndDate - is/should be populated with the **OnDate** value for this **RecType**.

AlarmFlag (**Alarm**) - this field must always be populated with either an **N** for no alarm or a **Y** if the **Alarm** option has been selected.

AlarmTime - would contain the time an alarm is to be triggered, if any. Again, this time is entered into the table in a 24 hour format (i.e. **09:00**). There is no alarm time field available on the dialog form, hence, the **AlarmTime** is set to **9:00am** (i.e. **09:00**).

AlarmDate - contains the date on which an alarm is to be triggered, if any. Again, there is no alarm date field available on this dialog form, and the field is automatically set to the value contained in the **OnDate** field. Programmers populating this field through an external application should be consistent with this paradigm.

ActvCode (**Code**) - could contain up to three characters which act as an activity code. This helps to granularize your GoldMine data for more focused reporting.

RSVP (**RSVP**) - is a one character field that must contain an **N** or a **Y** if the **RSVP** option has been selected.

Duration (**Routing**) - being a **Float** based field, is always defaulted to **0**, however, the value is controlled by the **Routing**: selection where a **0** = **Printer**, and a **1** = **FAX**.

RecType - for this section of the chapter is **F**.

AConfirm - despite what the structure table mentions, I have not found this to be used anywhere in my **Cal** table. This field always appears to contain a **.null** value.

ApptUser - There is no color coding available for this **RecType**, hence, this field is populated empty for this **RecType**.

Status - the 2nd byte will be either a **0** or a **1** if there are no notes or if there are notes, respectively. The 2nd byte must always contain a value. For this rectype there are almost always **Notes**, as I will describe later, hence this value should always be a **1**.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has greater significance, hence should be populated. The **DirCode** will be found in the **SpFiles** table or through the GUI:

[Tools](#)
[Databases](#)
[Open Databases...](#)

Number1 - for this **RecType** there is no option to mark the activity as **Private**. As this field is a **Float** based field the default value of **0** is displayed.

Number2 - for the task will always be **0**. As this is a **Float** based field, the field will always contain a **0**, and there is no need for an application developer to populate this field externally.

Company - this field is used to store the **Contact** name, or **Company** name when the **Contact** field is blank, of the contact record for whom this activity was scheduled against. As stated previously, this **RecType** must always be scheduled against a contact record.

Ref (Special Instructions:) - is up to 80 characters of a short description for this scheduled activity. This field is populated by the information contained in the **Special Instructions:** in the dialog form shown in Figure 8-9 on the previous page. Take special note, even though the dialog form makes this appear to be a text based field, it is not. Your users will be able to type continuously, however, when saved, only the first 80 characters of the **Special Instructions:** are retained.

Notes (Literature List:) - for this **RecType**, is used to store the requested literature, along with the **RecID** relating to that piece of literature as stored in the **Forms** table. A typical example of this might be:

```
Computerese Corporate 4 Color Brochure 8K72TH5)B+!(+;  
Chair Special 7A7OO9N(@M(J[_V  
Mailroom Flyer 9JIWR65*DDV2[_V  
Monthly Flyer 7A7OMC9&9&3M[_V  
Price Comparison 7A7OPC2&%JRG[_V
```

LinkRecID (Cover Letter:) - will maintain the **RecID** from the **Forms** table item, if the creator of the activity has selected to include a **Cover Letter:** in their **Literature Fulfillment Request**.

LDocRecID - for this **RecType**, I have never seen this field populated with anything other than the default **.null.** value.

LOpRecID - employed when the activity is associated with an **Opportunity/Project/Case**, and, if there is such an association, this field must contain the **RecID** of the **Opportunity/Project** record or the **Case** record. As this is only one field a **Literature Request** can only be associated with either an **Opportunity**, a **Project** or a **Case**. One cannot associate an **Opportunity** and a **Case** for instance. An external application must populate this field with a space followed by a generated **RecID** if there is no association (i.e. **space(1)+9F6VPQ2%+IXJR#**).

CreateBy, CreateOn, CreateAt, LastUser, LastDate, LastTime, Ext, Service, and RecID - are all employee in the normal manner.

Cal.Rectype = H is used exclusively by GoldMine to maintain the various **Holiday** records that the administrator may choose to have displayed on the graphical calendar. There are only a few fields that are populated in the **Cal** table, and I will cover them here. However, you should keep in mind that the holidays are added to the **Cal** table via the **Option** GUI of a **UserID** possessing **Master Rights** within GoldMine.

UserID - maintains the GoldMine derived **UserID** for the holiday set. This takes the form of sequential, character based numbering, **000** through **012** in GoldMine Premium.

AccountNo - for this **RecType**, will always be **Holiday**.

RecType - for this **RecType**, will always be **H**.

Status - notice on this **RecType**, even though there are notes, that the 2nd character flag is not set as in the past **RecTypes** that we have discussed. On the other hand, notice that the 1st character flag is set to either a **2** or a **4**. FrontRange has stated that this is the flag value for categories:

- 2 - country (from the predefined set eg..US, UK etc)
- 4 - religion (Christian, Jewish etc)
- 8 - userdef category

Ref - contains the label assigned to the holiday set, and it is this label that is displayed when selecting the **Holiday** group via the **Options**.

Notes - this field contains the defined holidays associated with the given category. Here is an example of the information contained therein, and the astute observer will notice that there are beginning and ending tags, reminiscent of XML coding:

```
BEGIN:VCALENDAR  
PROPID:--FrontRange Solutions//GoldMine 8.0//EN  
VERSION:2.0  
METHOD:PUBLISH  
BEGIN:VEVENT
```

Note

*Programmers take note, each line in the Notes field for **RecType = F** is terminated with a **chr(10)**, while **chr(9)**, the tab character, is used to separate the document name from the **Form.RecID**.*

WARNING

*Programmers please note that the **LOpRecID** field must be populated. If there is no linked opportunity, then this field **must** have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.*

Holiday

Note

*Indentation is for presentation only. In the **Notes** field, no such formatting is encountered.*

Note

Indentation is for presentation only. In the **Notes** field, no such formatting is encountered.

```
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020101
DTEND;VALUE=DATE:20020102
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYMONTHDAY=1;BYMONTH=1;WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39324D5057424A294A3C5D30303031
DTSTAMP:20020722T081234Z
SUMMARY:New Year's Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020121
DTEND;VALUE=DATE:20020122
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=MO;BYMONTH=1;BYSETPOS=3;
WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B45394A284126285F462636
DTSTAMP:20020722T081234Z
SUMMARY:Martin Luther King Jr. Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020218
DTEND;VALUE=DATE:20020219
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=MO;BYMONTH=2;BYSETPOS=3;
WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B48343326602A3236462636
DTSTAMP:20020722T081234Z
SUMMARY:Presidents Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020527
DTEND;VALUE=DATE:20020528
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=MO;BYMONTH=5;BYSETPOS=-1;
WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B484D502441253235462636
DTSTAMP:20020722T081234Z
SUMMARY:Memorial Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020704
DTEND;VALUE=DATE:20020705
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYMONTHDAY=4;BYMONTH=7;WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B4941302A3B343635462636
DTSTAMP:20020722T081234Z
SUMMARY:Independence Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20020902
DTEND;VALUE=DATE:20020903
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=MO;BYMONTH=9;BYSETPOS=1;
WKST=SU
TRANSP:OPAQUE
```

Note

Indentation is for presentation only. In the **Notes** field, no such formatting is encountered.

```

SEQUENCE:0
UID:39334C4B495946254738414A462636
DTSTAMP:20020722T081234Z
SUMMARY:Labor Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20021014
DTEND;VALUE=DATE:20021015
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=MO;BYMONTH=10;BYSETPOS=2;
WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B4C4433245C4A3448462636
DTSTAMP:20020722T081234Z
SUMMARY:Columbus Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20021111
DTEND;VALUE=DATE:20021112
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYMONTHDAY=11;BYMONTH=11;WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B4E305A243B327B2A462636
DTSTAMP:20020722T081234Z
SUMMARY:Veterans Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
BEGIN:VEVENT
ORGANIZER;CN="MASTER ()":MAILTO:--X-GM-USER--<MASTER>
DTSTART;VALUE=DATE:20021128
DTEND;VALUE=DATE:20021129
RRULE:FREQ=YEARLY;COUNT=30;INTERVAL=1;BYDAY=TH;BYMONTH=11;BYSETPOS=4;
WKST=SU
TRANSP:OPAQUE
SEQUENCE:0
UID:39334C4B4F3446244F234056462636
DTSTAMP:20020722T081234Z
SUMMARY:Thanksgiving Day
PRIORITY:5
CLASS:PUBLIC
CATEGORIES:HOLIDAY
END:VEVENT
END:VCALENDAR
    
```

You may have noticed that there are **9** predefined holidays in this text file, and they happen to be in the **Cal.Notes** field for the **Cal.Ref = United States** record.

WARNING

Programmers, please note that the **LOpRecID** field must be populated. If there is no linked opportunity, then this field **must** have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.

LOpRecID - employed when the activity is associated with an **Opportunity/Project/Case**, and, if there is such an association, this field must contain the **RecID** of the **Opportunity/Project** record or the **Case** record. As the **Holiday RecType** has no link, an external application must populate this field with a space followed by a generated **RecID** (i.e. **space(1)+9F6VPQ2%+IXJR#**).

CreateBy, **CreateOn**, **CreateAt**, **LastUser**, **LastDate**, **LastTime**, **Ext**, **Service**, and **RecID** - are all employee in the normal manner.

E-mail, Queued E-mail Messages & Quotas

Cal.RecType = M and **Q** have a lot in common as **M** is for E-mail messages while **Q** is for Queued E-mail messages. Structurally, these two record types are virtually identical. One thing that an Administrator/Developer should understand is that these record types in the **Cal** table, and the **ContHist.sRecType** of **M** in the **ContHist** table, are little more than related referential pointers to the actual messages that reside in the **MailBox** table. These are **not** the actual messages themselves that are being sent/received.

Note

GoldMine will use **RecType = [Q]** for both **Queued E-mails** waiting to be delivered, and **Quotas** for a **UserID**.

UserID - maintains the GoldMine **UserID** for the user associated with the **From: /M** or **To: /Q** account in the e-mail itself. When **Q** is used for **Quota**, this field is blank.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. Outgoing messages are **Linked** by default, and incoming messages are linked to a record if the

e-mail address is contained within the GoldMine dataset, and the user Options has been selected to **Link e-mail address to contact record by default** on the **E-Mail | Advanced** tab. Having said this, then you can see that it is possible for this field to be **.null.**, as would be the case with unlinked outgoing/incoming e-mails. When **Q** is used for **Quota**, this field looks something like this:

QSDJ 20070731

Where the **QS** is followed by the full 8 characters of the userid, and then the **To Date**: for the quota period.

OnDate - stores the date from the **Date** value contained in the header section of the e-mail. When **Q** is used for **Quota**, this field contains the **From Date**: for the quota period.

OnTime - stores the time portion of the **Date** value contained in the header section of the e-mail. When **Q** is used for **Quota**, this field is used for the first 5 characters of the userid.

EndDate - is/should be populated with the **Cal.OnDate** value for this **RecType**.

AlarmFlag - this field must always be populated with either an **N** for no alarm or a **Y** if there is an alarm. This field can be set for outgoing Internet e-mail, but is irrelevant, as it is not received by the e-mail recipient. As the dialog form is used for both GoldMine Internal E-mail and Internet E-mail the **Alarm** checkbox is really only applicable to the GoldMine Internal E-mail. In some cases, i.e. GoldMine sending the user **RSVP**, you may not see the normal **N** value. I have noticed, on occasion, that the GoldMine application, itself, is inserting an * into this field.

AlarmTime - contains the time an alarm is to be triggered, if any. Again, this time is entered into the table in a 24 hour format (i.e. **14:00**). There is no alarm time field available on the dialog form, hence, the **Cal.AlarmTime** is set to the send time minus the user preference decrement, default 10 minutes (i.e. **13:50**). Programmers populating this field through an external application should be consistent with this paradigm, and refer to **Cal.AlarmFlag**.

AlarmDate - contains the date on which an alarm is to be triggered, if any. Again, there is no alarm date field available on this dialog form, and the field is automatically set to the value contained in the **OnDate** field. Programmers populating this field through an external application should be consistent with this paradigm, and refer to **AlarmFlag**.

ActvCode - there is no place to enter a **Code** unless Queuing a message where the field is called **Activity**. Retrieved Internet E-mail will not have this field populated.

RSVP - is a one character field that must contain an **N** or a **Y** if this option has been selected, however, as with **Cal.AlarmFlag** this field is only relevant for GoldMine Internal E-mail, though it will show up in queued Internet e-mail. This field can be set for outgoing Internet e-mail, but is irrelevant as it is not received by the e-mail recipient. As the dialog form is used for both GoldMine Internal E-mail and Internet E-mail the **RSVP** checkbox is really applicable to GoldMine Internal E-mail only. In some cases, i.e. GoldMine sending the user **RSVP**, you may not see the normal **N** value.

Duration - being a Float based field, is always defaulted to **0**, and is not used with this **Cal.RecType**.

RecType - for this section of the chapter could be an **M** or **Q**.

AConfirm - despite what the structure table mentions, I have not found this to be used anywhere in our **Cal** table.

ApptUser - There is no color coding available for these rectypes, hence, this field is not populated for these rectypes with exceptions. When GoldMine, itself, is sending an **RSVP** to an activity which did possess a color coding the **Cal.ApptUser** field will also mimic that color coding.

Status - the second byte will be a **0**. The 2nd byte must always contain a value. For these rectypes, there are no notes as this record is only a pointer to the **Mailbox** e-mail message record.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has significance and, hence, should be populated. The **DirCode** will be found in the **SpFiles** table.

Number1 - for these rectypes defaults to **7**. If **Private**, this value is the sum of **7 + 16** or **23**. I have determined, with assistance from Paul Redstone, that the setting of the queue options of an Outgoing Internet E-mail determines the base value of this field.

Here is the base number as defined by the option selection.

- | | |
|---|--|
| 7 = <input checked="" type="checkbox"/> Create a history record | <input checked="" type="checkbox"/> Save the body text |
| 6 = <input type="checkbox"/> Create a history record | <input checked="" type="checkbox"/> Save the body text |
| 5 = <input checked="" type="checkbox"/> Create a history record | <input type="checkbox"/> Save the body text |
| 4 = <input type="checkbox"/> Create a history record | <input type="checkbox"/> Save the body text |

Number2 - for the task will always be 0. As this is a Float based field, the field will always contain a 0, and there is no need for an application developer to populate this field externally.

Company - unlike the **Company** field for other **Cal.RecType** values, the **Company** field for these rectypes comes from the **Contact's name**: in the **ContSupp** table that is associated with the **From**: address of the e-mail. When **Q** is used for **Quota**, this field is used for storing the **Quota**:, **Forecast**:, **Closed Sales**: and **Lost Sales**: values. There are 60 characters in this field, and each value could be up to 15 characters in length with no special characters being used.

Ref - up to 80 characters of the **Subject**: line from the e-mail message retrieved or queued.

Notes - There are no **Notes** for these rectypes. Remember that the e-mail message is stored in total in **Mailbox.RFC822**, and that there is only a link to that **Mailbox** record contained in the **Cal** record for these rectypes.

LinkRecID - will maintain the **Mialbox.RecID** for the associated/linked **Mailbox** record for these rectypes.

LDocRecID - for this rectype, we have never seen this field populated, and it appears to maintain the default **.null**. value.

LOpRecID - as there can now be an association with an **Opportunity/Project/Case**, then this field could be populated with the related **RecID** for the **Opportunity/Project/Case**. Any external application populating this rectype record must populate this field with a space followed by a generated **RecID**. (i.e. **space(1)+9F6VPQ2%+IXJR#**)

CreateBy, **CreateOn**, **CreateAt**, **LastUser**, **LastDate**, **LastTime**, **EXT**, **SERVICE**, and **RecID** - are all employed in the normal manner. **CreateBy** for received Internet e-mails is always **INTERNET**.

Forecasted Sales

Cal.RecType = S is the last rectype that I will be covering with respect to the **Cal** table. This rectype stands for, appropriately enough, **Sale** as the GoldMine menu displays it.

UserID (User:) - maintains the GoldMine **UserID** for the user associated with the sale.

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. Scheduling a **Sale** is **Linked** by default although it is possible to have an unlinked sale (no **AccountNo**) it is certainly never recommended as this tends to defeat the premise behind a CRM solution.

OnDate (Sale Date:) - stores the **Sale Date**: which is better known as the expected close date for the sale. The key words here are "expected close date".

OnTime - Forecasted Sales are untimed activities that display under the Task list in the graphical calendar, hence, this field is usually empty (not **.null**.). However, this does not prevent the programmer from placing a value into this field causing their Forecasted Sale to be rendered in the timed activity area of the graphical calendar. Again, this time would be entered into the table in a 24 hour format (i.e. **14:00**).

EndDate - is and should be populated with the **Cal.OnDate** value for this rectype.

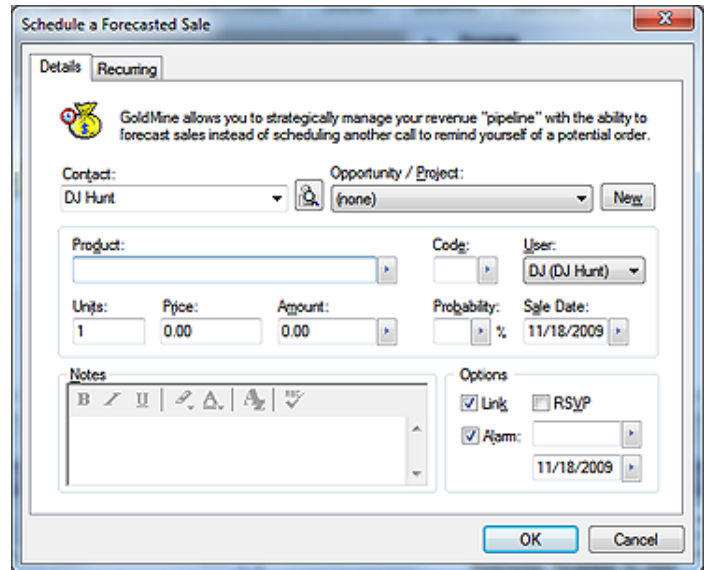


Figure 8-10

AlarmFlag - this field must always be populated with either an **N** for no alarm or a **Y** if the **Alarm**: option has been selected.

AlarmTime - contains the time an alarm is to be triggered, if any. Again, this time is entered into the table in a 24 hour format (i.e. **14:00**).

AlarmDate - contains the date on which an alarm is to be triggered, if any.

ActvCode (Code:) - could contain up to three characters which act as an activity code for the given activity. This helps to granularize your GoldMine data for more focused reporting.

RSVP (**RSVP)** - is a one character field that must contain an **N** if the **RSVP**: option has not been selected in the scheduling dialog form, and a **Y** if this option has been selected.

Duration (Probability:) - being a Float based field, is always defaulted to **0**, however, for this rectype this field maintains the **Probability**: value as entered via the dialog form, and is used in many reports within GoldMine.

RecType - for this section of the chapter must be **S**

AConfirm - despite what the structure table mentions, we have not found this to be used anywhere in our **Cal** table.

ApptUser - There is no color coding available for this rectype as this type of activity is displayed in the untimed Task area of the graphical calendar, hence, this field is not populated for this rectype. However, the programmers among us, that have placed this activity in the timed area of the graphical calendar, may wish to color code this activity (**Green = \$**).

Status - the 1st byte in this field may contain a **B** if this activity is one of multiple activities that were scheduled at the same time, while the second byte will be either a **0** or a **1** if there are no notes, or if there are notes, respectively. The 2nd byte must always contain a value regardless.

DirCode - is, quite simply, the **File Code** for the contact dataset for which this activity has been scheduled against. In the past, this field was not employed. Today, the information in this field has significance and, hence, should be populated. The **DirCode** will be found in the **SpFiles** table.

Number1 (Amount:) - for this rectype, this field is used to store the **Amount**: as supplied from the dialog form shown in Figure 8-10 on the previous page. One must remember that this is the value contained in the **Units**:, refer to **Cal.Number2**, times the value of the **Price**: as displayed in the same dialog form. As this field is being used to store the amount of the sale, there is no option for privatizing this type of rectype.

Number2 (Units:) - for this rectype, this field stores the **Units**: as supplied from the dialog form.

Company (Contact:) - this field is used to store the **Contact** name, or **Company** name when the **Contact** field is blank, of the contact record for whom the activity was scheduled against.

Ref (Product:) - is up to 80 characters of a short description for this scheduled activity.

Notes (Notes:) - is employed to hold any plain text or rich text type **Notes**: that were entered for this activity. Be mindful, in GoldMine Premium that these notes will be in an HTML format unless that option was specifically turned off in the **GM.ini** as I discussed in Chapter 3, although it is now preferable to use rich text (HTML) Notes.

LinkRecID - will remain empty for this rectype.

LDocRecID - for this rectype we have never seen this field populated, and retains the default **.null**. value.

LOpRecID - as there could be an association with an Opportunity/Project, this field could contain the RecID for the related Opportunity/Project. Otherwise, the default value is a space followed by a generated RecID. (i.e. **space(1)+9F6VPQ2%+IXJR#**)

CreateBy, CreateOn, CreateAt, LastUser, LastDate, LastTime, EXT, SERVICE, and RecID - are all employed in the normal manner.

WARNING

*Programmers please note that the **LOpRecID** field must be populated. If there is no linked opportunity, then this field **must** have a space as the 1st character followed by an API generated **RecID**. This requirement was included to help with indexing, hence performance.*

MailBox

MailBox Table

Indexes

MBOXACCNO	AccountNo (Non-Unique, Non-Clustered)
MBOXFOLDER	Folder (Non-Unique, Non-Clustered)
MBOXLINK	LinkRecID (Non-Unique, Non-Clustered)
MBOXMAILID	MailID (Non-Unique, Non-Clustered)
MBOXUSER	UserID, Folder, Folder2, MailDate (Non-Unique, Non-Clustered)
MBXRECID	RecID - Unique (Unique, Non-Clustered)

Relationships

MailBox.RecID	Many-to-One	Cal.LinkRedID
MailBox.RecID	Many-to-One	ContHist.LinkRecID
MailBox.AccountNo	Many-to-One	Contact1.AccountNo
MailBox.LOpRecID	One-to-One	OpMgr.RecID
MailBox.LOpRecID	One-to-One	Cases.RecID

Structure

AccountNo	VarChar 20	Linked Contact AccountNo - not Null
CreateOn	DateTime 8	Creation Date
Ext	VarChar 5	Type
Flags	VarChar 8	Flags*
Folder	VarChar 20	Folder identification** - not Null
Folder2	VarChar 20	Subfolder identification - not Null
LinkRecID	VarChar 15	RecID of linked CalContHist record - not Null
LOpRecID	VarChar 15	RecID of linked OpMgr/Cases record
MailDate	DateTime 8	Mail Date
MailID	VarChar 200	Mail identification - not Null
MailSize	VarChar 8	Mail Size
MailTime	VarChar 8	Mail Time
MailRef	VarChar 100	Reference
RFC822	Image 16	Entire mail message including coding
RecID	VarChar 15	Record ID - not Null
UserID	VarChar 8	GoldMine UserID - not Null

Notes

* The flags field, although character based, contains the converted binary number which relates to the following states:

<u>Bit</u>	<u>On</u>	<u>Off</u>
1	Read	Not Read
2	In History	Not in History
3	Outbound	Inbound
4	Attachments	No Attachments

** The Folder field contains the name of the folder in which mail is stored. There are several predefined folder names:

X-GM-FOLDERS	-	Main Level Folder
X-GM-GROUPS	-	Groups (Legacy Identifier)
X-GM-HTMLTAB	-	GM+Views
X-GM-ICALINFO	-	iCalendar Information
X-GM-INBOX	-	Inbox
X-GM-PROP-HTMLTAB	-	HTML Tab Settings
X-GM-OUTBOX	-	Outbox
X-GM-RULES	-	E-mail Rules
X-GM-SUBXXXXX	-	Subfolder of a Main Level Folder, XXXXX might be FILED
X-GM-SURX-GMRXXXXXXXX	-	Subsubfolder, XXXXXXXX = left 7 Chr of the Subfolder RecID
X-GM-TD-ITEMS	-	MyGoldMine Items
X-GM-TD-SETTINGS	-	MyGoldMine Settings
X-GM-TEMPLATES	-	E-mail Templates
X-GM-TRASH	-	Trash

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. This field may be blank in the **MailBox** table if the message was not linked. Additionally this field may contain the display name for a GoldMine predefined Folder.

<u>Folder</u>	<u>ACCOUNTNO</u>
X-GM-FOLDERS	Main Level folder name
X-GM-SUBXXXXX	Main Level folder name for this subfolder
X-GM-SURX-GMRXXXXXXXX	X-GMRXXXXXXXXXXXXXXXXX RecID for folder

CreateOn - as in all other tables is the creation date of the record. Depending on the Users Options this may be the date that this record was created in GoldMine or it may be the date from the e-mails header.

Flags - which is actually a character based field, and holds a numeric value. This value is derived from the binary number that represents the value for the four states (see table on previous page).

Folder - may contain any of the default GoldMine folder identifiers (see table on the previous page). This field may also contain any user defined top level folder names. Said folders may contain sub-folders.

Folder2 - contents is related to **Folder** contents, and may contain:

Folder	Folder2
X-GM-FOLDERS	Empty
X-GM-GROUPS	Empty
X-GM-HTMLTAB	GM+Views template name
X-GM-ICALINFO	iCal request information
X-GM-INBOX	Empty
X-GM-OUTBOX	Empty
X-GM-PROP-HTMLTAB	Empty
X-GM-RULES	Empty
X-GM-SUBXXXXX	Subfolder name
X-GM-SURX-GMRXXXXXX	Subsubfolder name
X-GM-TD-ITEMS	Empty
X-GM-TD-SETTING	Empty
X-GM-TEMPLATES	Empty
X-GM-TRASH	Empty

Note

*Solica Consulting Limited, at <http://www.Solica.com> has developed a tool to fix the "Cal record not found" errors. This tool, **FixCal**, is distributed freely, and you use it at your own risk.*

Note

*Unlike other tables, the **Mailbox**. **LOpRecID** field does not need to be populated when there is no relationship to an **Opportunity/Project/Case**.*

LinkRecID - whenever there is an e-mail in the **Mailbox** table there is also a related record in the **Cal** table if the e-mail has not been **Filed**, and if it has been filed, then the relationship is to the **ContHist** table. The **RecID** of the record from the related table is stored in this field, and, correspondingly, the **RecID** from the e-mail record, in the **MailBox** table, is stored in the **LinkRecID** field of the related record. Occasionally, you may encounter a "**Cal record not found.**" error. There is no known cause as to why the link breaks, only that it does get broken occasionally (refer to sidebar).

LOpRecID - maintains any links that may result with this e-mail being associated with an Opportunity/Project/Case. This field would contain the RecID of the linked Opportunity/Project/Case.

MailDate - holds the date which is recorded for the e-mail in its header if the user has selected **Use date from mail header** in their GoldMne Options, otherwise this field should contain the retrieval date for any given message.

MailID - This field was newly added in 8.5.1.12, and contains the **E-mail Identification** for a given e-mail message. This should prevent GoldMine from downloading an e-mail message that it had previously downloaded (no more duplicates hopefully). Syntactically the ID could resemble:

<RDQ1UIVBSVF7QD85UiNZMzA3OTgxOA@DJ>

MailSize - retains the size in bytes of the e-mail message.

MailTime - holds the time which is recorded for the e-mail in its header if the user has selected **Use date from mail header** in their GoldMne Options, otherwise this field should contain the retrieval time for any given message.

MailRef - maintains a concatenated string which is the sum of the **From**: value in the e-mail header, the **Tab** character, **chr(9)**, and the **Subject**: value in the e-mail header.

RFC822 - a memo type field that maintains the whole enchilada, so to speak. I believe that **RFC822** was an e-mail format type, hence the name of the field. This field contains the e-mail in its entirety. Here is a sample message:

```
Date: Wed, 18 Nov 2009 19:24:15 -0500
From: "Kelly & Danny" <k41875@djhunt.us>
Subject: Eric Superbowl Game
To: <DJ@DJHunt.US>
Message-ID: <9E2E0FC0CFF04FE6A342B5967AAC949D@homebb7fb904c2>
Mime-Version: 1.0 X-Mailer: Microsoft Office Outlook 11
Received: (gmail 16595 invoked by uid 78); 19 Nov 2009 00:26:52 -0000
Received: from unknown (HELO cloudmark1) (10.49.16.78) by 0 with SMTP; 19 Nov 2009 00:26:52
Return-Path: <k41875@djhunt.us>
Received: from [206.46.173.1] ([206.46.173.1:53871] helo=vms173001.mailsvcs.net) by cm-mr4
(envelope-from <k41875@djhunt.us>) (ecelerity 2.2.2.41 r(31179/31189)) with ESMTTP id E7/11-00355-
CC0940B4; Wed, 18 Nov 2009 19:26:52 -0500
Received: from homebb7fb904c2 ([unknown] [72.85.204.48]) by vms173001.mailsvcs.net (Sun
Java(tm) System Messaging Server 7u2-7.02 32bit (built Apr 16 2009)) with ESMTTP id <0KTB-
00K6UYGGXCC1@vms173001.mailsvcs.net> for DJ@DJHunt.US; Wed, 18 Nov 2009 18:26:51-0600
```


Cases

Note
There is a significant structural change as of at least build 8.5.2.5, whereby many fields that were previously **not Null** are today capable of accepting **Null** values.

CASES Table Indexes

CASEACC	AccountNo (Non-Unique, Non-Clustered)
CASENUM	Number (Non-Unique, Non-Clustered)
CASEOWNER	Owner (Non-Unique, Non-Clustered)
CASERECID	RecID (Unique, Non-Clustered)

Relationships

Cases.RecID	One-to-Many	CaseTeamLink.CaseID
Cases.RecID	One-to-Many	CaseInfoLink.CaseID
Cases.RecID	One-to-Many	CaseAttachment.CaseID
Cases.RecID	Many-to-One	Cal.LOpRecID
Cases.AccountNo	One-to-One	Contact1.AccountNo

Structure

AccountNo	VarChar	20	Linked Contact AccountNo - not Null
Category	VarChar	40	Category of Case
Created_By	VarChar	8	UserID
Created_Date	DateTime	8	Case Creation Date
Description	Text	16	Description
Due_Date	DateTime	8	Autogenerated Due Date
FType	VarChar	40	Type of Case
Is_Read	SmallInt	2	Read - not Null
Is_Template	SmallInt	2	Template Identification - not Null
Modified_By	VarChar	8	Last Modified By
Modified_Date	DateTime	8	Last Modified Date
Notes	Text	16	Notes
Number	VarChar	40	Autogenerated Case Number - not Null
Offering	VarChar	200	Product/Service Associated with Case
Owner	VarChar	8	GoldMine UserID of Case Owner - not Null
Priority	VarChar	40	Priority of Case
RecID	VarChar	15	Record ID - not Null
Resolution_Note	Text	16	Resolution Note
Resolution_Type	VarChar	40	Resolution Type
Resolved_By	VarChar	8	UserID of Resolver
Resolved_Date	DateTime	8	Time Resolution Achieved
Source	VarChar	40	Source of Case
Status	SmallInt	2	Current State*
Subject	VarChar	200	Subject of Case

* The following are possible values for the Status field:

- | | | |
|-------------|------------|--------------|
| 0 Template | 1 Assigned | 2 Reassigned |
| 3 Escalated | 4 Resolved | 5 Abandoned |

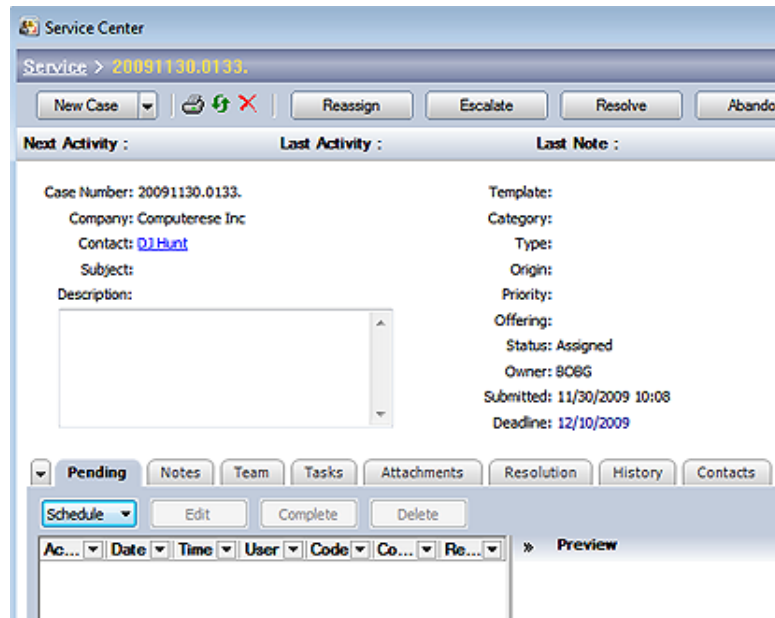


Figure 8-11

AccountNo - stores the relational link back to the appropriate record in the **Contact1** table. This field may be blank in the **Cases** table as in the **Case Template** record.

Note

If your organization is not using Case Templates then I highly recommend that you populate your F2 Lookup List for each field so as to derive some form of consistency of data input.

As we all know, this will assist you when reporting against this data.

Category (Category:) - stores up to 40 characters for your **Case Category**. This value may be predefined via your **Case Templates**. This helps to granularize your Cases read/opened by the assigned **Owner**.

Due_Date - self descriptive, and autogenerated based on the **Case Manager Preferences** dialog form **Default Follow-up Interval**: value.

FType - record type field that appears to be used when creating **Case Templates** when completing the **Case Type**: field.

Is_Read - self descriptive, however, this field stores a **0** or a **1**. If programmatically being set, the coder must make certain that one initiates this value to **0**.

Is_Template - self descriptive, however, this field stores a **0** if this is not a **Case Template** record or a **1** if this is a **Case Template** record. If programmatically being set, the coder must make certain that one initiates this value to **0**.

Notes (Notes tab) - being a Text based field, this field can store an almost infinite amount of information, and, although displayed differently then we have been used to in the past, it still contains the same information as we have all come to know and expect with the addition of a little HTML coding.

```
*** DJ (DJ Hunt) *** 12/2/2009 10:04:56 AM
<div>
    This ia a Note added under the Case Notes tab.
</div>
```

Number (Case Number:) - Each organization may configure this differently. A programmer will need to look at the **Cases.Number** field of the last **Cases** record created to see the current formatting configuration. You may expect that the second segment is the number that is incremented for each new Case.

Example: **DJ.0104.04112007**

Offering (Offering(s): Main Dialog | Product/Service: Template Dialog) - stores up to 200 characters for this fields value. Again, this value can be predefined in your **Case Templates**, and has an F2 Lookup list associated with the field as well. However, you are reminded that the F2 Lookup list values may not, themselves, exceed 40 characters which is a limitation of the **Lookup** table structure.

Owner (Owner:) - holds the **UserID** of the individual who is responsible for the flow of this particular Case record.

Priority (Priority:) - stores up to 40 characters for this fields value. Again, this value can be predefined in your **Case Templates**, and has an F2 Lookup list associated with the field as well.

Resolution_Note (Resolution tab) - being a Text based field, this field can store an almost infinite amount of information, and, although displayed differently than we have been used to in the past, it still contains the same information as we have all come to know and expect with the addition of a little HTML coding.

```
*** DJ (DJ Hunt) *** 12/2/2009 9:55:54 AM
<div>
    <FONT face=Tahoma size=2>
        <DIV>
            This is a test of a Resolved Case
        </DIV>
    </FONT>
</div>
```

Resolution_Type - currently I do not see that this field is being utilized, however, it is being defaulted to **empty** as opposed to having a **.null**. value.

Source (Origin:) - this value may contain up to 40 characters. Again, this value can be predefined in your **Case Templates**, and has an F2 Lookup list associated with the field as well if you are not using templates.

Status - refer to the footnote for the table on the previous page for the possible value that may appear in this field.

Subject (Subject:) - stores up to 200 characters for this fields value. Again, this value can be predefined in your **Case Templates**, and has an F2 Lookup list associated with the field as well. However, you are reminded that the F2 Lookup list values may not, themselves, exceed 40 characters.



Modified_By, Modified_Date, Created_By, Created_Date, LastDate, Resolved_By, Resolved_Date, and RecID - are all employed in the expected manner.