

# Record Typing in GoldMine

Copyright © 2004 FrontRange Solutions, Inc.

---

**Author:** Mark Acquista  
Beverly Albright  
**Department:** Technical Support  
**Created:** April 1, 2004  
**Last Updated:** June 7, 2004

## Copyright

*Copyright © 2004 FrontRange Solutions Inc. All Rights Reserved. GoldMine, HEAT, and other FrontRange products and brands are registered trademarks or trademarks of FrontRange Solutions Inc. in the U.S. and/or other countries. Other products and brands are registered trademarks or trademarks of their respective owners/companies.*

*USE OF THE SOFTWARE DESCRIBED IN THIS PAPER AND ITS RELATED USER DOCUMENTATION ARE SUBJECT TO THE TERMS AND CONDITIONS OF THE APPLICABLE END-USER LICENSE AGREEMENT (EULA).*

The information contained in this document is provided “as is” without warranty of any kind. To the maximum extent permitted by applicable law, FrontRange Solutions disclaims all warranties, either express or implied, including the warranties for merchantability and fitness for a particular purpose; and in no event shall FrontRange Solutions or its Sellers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if advised of the possibility of such damages.

# Table of Contents

- COPYRIGHT ..... 2
- RECORD TYPING ..... 4**
- OVERVIEW ..... 4
- PLANNING FOR RECORD TYPES ..... 4
- RECORD TYPES ADMINISTRATION CENTER ..... 5
- CREATING PRIMARY FIELDS VIEWS ..... 5
- MODIFYING PRIMARY FIELDS VIEWS ..... 7
- Recovering Deleted Fields ..... 9
- CONFIGURING CUSTOM SCREENS AND GM+ VIEW ..... 9
- CREATING A RECORD TYPE ..... 9
- RECORD TYPE RULES ..... 11
- NEW RECORD CREATION OPTIONS ..... 17

# Record Typing

## Overview

Record Typing is a feature that allows you to configure GoldMine so that contact records can be displayed in a particular format based on the predefined components for any specific contact type.

Once the different types of contacts are identified, GoldMine can then be configured to display each contact type using a different format. As you scroll through the contact records in a GoldMine database configured with Record Typing, the view can dynamically change depending upon the record type assigned to each record.

## Planning for Record Types

Prior to actually creating Record Types for your GoldMine database, it is strongly recommended you outline and document your plan. You may wish to read through this document to gain a better understanding of Record Types before outlining your plan. This document will instruct you regarding the components of a Record Type. Primary Fields Views are a required component while Custom Screens and GM+ Views are optional.

- **Primary Fields Views**
  - Which fields will remain and which will be deleted?
  - Will the field label names change?
  - Will you want the font to be a different color for fields or data?
- **Custom Screens**
  - Will you want to create a custom screen in order to group certain fields together on the *Fields* tab or a new tab?
- **GM+ Views**
  - Will you want to create a custom GM+ View tab for your record types?

**TIP:** As you begin the process of defining Record Types, we recommend keeping everything as simple as possible. It is a good idea to use the same name for the Record Type, the Primary Fields View, the Custom Screen and the GM+ View tab. If the name is the same for all of these components, it will make it much easier to manage and configure.

## Record Types Administration Center

To begin the process of defining Record Types go to **File>>Configure>>Record Types Administration Center**.

Below is a screen shot of the Administration Center before any record types have been defined.

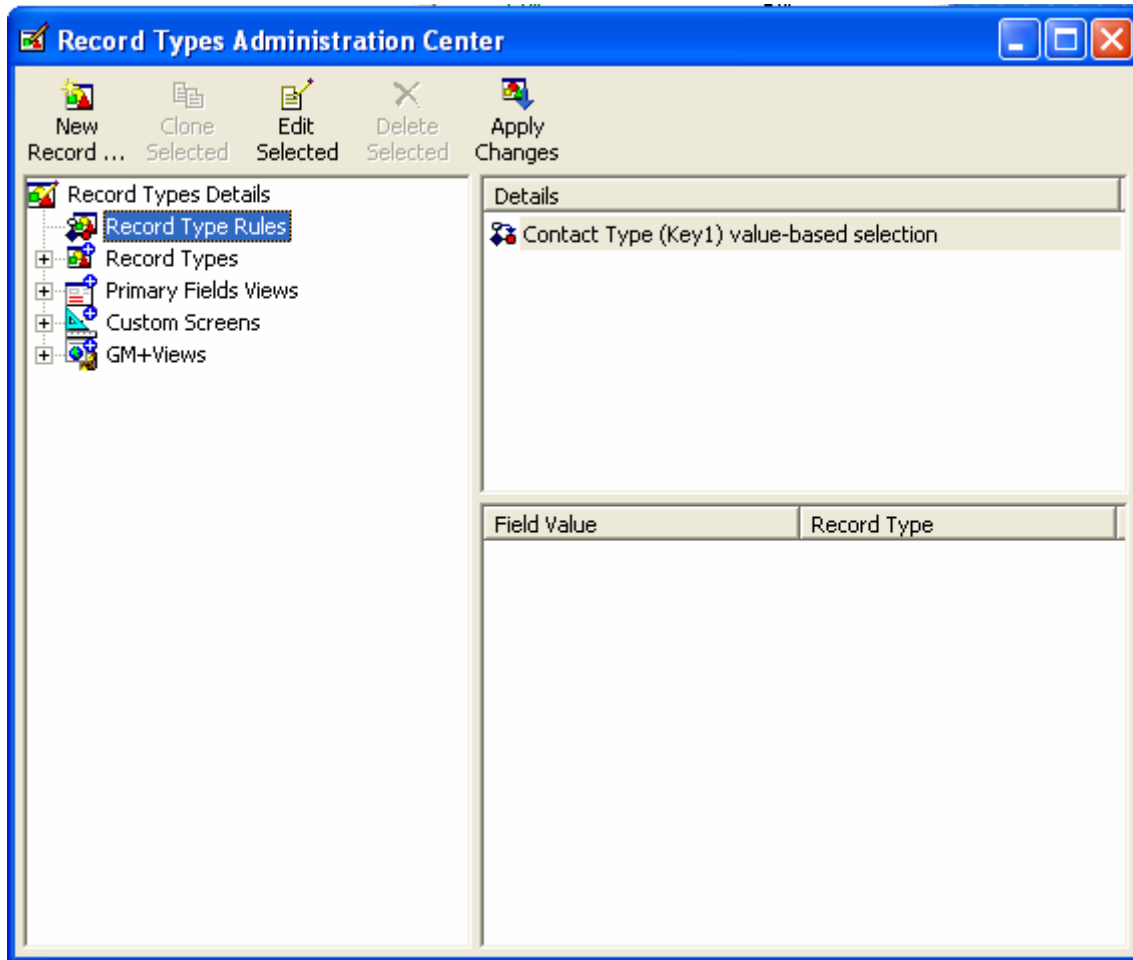


Figure 1

## Creating Primary Fields Views

Primary Fields Views are a key component of Record Typing and their creation is the first step in establishing record types. They are customized views of the GoldMine primary fields, those in the upper four panes of a Contact Record. Since it is recommended that you not move any of these fields from the upper four panes, you can customize the views to reflect different business

needs while still working within one database. The customizations can include changing the displayed label for a field, the displayed color, or hiding a field entirely.

You can create as many Primary Field Views as needed to meet your business needs. For example, a Real Estate agent may use GoldMine to store contacts that are buyers, sellers, appraisers, and lenders. In this scenario the Agent could divide the database into four different record types, each having an associated Primary Fields View. There is no set limit to the number of views you can have in GoldMine; the number is limited only by the amount of available computer memory.

1. Expand the *Primary Fields Views* option so that you see the *Main View*. (Figure 2)

**NOTE: We recommend never making changes to the Main View; this should always remain untouched and never be modified.**

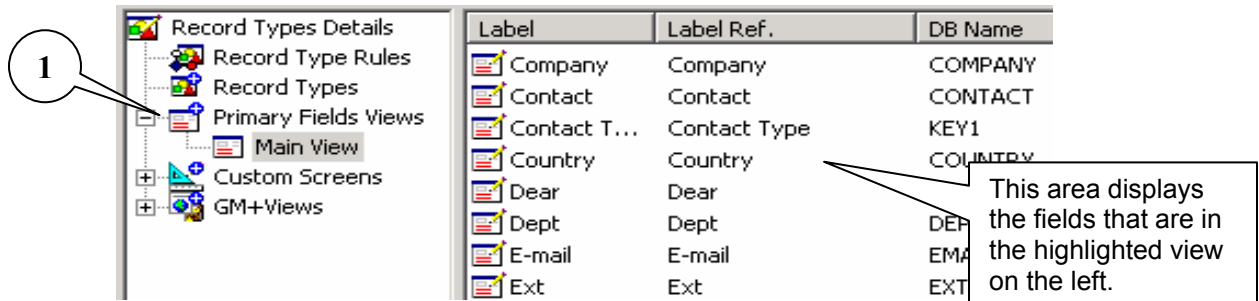


Figure 2

2. Right click on the *Main View* and select the **Clone** option (Figure 3-A). This displays the *Main View Profile* window (Figure 3-B).

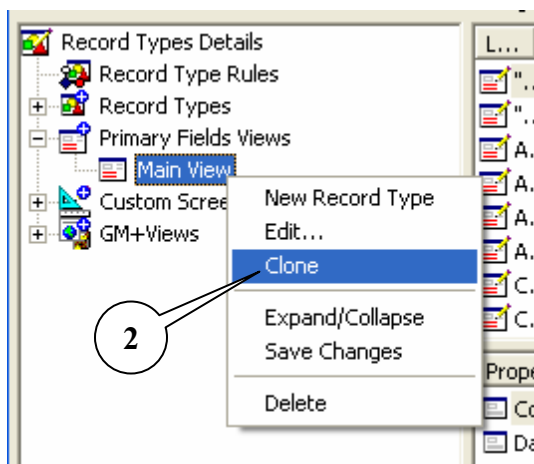


Figure 3-A



Figure 3-B

3. **Enter a name for the view** and leave the *User Access* set to public. For the Real Estate scenario, the Main View would be cloned four times with each view assigned a name corresponding to the record types (Buyers, Sellers, Lenders, Appraisers). The resulting view of the Administration Center is seen in Figure 4.

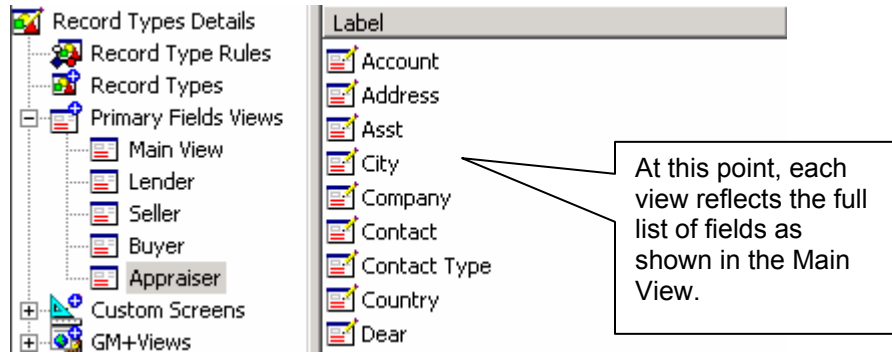


Figure 4

## Modifying Primary Fields Views

Now that the Primary Fields Views are created, you can begin modifying what GoldMine will look like when each View is activated. Fields can be completely deleted from the View so that they will not appear at all; this removes the field from the View only, not from the database. Fields can be edited to alter the labels used, and to alter the font colors for the labels or the actual field data. The changes made here are specific to the View that you have selected and will only be displayed when this View is activated.

1. Click on the View you wish to customize.
2. Right click on a field and choose either to **Delete** it or **Edit** it.

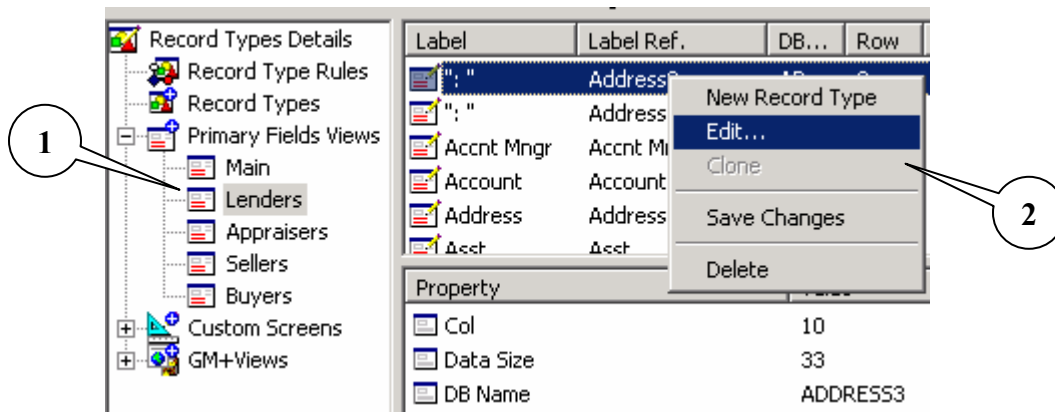


Figure 5

- When selecting to edit a field, the *Field Properties* window opens.

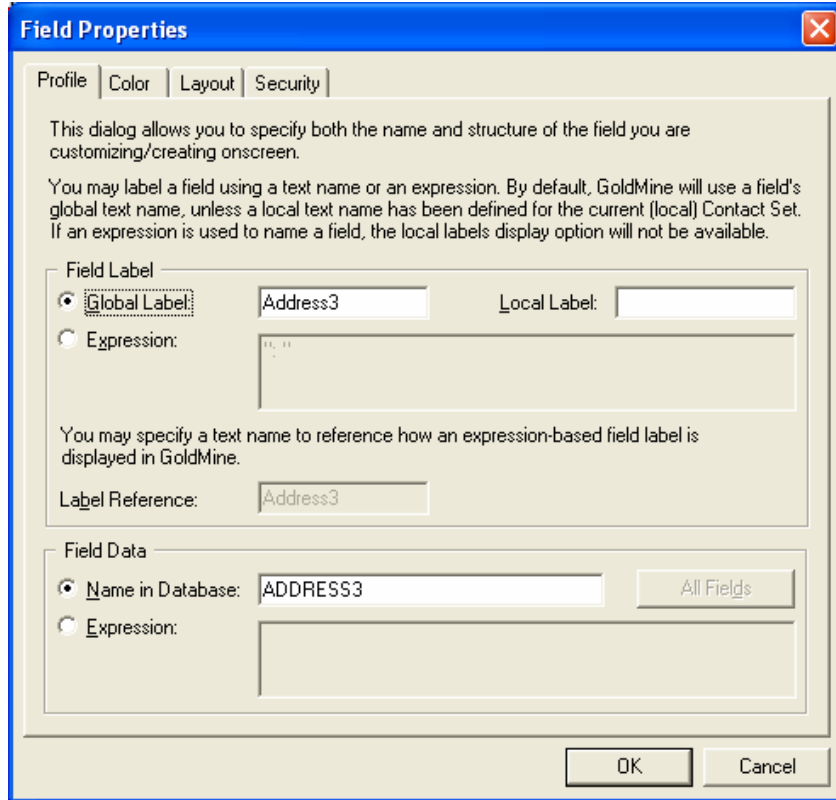


Figure 6

- Use the tabs and associated selections to modify the appearance of the field. For example, you may choose to have the labels appear in different colors based on the type of contact: Lenders in green, Appraisers in black, Sellers in blue, and Buyers in red. Each field must be edited individually; there is no way to globally assign settings to all fields in a View.
- Figure7 shows the Lenders View with some of the primary fields deleted as well as the field labels appearing in green.



Figure 7 - Lenders

## Recovering Deleted Fields

If you chose to delete a field from a View and decide to add it back, this procedure can be used:

1. Right click anywhere in the upper four quadrants of a GoldMine record.
2. Choose **Select Primary Field** and then choose the View that is missing a field.
3. Once the View is displayed, right click and choose **New Field**.
4. Select the field from the dropdown list and reinsert back to its original location. Note that you may have to adjust the properties of the field to match its original. You can do this by viewing the properties of the field in the Main View.

## Configuring Custom Screens and GM+ View

In addition to the Primary Fields Views, two other possible components of a Record Type are Custom Screens and GM+ View. These two components are entirely optional; you could have a Record Type that contains just a Primary Fields View and no custom screen or GM+ View.

The Record Types Administration Center includes Custom Screens and GM+ Views in the tree structure. If you expand each of these you will see which screens and/or GM+ Views have already been created. Note, however, that you cannot *create* new Custom Screens or GM+ Views from the Record Types Administration Center, but you can *edit* them just as you edit your Primary Fields Views.

All Custom Screens and GM+ Views must be created using the already defined methods in GoldMine.

- Custom Screens - Right click on the *Fields* tab and select *Screens Setup*; or go to File>>Configure>>Custom Screens.
- GM+ View - Go to File>>Configure>>GM+ View.

Once you have created these items, they will appear in the Record Types Administration Center as potential components for a Record Type.

## Creating a Record Type

Once Primary Fields Views are created, and optionally the Custom Screens and GM+ Views, the actual Record Type can be created. In the Record Type Administration Center:

1. Right click on *Record Type* and choose the **New Record Type** option.

2. Enter a name for the Record Type and select which Primary Field View will be used.
3. If desired, select the Custom Screen and/or GM+ View for the Record Type.
4. Additionally, you can choose to set this Record Type as the default for your GoldMine system.

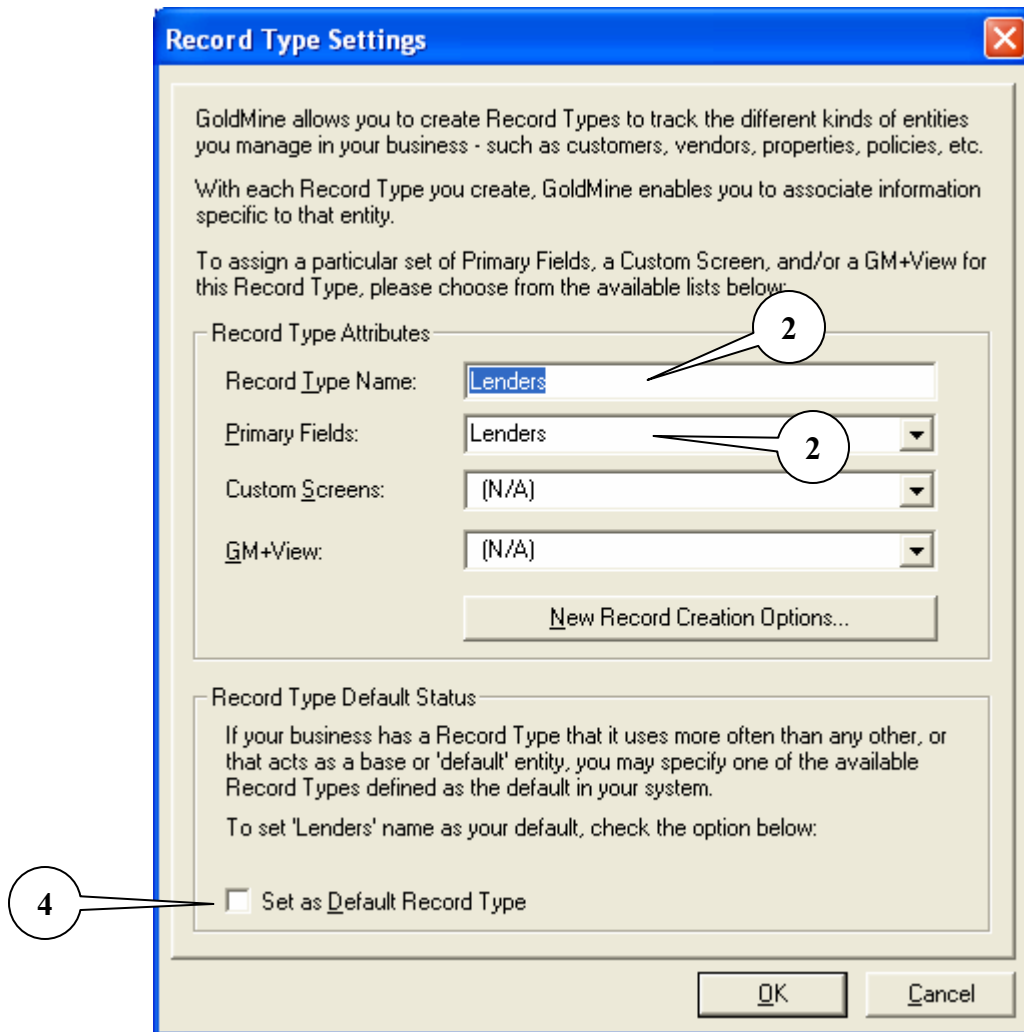
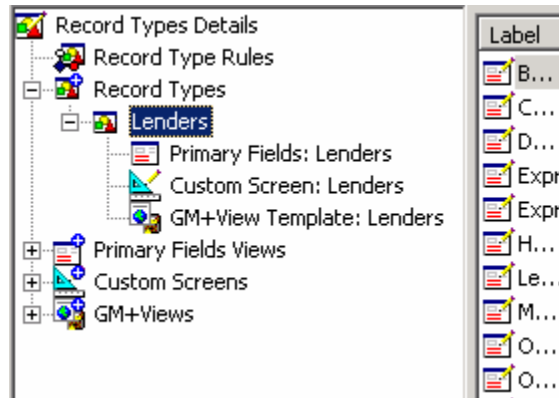


Figure 8

5. Click **OK**.
6. Figure 9 on the following page illustrates the new Lenders Record Type and its components.



**Figure 9**

Once you have configured all of the components and the actual Record Types you want to use, rules must be defined to instruct GoldMine when to use each of the Record Types.

## Record Type Rules

Record Types can be triggered in one of two ways: **Field Value-Based Rules** or **dBase Expression Result-Based Rules**. The *Field Value-Based* rule can only use the value of one field to determine which Record Type will be used. The *dBase Expression Result-Based* rule can be configured to evaluate the value of multiple fields to determine the correct Record Type.

### Field Value Based Rules

For the *Field value-based* rule you will choose which field in your database will be used to trigger the Record Typing, and then define which Record Type to use based on the value in that field. In the scenario we've been using, the Key 1 (Contact Type) field is the field used to record whether a contact is a buyer, seller, lender or appraiser. If this field has "Lenders" in it, then the Lenders Record Type will be used; if this field has "Buyers" in it, then the Buyers Record Type will be used, and so forth

1. Double click on *Record Type Rules* in the Record Type Administration Center.
2. The *Rule Definition for Record Type* window, as seen in Figure 10 on the next page, will open.

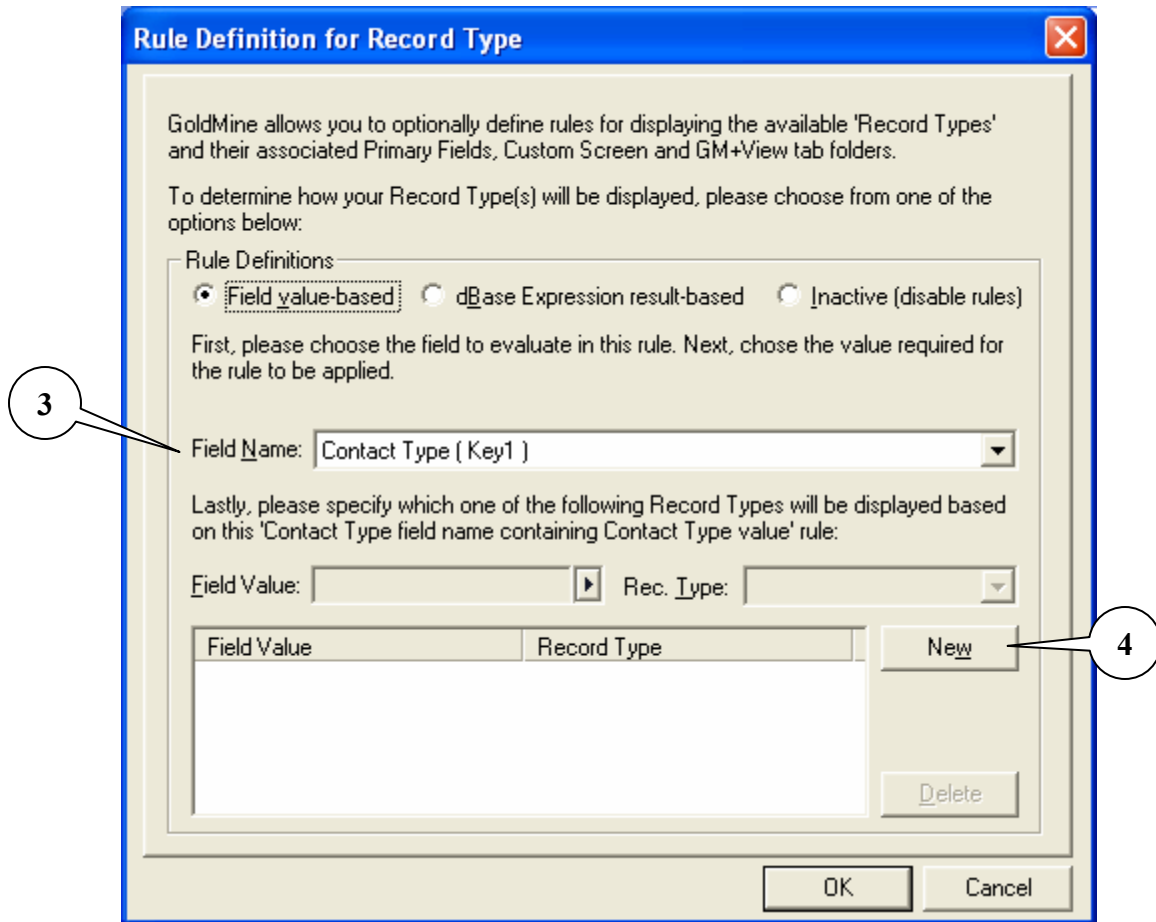


Figure 10

3. Select the field for *Field Name* that will be used to trigger Record Typing. In our scenario this is the Contact Type (Key1) field.
4. Click the **New** button to activate the *Field Value* and *Rec. Type* fields.
5. *Field Value* – You may type the value or select it from the Lookup list, provided the entry has been added to the Lookups.

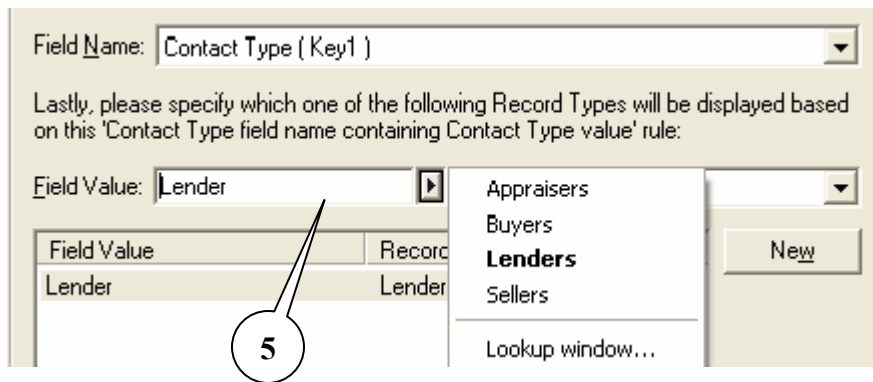


Figure 11

6. *Rec. Type* – Select the corresponding Record Type from the dropdown list.

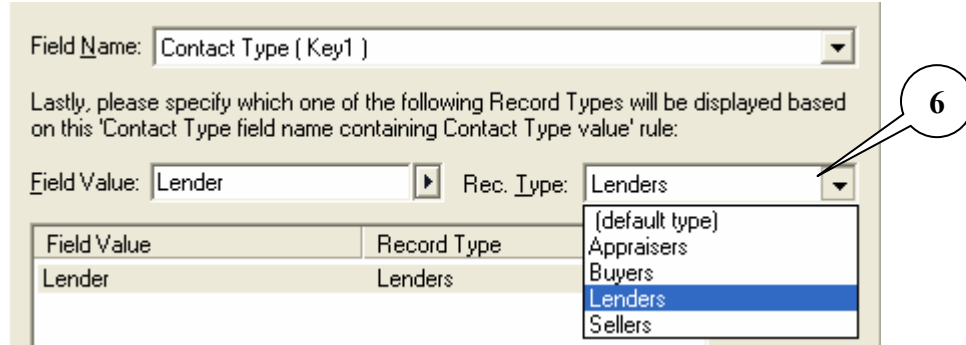


Figure 12

7. To add rules for additional Record Types, click the **New** button again. (See Figure 13 below for an example of what the *Rule Definition* window would look like for the real estate scenario.)

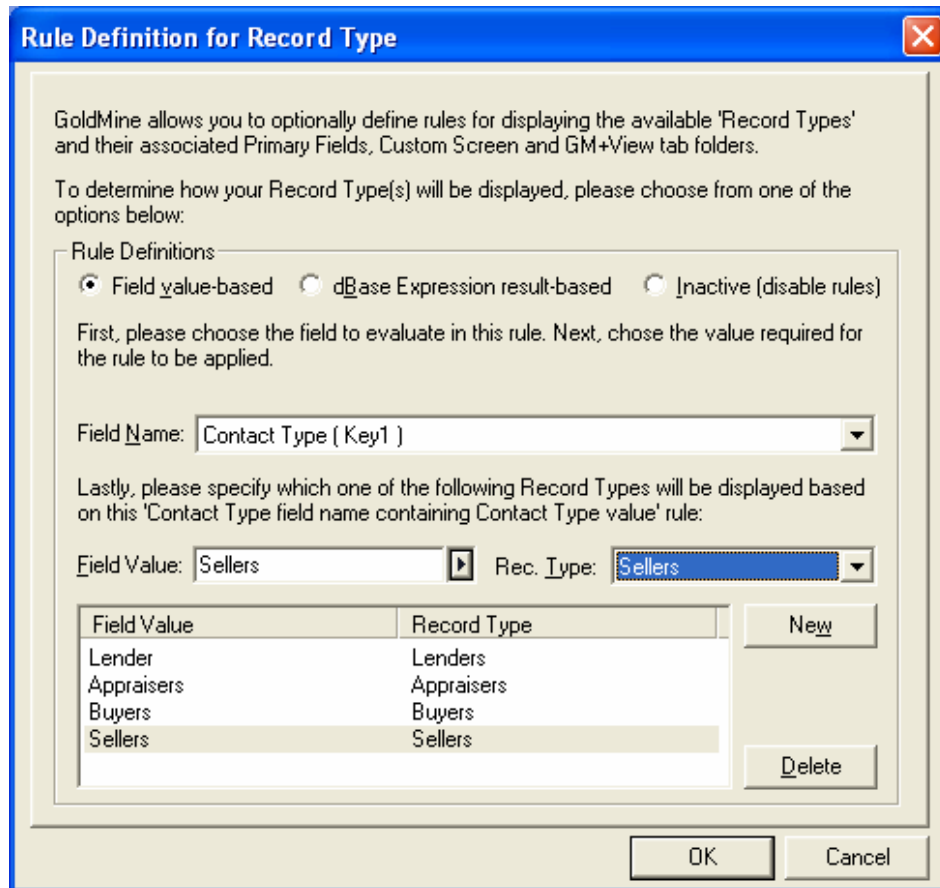
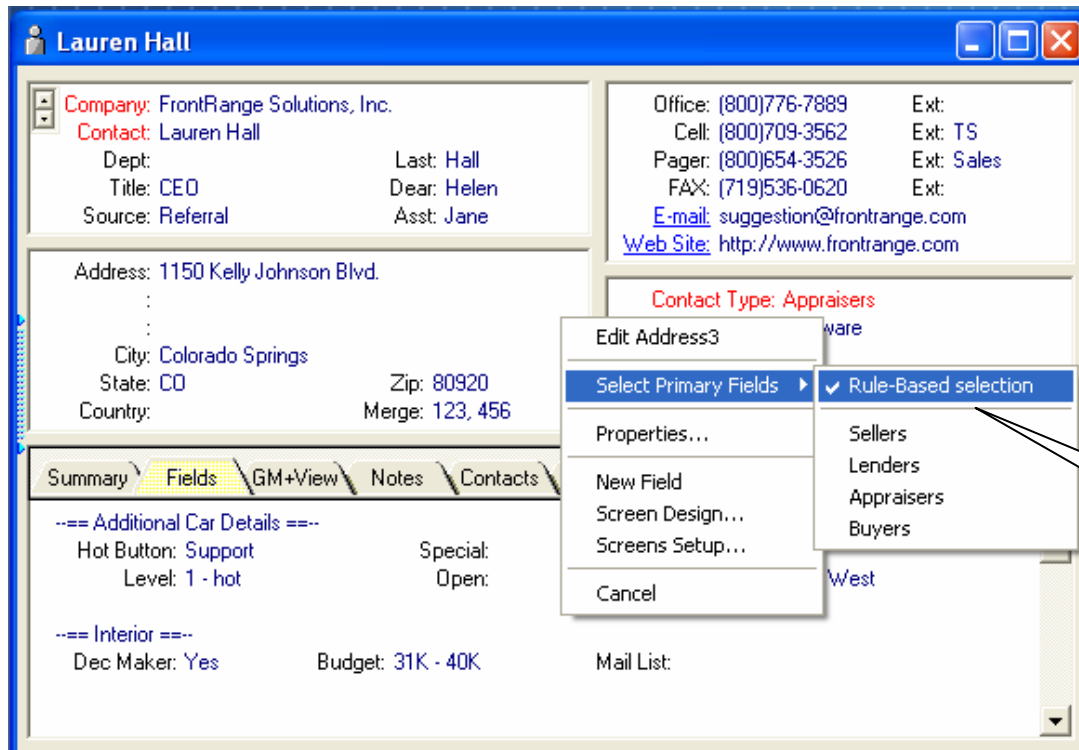


Figure 13

8. Click **OK** when your rules are defined.

9. Click the **Apply Changes** button. This will save all modifications made to any of the components.
10. Close the Record Types Administration Center.
11. To verify your record type selection is based on rules, right-click in the upper four panes of the contact record and select **Select Primary Fields**. On the expanded local menu, **Rules-Based selection** should have a check mark next to it; otherwise, select it.



**Figure 14**

**TIP:** This is a great way to preview what your Primary Field Views look like as you create them. You can select one to activate it and make sure it displays correctly.

## dBase Expression Result Based Rules

Remember, in order to evaluate multiple fields to determine Record Type, you must create rules based on dBase expressions. Creating rules based on dBase expressions is primarily for advanced users. However, this document does include some sample expressions using the real estate scenario, that can be modified to fit your Record Types. For more information on dBase expressions, refer to the technical document entitled *Using dBase Expressions* posted in the knowledge base.

**NOTE:** If you activate the dBase expression rules, you are not able to create new records based on record types. You must activate field value-based rules to create new records based on record types.

1. Launch the Record Types Administration Center and double click on the *Record Type Rules*.
2. Select **dBase Expression result-based** to display an *Expression* text box.

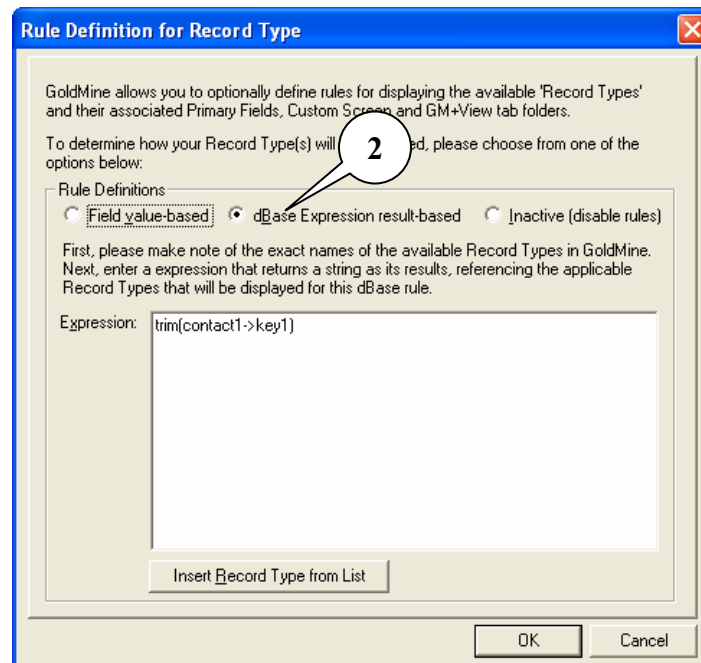


Figure 15

3. Type (or copy and paste) the expression you are creating in the text box. (See samples below.)
4. As you type or modify the expression, use the **Insert Record type from List** button to select a record type from the list and guarantee data consistency.
5. When you are finished creating your expression, click **OK** and then click **Apply Changes** on the Record Types Administration toolbar.
6. To verify your record type selection is based on rules, right-click in the primary fields and select **Select Primary Fields**. On the expanded local menu, **Rules-Based selection** should have a check mark next to it; otherwise, select Rules-Based selection (as seen in Figure 14 on the previous page).

## Sample expressions using values corresponding to Record Type Names

When the string value in the field being evaluated corresponds to the name of the record type, you can use the following types of expressions.

### Simple Expression

Assuming Key1 can have Sellers, Buyers, Appraisers or Lenders as a value and assuming you have record types named Sellers, Buyers, Appraisers and Lenders, you can use the following expression:

**trim(contact1->key1)**

**Results:** Each record where the key1 field value equals Sellers, the Sellers record type is displayed. Each record where the key1 field value equals Buyers, the Buyers record type is displayed. Each record where the key1 field value equals Appraisers, the Appraisers record type is displayed. Each record where the key1 field value equals Lenders, the Lenders record type is displayed.

### Sum of Two Fields Expression

Assuming Key1 can contain the value Sellers or Buyers and Key3 can contain Insured or Uninsured and assuming you have four record types named Sellers Insured, Sellers Uninsured, Buyers Insured, Buyers Uninsured, you can use the following expression:

**trim(contact1->key1)+" "+trim(contact1->key3)**

**Results:** The expression returns <Key1string><space><Key2string>. If the value in Key1 is Sellers and the value in Key2 is Uninsured, then the Sellers Uninsured record type is displayed for the record. When you move to another record where the value of Key1 is Buyers and the value in Key2 is Insured, then the Buyers Insured record type is displayed for the record.

### If...Then...Else Expressions

An even more effective expression is the dBase *iif* statement. The structure of the iif statement, for example, `iif((trim(contact1->key1)="Sellers"),"Sellers","Buyers")` reads as follows: **If** the statement is true (`contact1->key1="Sellers"`), **then** display the Sellers record type, **else** (if statement is not true) display the Buyers record type.

### Simple iif Expression

Assuming Key1 can have Sellers or Buyers as a value and assuming you have record types named Sellers and Buyers, you can use the following expression:

**iif((trim(contact1->key1)="Sellers"),"Sellers","Buyers")**

**Results:** For each record where the Key1 field value equals Sellers, the Sellers record type is displayed; otherwise the Buyers record type is displayed.

### Complex iif with .AND. Expression

Assuming Key1 can contain the value Sellers or Buyers and Key2 can contain Insured or Uninsured and assuming you have four record types named Sellers Insured, Sellers Uninsured, Buyers Insured, Buyers Uninsured, you can use the following expression:

```
(iif((trim(contact1->key1)="Sellers").and.(trim(contact1->key2)="Insured"),"Sellers Insured","Sellers Uninsured"))
```

**Results:** If the value in Key1 is Sellers and the value in Key2 is Insured, then the Sellers Insured record type is displayed for the record. When you move to another record where the value of Key1 is Sellers and the value in Key2 is Pending, then the Sellers Uninsured record type is displayed for the record (because Key2 is not equal to Insured).

### Complex Multiple iif Expression

Assuming Key1 can have Sellers, Buyers, Appraisers or Lenders as a value and assuming you have record types named Sellers, Buyers, Appraisers, Lenders, and Main, you can use the following expression:

```
iif((contact1->key1)="Sellers","Sellers", iif((contact1->key1)="Buyers","Buyers", iif((contact1->key1)="Appraisers","Appraisers", iif((contact1->key1)="Lenders","Lenders","Main"))))
```

**Results:** If the value in Key1 is Sellers, then display the Sellers record type. If the value is not Sellers, evaluate whether Key1 is Buyers. If the value is Buyers, then display the Buyers record type; otherwise, evaluate whether Key1 is Appraisers. If the value is Appraisers, then display the Appraisers record type; otherwise, evaluate whether Key1 is Lenders. If it is Lenders, then display the Lenders record; otherwise, display the Main view record type.

## New Record Creation Options

The *New Record Creation Options* button on the *Record Type Settings* window allows you to customize a *New Record* window for each of your Record Types. It will also create a new option under the *File>>New Record* menu options for creating new records as a specific Record Type.

**NOTE:** Keep in mind that if you have enabled *dBase Expression result-based* rules, the new menu option will not be added under *File>>New Record* menu. You must use *Field value-based* rules.

1. In the Record Types Administration Center, right click on any of your Record Types and choose **Edit**.

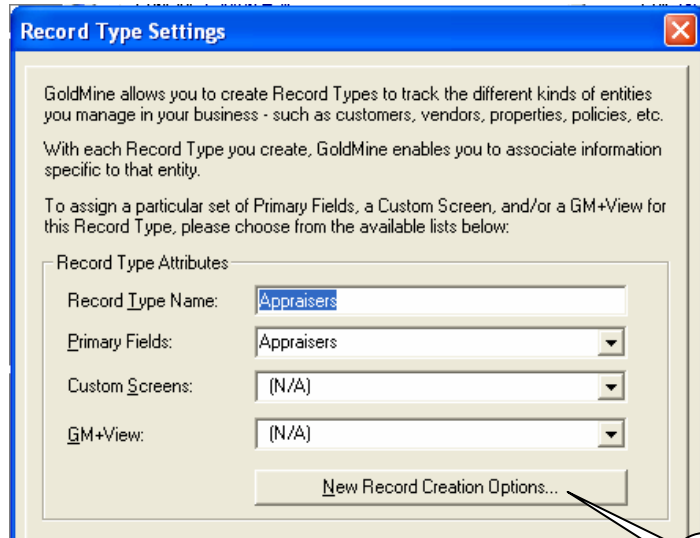


Figure 16

2. Click the **New Record Creation Options** button.
3. First, click the *Use a type specific new record dialog* checkbox to activate the list of fields.

The Record Type is reflected here.

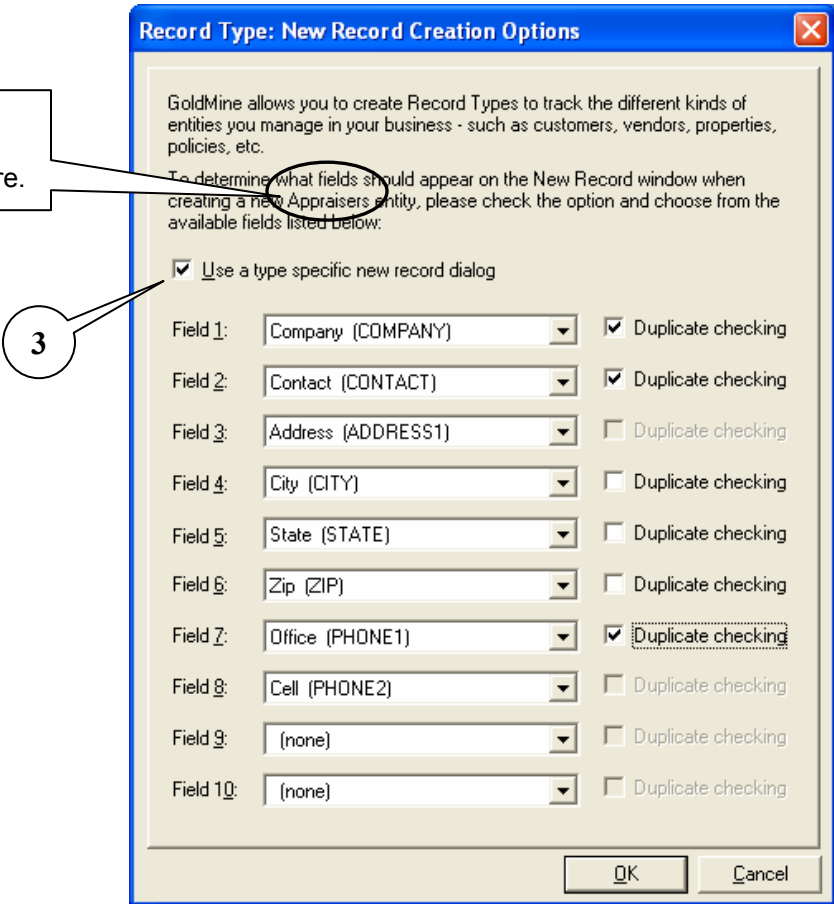
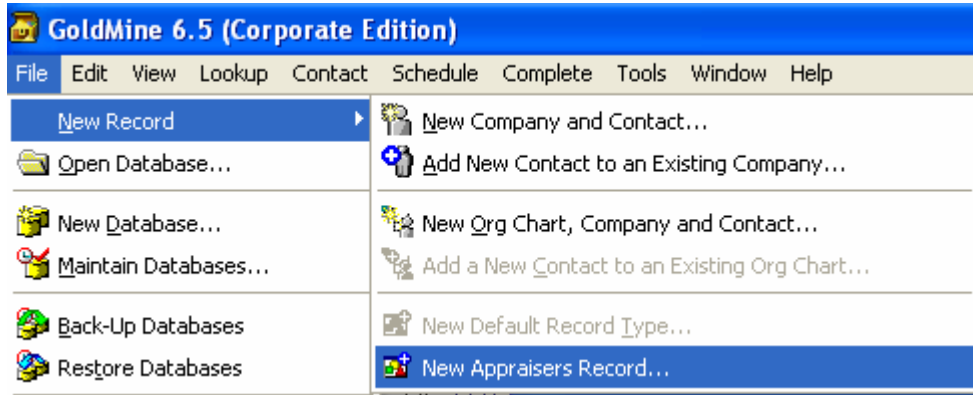


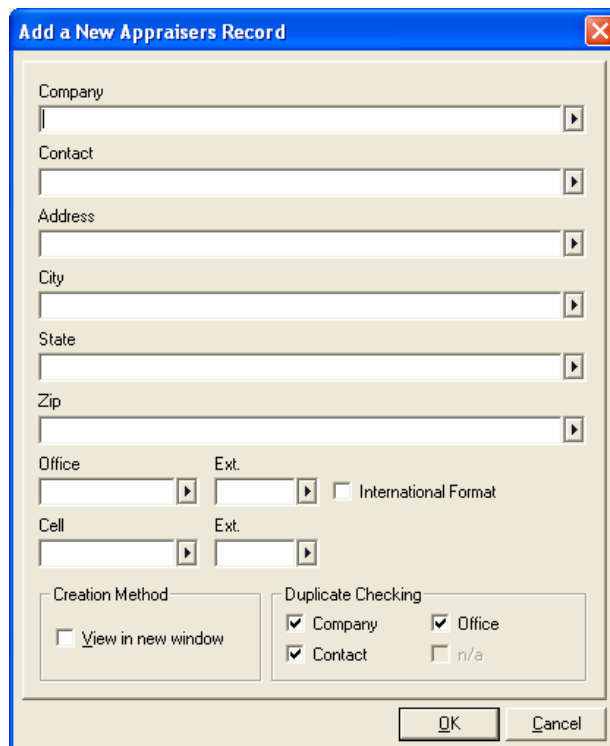
Figure 17

4. You are allowed to choose up to 10 fields for the New Record Creation Window that will be specific to this Record Type. Select the fields from the dropdown list, in the order you'd like them displayed. You can also choose to have GoldMine perform duplicate checking on the fields you have selected.
5. Click **OK** after all selections are complete.
6. Click **OK** again to return to the Record Types Administration Center.
7. Click **Apply Changes**.
8. Figure 18 shows the new option for the Record Type under the File>>New Record menu.



**Figure 18**

9. Selecting this option opens a New Contact window (shown in Figure 19) displaying the fields selected. The title bar reflects the specific Record Type for this new record.



**Figure 19**