

In This Chapter

GM.ini

Contact_Info.html

Process.asp

E-mail Rule

WebImport - Script Generator

DJ's Registration WebImport

Note

If you have purchased a paperback copy of this book, please forward a copy of your paid Invoice to:

DJ@DJHunt.US

Once I have proof of purchase, you will be sent the eBook version along with all of the documents at no extra charge.

Remember that the eBook version is Searchable, whereas the printed version can only be searched page by page.

WebImporting is a feature that has been included in the GoldMine application for some time, albeit, unsupported by FrontRange Solutions. In its first incarnation, WebImporting only allowed the users to add new contacts to their GoldMine from a web site form. One could also have the users update their GoldMine record in your database via an e-mail message (an example that I will discuss in this chapter). Computerese Incorporated also uses the WebImport script to generate e-mail messages that notify us that a user wishes to schedule a meeting. We also notify product vendors when a visitor has viewed their product on our web site, and requests additional information. The possibilities for WebImport are endless, and, again, you are only limited by your imagination. Alas, not many utilize this feature capability within GoldMine. If you maintain a web site, then there is no excuse for not capturing any registration leads from that web site directly into your GoldMine.

FrontRange supplied us with a sample web form, and the accompanying PERL script. We could modify the form to fit in with the theme of our web site, and place the PERL script on our web site for processing, assuming that your web site was capable of processing PERL (Linux Servers). Unfortunately, I was using a Windows Server so, as you'll see, my scripts needed to be written in ASP. The important thing to understand is the functionality of the script, and then you can script the code in any language with which you are familiar.

FrontRange has incorporated into GoldMine Premium the GoldMine WebImport - Script Generator. I have also supplied my own files to accompany my descriptions/definitions in this chapter. When you downloaded this book, the WebImportFiles.zip were included as part of the eBook download. If you've lost them, or you ordered the paperback version of this book, you may E-mail me directly and request the book inclusions. The file contains the documents that I will be discussing in detail in this chapter. **The Contact_Info.htm** document, and the **Process.asp** document to be specific.

Maybe I should explain the functional attributes of these documents before I actually get into them, and analyze them in some detail. I never, ever use the GoldMine HTML Editor for my E-mail Templates, or the GM+Views HTML Editor for that matter, as it does not maintain a legible structured format. Once created in GoldMine, I never know where to go in, and where to make any changes that I require. I, therefore, create all of my HTML documents in DreamWeaver, and then use the GoldMine **include a text file** feature to include a pointer in our HTML document in the body of the GoldMine E-mail Template or for the GM+View. The actual pointer in the body of my GoldMine template looks like this:

```
<<file:Y:\GoldMine\HTML Templates\DJ Blue Contact Info.htm>>
```

When I use this template as my E-mail template for any client, it pulls all of the contacts information from the currently active record in GoldMine, and sends that information to the contact for review. They are asked to change any information that is not correct, and **Submit** the form to have the corrections made automatically back in my GoldMine. Alternatively, they may select that the information is correct as it stands, and I will receive an E-mail stating such, or they may even select to be removed from my data set. In this case I would also receive an E-mail requesting that they be removed from my GoldMine. Other than the E-mail to me, this step is not automatic for, I hope, obvious reasons.

The form, as I sent it to myself, is displayed in Figure 9-1 on the next page. Now, this form, could just as easily have been put on my web site, without the user information being pushed from GoldMine, and any end user could then have registered, updated their information, or even supplied their information, and asked to be removed from our data set with only slight modifications. Again, if the registrant were new to our data set, a record would be created automatically. If the registrant were in our data set, the information could automatically update their information. A removal request, would send us an

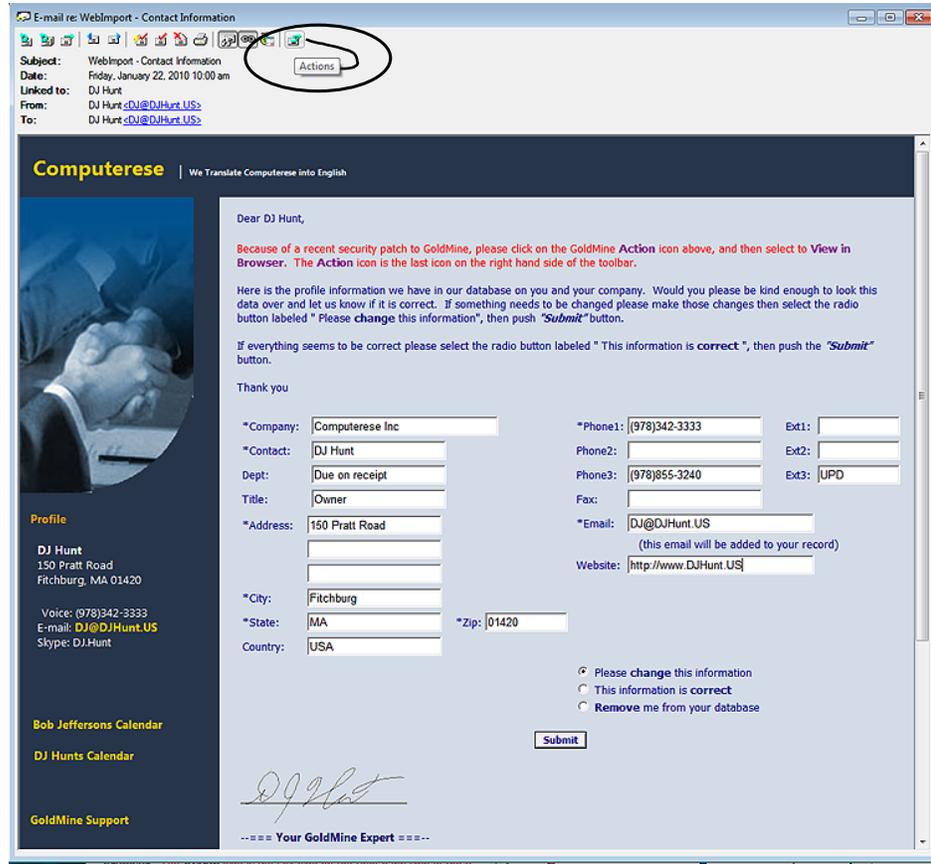


Figure 9-1

E-mail requesting to be removed. This action, is not automatic, and must be performed by the recipient of the message. I chose to review it as an E-mail template, as it is something that you could use in your practice as well, and as I said, it is easily converted to a web site form by just changing the duplicate checking statement for AccountNo to Email.

GM.ini

First, let's talk about the updatability of the GoldMine contact record. Having the ability to update a contact record, via a WebImport identified E-mail, was available as far back as in GoldMine 6. To use this ability, the GoldMine Administrator, must first tell GoldMine that they are willing to allow records to be updated, and specifically which **Contact1/Contact2** fields that they will permit to be updated via the WebImport message. This is accomplished through the global GoldMine instruction set, the **GM.ini**. I will show you an easier way to work with the GM.ini later on in this chapter, but for now, here are the necessary components along with an explanation.

Note
This information is only indented/formatted for presentation purposes in this book. The actual **GM.ini** is modified via **NotePad** and must contain no formatting what-so-ever.

In the GM.ini, the GoldMine Administrator must add a new section, as shown in my example here:

```
[WebImportOverwrite]
IAgree = I Want To Allow Fields To Be Updated By WebImport
Fields = Company, Contact, Department, Title, Address1, Address2, Address3, City, State, Zip, Country, Phone1, Phone2, Phone3, Fax, Ext1, Ext2, Ext3
```

The **[WebImportOverwrite]** section was newly added in GoldMine 6, and the first statement in this section leaves me a little baffled. Why would I even add this section if it were not my intent to update GoldMine records? However, the line is required **exactly** as is. That is spaces where I show you there are spaces, and no spaces in the variable, **IAgree**.

The **Fields =** statement, identifies those **Contact1/Contact2** fields that, you as the Administrator, will allow to be updated externally. Only those fields that are contained in this list will be considered to be updatable by the GoldMine application. Here we encounter a potential problem. The field list must be contained in one continuous line in the **GM.ini**. There is a 256 character limit to a single line in an ini file. Remember that we ran across that limitation when I was discussing the **Lookup.ini** earlier. Therefore, if you have many field names, and you wanted to be able to update a lot of fields through the WebImport, you would need to make certain that each line is under 256 characters. You may add consecutive **Field(x)** statements if you have more fields than can be included in a single statement.

Note

Had this been done using the GoldMine WebImport Tool, the Contact1/Contact2 field names would have all been in caps.

WARNING

Some features of WebImport in GoldMine Premium now require that you utilize a **Password**. In fact, when using the built in tool in GoldMine Premium to modify the **GM.ini**, the **OK** button is disabled until you have entered a **Password**.

WARNING

The **Password=** statement of the **[WebImpPassword]** section has limitations. The password may not contain spaces, and may not exceed 20 characters in length.

Although you may add additional passwords, as many as 999, in the **GM.ini**, each **Instruction Set** may only contain 1 password. (See **Process.asp** later in this chapter)

Contact_Info.htm

Note

Throughout this chapter I will be referring to documents that are included with the downloadable edition of this book. You may want to open those up in **NotePad** to follow along.

Process.asp

Here is an example of just such a situation:

[WebImportOverwrite]

IAGree=I Want To Allow Fields To Be Updated By WebImport

Fields=phone1, phone2, phone3, fax, ufg1, ufg2, ufg3, ufg4, ufg5, ufg6, ufg7, ufg8, ufg9, ufg10, ufin-goals, umg1, umg2, umg3, umg4, umg5, umg6, umg7, umg8, umg9, umg10, umgtgoals, uredux, ulnlife, upcomm, upcont, source, umaxdown

Fields1=urlf,ucdrpt, uinchg,uinchgexp, uastchg, uastchgexp, udbchg, ubtrm6, udbcrd1, udbcrd2, udbcrd3, udbcrd4, udbcrd5, udbcrd6, udbpurp1, udbpurp2, udbpurp3, udbpurp4, udbpurp5, udbpurp6

Fields2=upaid, upaidexp, uregls3, uregls5, uregls10, ucpa, ucpa, ucfp, ucfpr, uins, uinsr, urltr, ubnk, ubnkr, urefneeds, uskip, uothermtg, uotherdebt, urinsco, email, notes

If you are planning on passing a password via your asp, then you must make sure that GoldMine is prepared to receive that password. The **GM.ini** section, and statements for that might be something like:

[WebImpPassword]

Password1 = ShowBoat
Password2 = Musical

...

Password999 = Show

Now I will take a look at the form that I will be using, the **Contact_Info.htm** form. As I eluded to earlier, this could just as easily have been a form on a web site as it is a form in our E-mail message.

The **Form Action** is an extremely important part of the form HTML code as it tells the action to be taken when the user clicks on the **Submit** button of the form.

```
<form action="http://DJ-Hunt.com/DWSite/Asps/Process.asp" method=POST name=form1 OnSubmit="return validateForm(this)">
```

This code could actually be truncated to:

```
<form action="http://DJ-Hunt.com/DWSite/Asps/Process.asp" method=POST>
```

Here is where I instruct you how the form action is to occur, and the method of the form action. In this case I am pointing the action to an asp file located on my site which will process the received form information into an E-mail message, and then send it to a GoldMine user. I will discuss the asp file in the next section of this chapter. The method that I have chosen, is the **POST** method. If the method of **POST** is not entered, then the default method of **GET** is employed. The **GET** method is not as secure as all of the data is exposed when it is added to the end of the URL. Also, our response asp is set up to read from a **POST**. It is very important that the **Form**, and **Response** relationship is maintained.

Another important element of the **Contact_Info.htm** is that each field has a name and, in this case, each field has a default value pulled from the GoldMine contact record. On a web form, these fields would not have a default value, and the user would be expected to supply all of the information. Here is some typical HTML code for a field on my form:

```
<input name="address1" type="text" value="&&&&&address1&&&&" >
```

On **Submit**, the **Address1** name will be passed with the present field value to the response asp on my site for processing.

That is basically it for the **Form** portion of the **Form/Response** set. Yes there is a lot of formatting, and script on the form that I have supplied you. Make a copy of that form, and play around with it in your favorite HTML editor. However, there is one thing that I want to make you cognizant of, in this form I am passing two GoldMine fields as hidden fields, and it is these fields that will be utilized for matching purposes. If using this form on a Web Site, this information would not be readily available, and you would want to use something like the E-mail Address for matching purposes. They are coded thusly in my form:

```
<input name="accountno" type="hidden" value="&&&&&accountno&&&&">  
<input name="email" type="hidden" value="&&&&&emailaddress&&&&">
```

Before I begin to analyze this asp, I want to dump a few prerequisites on you. Your ISP must provide some sort of mail service component, and most do already. In my case, I use the CDO mail component, and that works out quite well from my site host (Network Solutions).

Note

Please note that the instructions are passed surrounded by quotes ("). If you were building this string with concatenated elements, you would need to add to both sides of the string `chr(34)` which is the quote (").

WARNING

You may either pass:
`DupLogic=AND`
or
`DupLogic =OR`
You should never pass both.

WARNING

Statements must be one continuous line unless programmatically broken. Any wrapped lines shown here are only for eBook readability.

WARNING

Spaces in the Track name used to cause issues in prior versions of GoldMines WebImport. If you are seeing that the proper Track names are not being assigned via the WebImport, try renaming your Tracks, removing all spaces.

I instantiate the object in my asp via:

```
Set WebImport = Server.CreateObject("CDO.Message")
```

Not only must I bundle the fields that the user responds with, but I must also bundle our processing instructions. I would like to list the various instructions that could be passed back to GoldMine for processing. Each line of the instructions, and data for that matter, must be followed by a Carriage Return and Line Feed. `vbCRLF` is Visual Basic predesignated value for a Carriage Return and Line Feed.

The instruction set **must** begin with a section designator:

```
WebImport.TextBody = "[Instructions]" & vbCRLF
```

In the above example, I am concatenating a string, and placing its value into the `WebImport.TextBody` property. That was for you programmers. For you non programmers, I'm stuffing a variable with a string of characters.

Some of the instructions that one could pass after the section designator are:

Sending a **Password** if GoldMine is setup to look for one, and it should be in GoldMine Premium v9:

```
WebImport.TextBody = WebImport.TextBody & "Password=YourPassword" & vbCRLF
```

Telling GoldMine **Not to Import** any data as you would like an external application to do it instead:

```
WebImport.TextBody = WebImport.TextBody & "ImportData=0" & vbCRLF
```

Telling GoldMine to **Check for the Duplicates** based upon, and these can be compounded:

```
WebImport.TextBody = WebImport.TextBody & "DupCheck=Email" & vbCRLF  
WebImport.TextBody = WebImport.TextBody & "DupCheck1=Contact" & vbCRLF  
WebImport.TextBody = WebImport.TextBody & "DupCheck2=Accountno" & vbCRLF
```

If you are passing more than a single **DupCheck** instruction, you must tell GoldMine that you want those DupChecks handle as if they were in an **AND** or an **OR** statement:

```
WebImport.TextBody = WebImport.TextBody & "DupLogic=AND" & vbCRLF
```

or

```
WebImport.TextBody = WebImport.TextBody & "DupLogic=OR" & vbCRLF
```

Telling GoldMine to **Launch an External Application** from within the WebImport:

```
WebImport.TextBody = WebImport.TextBody & "Run= C:\WinNT\notePad.exe" & vbCRLF
```

Telling GoldMine to **Create an E-mail** for a New/Dupe record, and this too may be compounded as well:

Syntax:

```
"OnNewSendEmail=<<UserID>>, <<ActvCode>>, <<Reference>>"  
  
WebImport.TextBody = WebImport.TextBody & "OnNewSendEmail=DJ, NEW, New WebImport" & vbCRLF  
WebImport.TextBody = WebImport.TextBody & "OnNewSendEmail1=DJ, NEW, New WebImport" & vbCRLF  
WebImport.TextBody = WebImport.TextBody & "OnDupSendEmail=DJ, OLD, Old Web Import" & vbCRLF  
WebImport.TextBody = WebImport.TextBody & "OnAnySendEmail=DJ, ANY, Any Web Import" & vbCRLF
```

Telling GoldMine to **Save a Copy** of the imported message in the **ContHist** table linked to this **Contact** record:

```
WebImport.TextBody = WebImport.TextBody & "SaveThis=Web import file" & vbCRLF
```

Telling GoldMine to **Attach a Track** to a New/Dupe record, and this statement may be compounded:

Syntax:

```
"OnNewAttachTrack=<<Track Name>>, <<Attaching UserID (optional)>>"
```

WARNING

GoldMine Premium requires that you pass a WebImport Password in order for it to attach any Automated Process to a Contact record.

```
WebImport.TextBody = WebImport.TextBody & "OnNewAttachTrack=WEB Lead" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnDupAttachTrack=WEB Lead" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnDupAttachTrack1=WEB LeadA" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnDupAttachTrack2=WEB LeadB" & vbCRLF
```

That pretty much wraps up the available instructions as I know them. On the next few pages I will provide you with my entire sample asp file. I will comment on items that you will need to change.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Process Page</title>
</head>
<body>
Begin script
<%
```

I am just putting the form submitted data into variables in this section of the script. This step is not necessary, however, it does add to the readability of the code.

```
f_accountno = Request.form("AccountNo")
f_company = Request.form("Company")
f_contact = Request.form("Contact")
f_department = Request.form("Department")
f_title = Request.form("Title")
f_address1 = Request.form("Address1")
f_address2 = Request.form("Address2")
f_address3 = Request.form("Address3")
f_city = Request.form("City")
f_state = Request.form("State")
f_zip = Request.form("Zip")
f_country = Request.form("Country")
f_phone1 = Request.form("Phone1")
f_ext1 = Request.form("Ext1")
f_phone2 = Request.form("Phone2")
f_ext2 = Request.form("Ext2")
f_phone3 = Request.form("Phone3")
f_ext3 = Request.form("Ext3")
f_fax = Request.form("Fax")
f_email = Request.form("Email")
f_addedemail = Request.form("AddedEmail")
f_website = Request.form("Website")
f_change = Request.form("R1")
f_correct = Request.form("R1")
f_remove = Request.form("R1")
```

If user has selected **CHANGE**:

```
If f_change = "Change" then
Set WebImport = Server.CreateObject("CDO.Message")
WebImport.Subject = "Changes to contact record: " & f_contact
WebImport.From = f_contact & "<" & f_email & ">" ' Specify sender's address
WebImport.To = "{$GM-WEBIMPORT$}<DJ@DJHunt.US>"
```

Create the **INSTRUCTIONS** section of your WebImport:

```
WebImport.TextBody = "[Instructions]" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "DupCheck=Accountno" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnNewSendEmail=DJ, NEW, New Web Import" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnDupSendEmail=DJ, OLD, Old Web Import" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "SaveThis=Web import file" & vbCRLF
```

Creating your **DATA** section of WebImport:

```
WebImport.TextBody = WebImport.TextBody & "[Data]" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "AccountNo=" & f_accountno & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Company=" & f_company & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Contact=" & f_contact & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Department=" & f_department & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Title=" & f_title & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Address1=" & f_address1 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Address2=" & f_address2 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Address3=" & f_address3 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "City=" & f_city & vbCRLF
WebImport.TextBody = WebImport.TextBody & "State=" & f_state & vbCRLF
```

WARNING

Statements must be one continuous line unless programmatically broken. Any wrapped lines shown here are only for eBook readability.

WARNING

The GoldMine WebImport is incapable of properly handling imbedded carriage returns and line feeds when using the **ini** type of WebImport, and will stop processing the **Notes** at the first instance of a **vbCRLF**. As you cannot prevent users from using the carriage return, you may want to remove the **vbCRLF**, via your script, from the **Notes** string before parsing it into the **[Data]** section.

Note

I cover the **Notes** here, not because I use them via this E-mail WebImport as I don't, but because you could easily convert the E-mail WebImport into a Website WebImport into GoldMine where you would want to capture Comments and/or Notes. The **Process.asp** could almost be utilized as is, while the **Contact_Info.htm** need only be modified to remove the GoldMine feeds, and to fit your website theme.

Note

ContSupp records are **not** updatable via the WebImport, hence, one must create a **New ContSupp** record via WebImport.

As of my last test, you could only create one **New ContSupp** record for each **ContSupp.Contact** type per WebImport. I.e. Don't try to create two **ContSupp.Contact = [Web Site]** during the same WebImport process.

Your mileage may vary, but let me know if it does please, by E-mailing me at DJ@DJHunt.US.

Note

Notice in the **Correct** and **Remove** sections that the **WebImport.To** is different than our normal WebImport. To statement as these two only need to be messages to the GoldMine UserID, and are not really a WebImport.

WebImport:

```
WebImport.To = "{$GM-WEBIMPORT$}
<DJ@DJHunt.US>"
```

E-mail Message:

```
WebImport.To = "DJ@DJHunt.US"
```

```
WebImport.TextBody = WebImport.TextBody & "Zip=" & f_zip & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Country=" & f_country & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Phone1=" & f_phone1 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Ext1=" & f_ext1 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Phone2=" & f_phone2 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Ext2=" & f_ext2 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Phone3=" & f_phone3 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Ext3=" & f_ext3 & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Fax=" & f_fax & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Email=" & f_addedemail & vbCRLF
```

I need to break up the **Process.asp** here to be able to explain a bit about **Notes** even though the **Contact_Info.htm** that accompanies this ASP does not include any Notes, there will be cases where you need to compensate for them. Notes sent via WebImport must be broken up into chunks of 255 characters. I show you a sample of syntax below.

We'll have to assume that I have set up a variable **f_Notes** to capture the **Form** field **Notes**, and now the script must begin parsing same:

```
If Len(trim(f_Notes)) > 0 then
    WebImport.TextBody = WebImport.TextBody & "Notes=" & mid(f_Notes, 1, 255) & vbCRLF
End If

If Len(trim(f_Notes)) > 255 then
    WebImport.TextBody = WebImport.TextBody & "Notes1=" & mid(f_Notes, 256, 255) & vbCRLF
End If

If Len(trim(f_Notes)) > 511 then
    WebImport.TextBody = WebImport.TextBody & "Notes2=" & mid(f_Notes, 512, 255) & vbCRLF
End If

If Len(trim(trim(f_Notes))) > 766 then
    WebImport.TextBody = WebImport.TextBody & "Notes3=" & mid(f_Notes, 767, 255) & vbCRLF
End If
```

Creating a new **ContSupp** record via the WebImport:

```
WebImport.TextBody = WebImport.TextBody & "[ContSupp]" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "cs1_RecType=P" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "cs1_Contact=Web Site" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "cs1_ContSuppRef=" & f_website & vbCRLF
WebImport.TextBody = WebImport.TextBody & "cs1_Zip=0000"
```

```
On Error Resume Next
WebImport.Send
```

```
If Err <> 0 Then
    Response.Write "Error encountered: " & Err.Description
End If
```

Response.Write "Thank you for your help. Your changes will be made to our database."

User checked **CORRECT** option:

```
Elsif f_correct = "Correct" then
    Set WebImport = Server.CreateObject("CDO.Message")
    WebImport.Subject = f_contact & " said no corrections required"
    WebImport.From = f_contact & "<" & f_email & ">" ' Specify sender's address
    WebImport.To = "DJ@DJHunt.US"
    WebImport.TextBody = f_contact & " states that as of " & Date
    & " their contact information is correct"
```

```
On Error Resume Next
WebImport.Send
```

```
If Err <> 0 Then
    Response.Write "Error encountered: " & Err.Description
End If
```

Response.Write "Thank you for your help"

User checked the **REMOVE** option:

```
Else
    Set WebImport = Server.CreateObject("CDO.Message")
    WebImport.Subject = f_contact & " asked to be REMOVED"
    WebImport.From = f_contact & "<" & f_email & ">" ' Specify sender's address
    WebImport.To = "DJ@DJHunt.US"
    WebImport.TextBody = f_contact & " states that as of " & Date
    & " their contact information should be REMOVED"
```

E-mail Rule

Note

If you purchased the printed copy of this book, then make sure that you send me the receipt as you are entitled to a free pdf copy of this book as well with all of the related sample files.

WebImport - Script Generator

Note

Please note, in GoldMine Premium, that you will not enable the **OK** button unless you have included a **...password**: Although, passwords are only required to enable certain features, to utilize the **Web Import Profiles** tool you must utilize a password.

```
On Error Resume Next
WebImport.Send

If Err <> 0 Then
  Response.Write "Error encountered: " & Err.Description
End If

Response.Write "Thank you for your help. You will be removed from our database."

End If
%>
</body>
</html>
```

The last item that I needed to take care of is an **E-mail Rule** for this recipient, just in case. The E-mail Rule in GoldMine under this recipients account, in this case **DJ@DJHunt.US**, would be something on the order of:

```
Incoming Rule : WebImport
IF
  Subject Contains $GM-WEBIMPORTS$ OR
  To Contains $GM-WEBIMPORTS$
THEN
  WebImport
```

...as show in the GoldMine E-mail Rule Center English Script.

This, then, just about wraps up our entire discussion on **WebImport**. WebImporting, and updating is an absolutely great feature within GoldMine.

In its continuing effort to add functionality to GoldMine, FrontRange had added a new utility to GoldMine as of the GoldMine 6.6 product build, and it is known as **Web Import Profiles**. This utility allows one to create, via a GUI, a WebImport entry form along with its associated script file. The forms can be used directly from the FrontRange web site on a temporary basis, and/or can be e-mailed to the creator for modification to be incorporated, let's say, into the users theme.

I would like you to look at this new utility as I do, and I will walk you through the creation of a single Web Import Profile. Click on

[Web Setup Web Import...](#)

(don't you just love the way that FrontRange keeps varying its usage of the term **Web Import/WebImport** throughout GoldMine?) to bring up the dialog form shown here in Figure 9-2.

Figure 9-2 shows a changed dialog form when compared to the dialog form in previous editions of GoldMine as the **More >>** button no longer exists. One form size fits all now a days.

I'm going to first cover the **Password is required to use Web Import** frame where there is a single field, and it is a required field if you wish to utilize the features of this dialog form. You must add a **Web Import authorization** password in order to enable this dialog form, and its many features. If this had been a first time use of this dialog form, then the word **Change** would have actually been the word **Save**. This dialog form read my GM.ini, hence, the dialog form is currently **Enable**. If the caption on the button had been **Save** then once the 1st letter of the password had been entered into the field, in this case **G** for **GoldMine**, all features would have been enabled automatically.

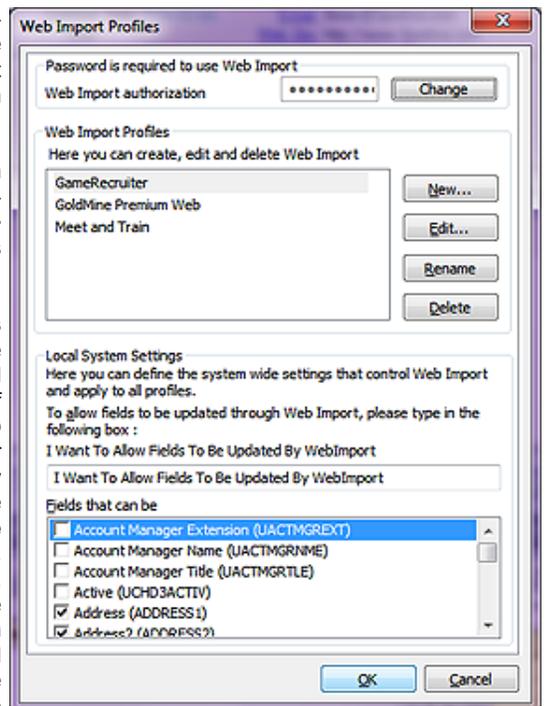


Figure 9-2

So why have a **Save** button at all for the password? That is an excellent question. You see the

developers are allowing you a way to configure your GM.ini via the GUI which is a very nice feature. After you have saved your password the **Save** button alters to a **Change** button as I had previously mentioned, and the ability to input another password is disabled. Various warnings are thrown at you when clicking upon these buttons.

Clicking upon the **Save** button will cause the below section to be incorporated into your GM.ini, and, although you have to add them by hand, I have also shown you how to add up to 999 addition passwords:

```
[WebImpPassword]
password=GoldMine
Password1 = ShowBoat
Password2 = Musical
...
Password999 = Show
```

Mind you, I have not actually tested whether the WebImport feature will honor these additional passwords as of this writing, however, this has always worked in the past, and I see no reason for the developers to have changed this feature.

Now I would like to discuss the **Local System Settings** frame. This component will manipulate your **GM.ini** as did the **Password is required to use Web Import** frame component. Earlier in this chapter, I showed you the required settings in the GM.ini for the proper functioning of the WebImport when wanting to update fields in your GoldMine. With the advent of this GUI, FrontRange has made that job a whole lot easier.

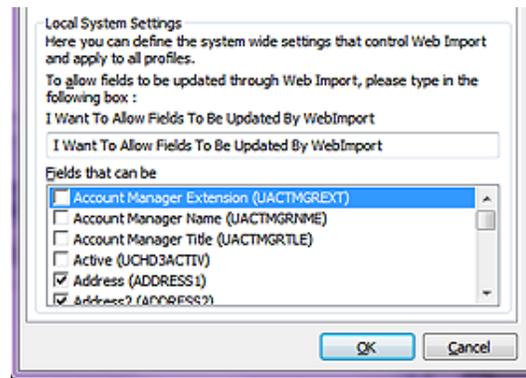


Figure 9-3

Unlike previous editions of this tool, you do not have to complete the field: **I Want To Allow Fields To Be Updated By WebImport** to enable the Fields that can be updated list. However, you must do this if you want any of your selected fields to be updatable. Go ahead, and type **I Want To Allow Fields To Be Updated By WebImport** into this field exactly as it is presented on the dialog form, and I do mean exactly. Caps where there are caps, and lower case where it is showing as lower case.

Having done that correctly, the fields from your **Contact1/Contact2** tables, those that are potential candidates for updating, can now be selected. You may simply select () those

Fields that can be updated by your incoming WebImport within your GoldMine.

These actions will then be represented in your GM.ini in the following manner:

```
[WebImportOverwrite]
IAgree = I Want To Allow Fields To Be Updated By WebImport
```

```
fields = ADDRESS1,ADDRESS2,ADDRESS3,PHONE2,USERDEF02,PHONE3,CITY,COMPANY,
CONTACT,KEY3,COUNTRY,UDOB,DEAR,EMAIL,EXT1,EXT2,EXT3,EXT4,FAX,KEY1,KEY2,LASTNAME,
SECR,KEY4,NOTES,PHONE1,UQBCUSTOMR,UQBVENDDOR,USERDEF10,UCEXPURE,UCHOURS,
SOURCE,UCSTART,STATE,KEY5,DEPARTMENT
```

```
fields1 = TITLE,USERDEF04,USERDEF07,USERDEF09,USERDEF01,USERDEF03,USERDEF05,
USERDEF06,USERDEF08,UWEBIMP,WEBSITE,ZIP,ULOGIN,UPW
```

Additionally, if you select more fields than can be handled by a single fields statement, GoldMine breaks the lines at the appropriate lengths, and inserts a continuation statement. This saves you a tremendous lot of character counting when typing your statements by hand. In fact, I use this tool exclusively as I don't have that many fingers, and toes to count that high. If doing this by hand, however, your statement might look like this:

```
fields = ADDRESS1,ADDRESS2,ADDRESS3,PHONE2,USERDEF02,PHONE3,CITY,COMPANY,
CONTACT,KEY3,COUNTRY,UDOB,DEAR,EMAIL,EXT1,EXT2,EXT3,EXT4,FAX,KEY1,KEY2,LASTNAME,
SECR,KEY4,NOTES,PHONE1,UQBCUSTOMR,UQBVENDDOR,USERDEF10,UCEXPURE,UCHOURS,
SOURCE,UCSTART,STATE,KEY5,DEPARTMENT
fields1 = ...
fields2 = ...
fields3 = ...
...
fields999 = ....
```

Note
You may have noticed, a space per field is a premium in an ini statement, and that this tool has not included any spaces into the statement string within the **GM.ini**.

Mind you I haven't tested the limitations of this statement area, however, going by similar rules for other sections this should work. If it is of any comfort to you, I have tested up to fields6 = for a client without issue.

One could simply click on the **OK** button, and the proper statements would be written to the GM.ini. No Web Form or ASP script would be created, but the GM.ini would be correctly formatted.

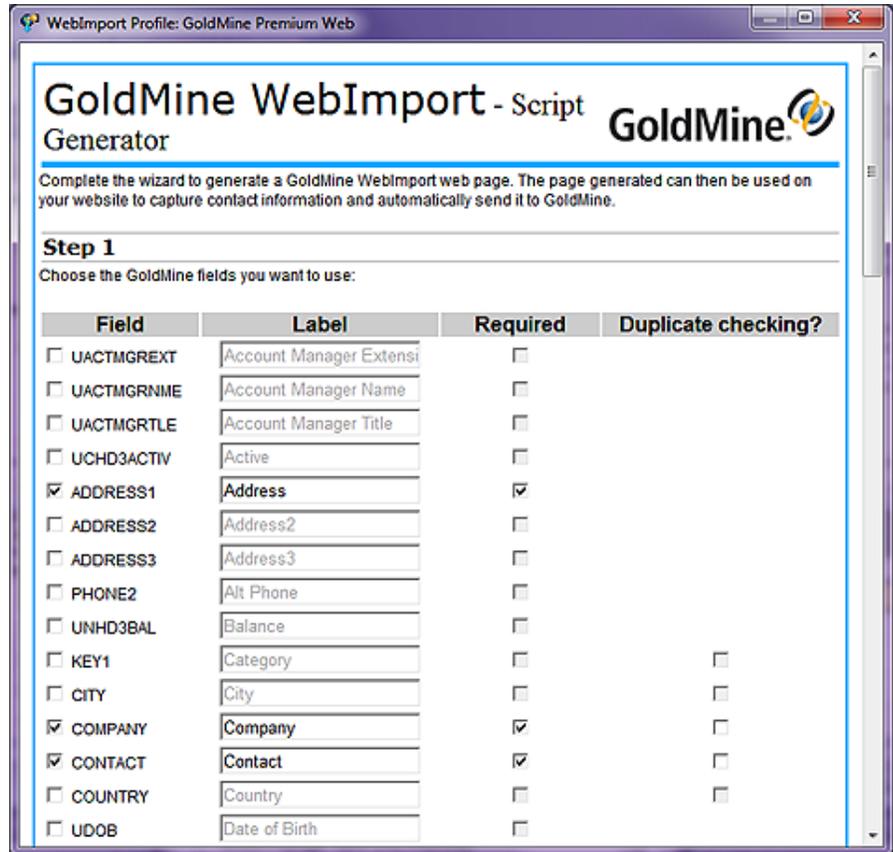


Figure 9-4

Note

You may have noticed, Figure 9-2, that I have two profiles for my clients already created, as well as the profile that I am using for this book.

Note

Did you notice that the form, Figure 9-4, did not pick up on any of the fields which you had previously selected as updatable? One would think that if one wanted those fields to be updatable, that one would also want those fields on their webform. Seems logical to me.

Note

As the **E-mail Address** is unique within the GoldMine default environment, this field alone may be sufficient enough for duplicate checking. The **E-mail Address** or the **AccountNo** information, if I have it, is all that I use for duplicate checking.

WARNING

Though it appears that one can create a label of unlimited length on the webform, only about 31 characters will actually display.

This is not my intent here, however, I am here to have the configurator develop a Web Form, and an ASP script file for this exercise. Returning now to the **Web Import Profiles** frame of Figure 9-2, and I quote, "Here you can create, edit and delete Web Import profiles". Profiles, hmmm! More than one. Yes you could develop as many different profiles as you wish. My intent here is to show you, simply, how to create a single profile. Shall we continue (a rhetorical question, I know)?

Clicking on the **New...** button creates an entry in the profiles list with **New Profile** selected. Simply begin typing in a new profile name. I have entered **GoldMine Premium WebImport**. When you have completed entry in the profile name, hit the **Enter** key, and watch what develops, see Figure 9-4 above.

You will notice that some of the fields have been preselected for you already, and, of those, some have been preselected as being **Required**. As well, in the fourth column, some have been preselected for **Duplicate checking?**. Naturally, you can alter this form at will. You must select all of the fields that you wish to have contained on your Web Form. Of those fields, you must identify those that will need to be required fields. i.e. The user will not be able to **Submit** the form from your web site, unless there is information contained within those fields selected as required. Lastly, you should select, at least, the **E-mail Address** for duplicate checking. Remember, GoldMine will check the WebImport to see if a record already exists within your GoldMine based upon your choice(s) for duplicate checking?. Depending on whether you are allowing the updating of the record when a duplicate is found, GoldMine will either do nothing or update the fields with the newer WebImport information when the WebImport is processed by GoldMine.

One thing that I haven't explained, is the second column on this form, the **Label** column. This column is where you may change the Web Form field display label that is associated with said field. You may type into the field in this column to change the associated label.

Once completed, as cannot be seen at the bottom of Figure 9-4, and similar to those shown in Figure 9-5 on the next page, there are buttons to either **Reset the Form**, **Next >**, or **Cancel** the activity.

WebImport Profile: GoldMine Premium Web

GoldMine Web Import - Script Generator

Complete the wizard to generate a GoldMine Web Import web page. The page generated can then be used on your website to capture contact information and automatically send it to GoldMine.

Step 2

Input the following preferences:
*Indicates required field.

E-mail address
This is the e-mail address GoldMine will check to retrieve the Web Import data. *

SMTP Server
Enter the IP address of an SMTP server that can be used to send the e-mail (must not require authentication.) *

Scripting language:
This is the type of web script that will be used to send the Web Import e-mail. The type of script you select must be installed for Web Import to work. *

New contact alert
If you want an e-mail sent to a GoldMine user whenever Web Import creates a new contact, select that user here.

Duplicate contact alert
If you want an e-mail sent to a GoldMine user whenever Web Import attempts to create a duplicate contact, select that user here.

New contact AP
If you want an Automated Process attached to each new contact Web Import creates, select that AP here.

Duplicate contact AP
If you want an Automated Process attached to each duplicate contact Web Import creates, select that AP here.

Duplicate Logic
To be considered a duplicate, the fields:

must match a single duplicate field.
 must match all fields.

Web Import format
Choose the format the Web Imports should be created in, based on the version of GoldMine in use.

INI Type(All GoldMine Versions)
 XML Type(GoldMine 6.6 or higher)

Figure 9-5

I have chosen to proceed with my development, so I have selected the **Next >** button. As a result, I am moved to **Step 2**, refer to Figure 9-5, where I am asked for additional options.

The astute reader will immediately notice that some of the fields are required as indicated by a red asterisk (*). These fields are the **E-mail address**, the **SMTP Server**, and the **Scripting Language**. Let's take a look at these options one at a time, although most of them are self explanatory.

E-mail address - This is the E-mail address to which, the developed script, will be sending any WebImport Instruction Sets to, and is the E-mail address that you will want to set up in your GoldMine to Auto-Retrieve as well as set up a WebImport E-mail Rule against.

SMTP Server - This may be a little more difficult to come by for most users, as they state that the SMTP **must not require authentication**, while virtually all SMTP's today require authentication. This is the vehicle through which the script will attempt to send the Web Import Instruction Set. You might use **Outgoing.Verizon.Net**, however, one could as easily have used **203.196.185.43** if one knew the IP address of their SMTP. Actually, there are quite a few that still do not require authentication, but it is up to you to locate one. My particular Web Hosting service, Network Solutions, allows me to use **Server.CreateObject("CDO.Message")** so I don't even have any clue as to what the SMTP IP address is, and I don't need to know, as CDO takes care of all of that for me. Talk to your ISP.

Scripting language: - Can you say redundant? Why FrontRange puts an asterisk (* Required input) next to a field that contains a drop list, and can't possible be left blank, is beyond my imagination. However, having said that, this is where one selects that scripting language to be used to compile the web import instruction set. Personally, I use a variant of the default **ASP - Using CDO**, however, the other choices that you have available are:

- ASP - Using ASPMail
- ASP - Using CDONTS
- ASP - Using CDO
- PHP
- PERL Using NET::SMTP

One or all may be possible on your server. PHP is fast becoming the scripting language de jour, while PERL is becoming passé. On the other hand, **ASP - Using CDONTS** (Microsoft Collaboration Data Objects for Windows NT Server) is associated with the arrival of IIS for NT. Again, talk to your ISP to determine which would be best for your web server.

Note

Had you followed my suggestions in Chapter 3:

*[GMAAlarm]
OnByDefault=ACTSOMDL*

The user may select any activity Rec-Type to have that activity alarmed by default. GoldMine uses the following RecTypes:

- A = Appointment*
- C = Call*
- T = Nex Action*
- D = To-do*
- M = Message (Internal)*
- S - Sales Potential*
- O = Other*
- L = Literature Request*

The M choice would cause the New contact alert or the Duplicate contact alert to pop an alarm for the requested UserID.

New contact alert - This is an item that I highly recommend that you complete. You will notice that the configurator reads your **UserID** table, and list the UserIDs here for your selection. If you don't have a UserID selected here, when GoldMine automatically reads the WebImport, and creates a new record, no one will know that a new record was added to the table unless they happen to trip over it. By notifying a user that a new record has been created, an alarmed activity is linked to the record that is created (see sidebar). No missing New Leads now.

Duplicate contact alert - Well what do you suppose this means? A duplicate, whether you choose to update or ignore, under automatic Web Import, goes unnoticed. As with the New contact alert, you may select a UserID and, upon duplicate, a Linked, Alarmed GoldMine Internal E-mail will be sent to the specified UserID (see sidebar).

New contact AP - Should you want an Automated Process attached to a New Contact record created by your WebImport, then you should select one here from your predefined processes. I always opt for my **Observer Process** as discussed in Chapter 10. If you have selected an AP to be attached to all New Contact records, as I'll discuss in Chapter 10, then the WebImport AP would attached it to the record in addition to the defined one. All of the available Automated Processes are read directly from your GoldMine, and presented here in the drop list. This is great if you are using an AP that automatically populates certain additional fields based on information contained in the WebImport or let's say you have a New Prospect Track.

Duplicate contact AP - Again we have the option of attaching an Automated Process, however, this time it is for when a duplicate WebImport has been received.

Duplicate logic - This is a radio button selection. Should you have selected multiple duplicate conditions, you must instruct GoldMine as to how to handle these duplicate conditions. You may choose one of the two possible options:

- must match a single duplicate field.** (translates to dbase **.OR.**)
- must match all fields.** (translates to dBase **.AND.**)

The default selection is shown above, where my selected duplicate fields must all be matched to be considered a duplicate record, however, I usually choose the other option as I am matching on just the required field of E-mail Address. As you know, in the default configuration of GoldMine, the E-mail Address must be unique within GoldMine, hence, a single match here is sufficient for our needs. If you'll remember on my WebImport E-mail form, I used the AccountNo as a hidden field, and as my dupcheck for the WebImport. That is because I had the AccountNo available to send hidden in the E-mail message, however, someone entering information from your Web Form will not be privy to the unique GoldMine AccountNo value, hence, you must capture something else that is unique such as their E-mail Address or a CustomerID.

Web Import format - This is your final option on this page. Again we have a two radio button choice only.

- INI Type (All GoldMine Versions)**
- XML Type (GoldMine 6.6 or higher)**

Why change what has been working well for years. I accept the default setting as shown above. Again, XML is another language de jour, and has been added to the GoldMine arsenal for just that reason.

I will now continue on with **Step 3** of the **Script Generator** by clicking upon the **Next >** button. You will notice that there is very little to be done on this next dialog form. All that one needs to do is select, or not, a template. Your choices are simple, with examples of each displayed to the right. Your

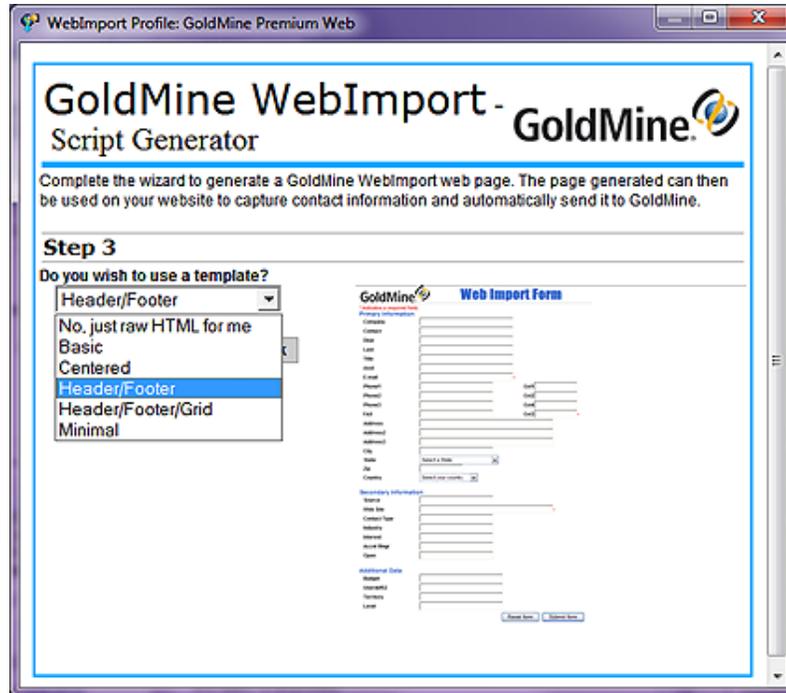


Figure 9-6

choices for Do you wish to use a template? are shown here in Figure 9-6. The default item in the drop list is **No, just raw HTML for me**.

All, except the default option, supply you with a form which you may later alter. The default option just provides the script, in your language of choice, for your reuse.

You will notice, in Figure 9-6, that I have selected, for presentation, the **Header/Footer** option. Simple logos, layout, and shading will be provided as displayed in the example. What is nice is that you can just use this form as the basis for your form in your Web Site theme schema, and you will not have to worry about passing the proper field names. This is, by itself, a blessing.

I can now, again, click on the **Next >** button to advance to **Step 4**, Figure 9-7.



Figure 9-7

You will notice that the only required information on the page is an **E-mail Address**. The E-mail Address to which you wish the script, and the form sent. The configurator will auto-populate the E-mail Address from the one which you entered earlier, however, you may change it if you wish the package to be sent to another E-mail Address.

You may click on the hotlink, **View WebImport Form**, to review the form as it has been designed, prior to having the package sent to you.

At this point, you may click to **Send the e-mail**, or **Save and Exit**. Additionally, you may choose to **Reset the Form**, to go < **Back** a page, or to simply **Cancel** the configurator entirely. I selected to **Send the e-mail**.

This result of my action can be seen in Figure 9-8. I received a nice little E-mail from **GoldMine WebImport <no-reply@frontrange.com>**, to which was attached the package that I had configured, **1064150406.zip**.

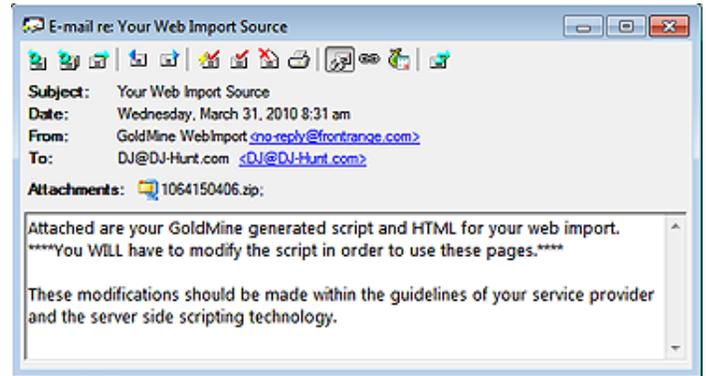


Figure 9-8

Additionally, the profile has been saved (although you may have to refresh your screen to see the **Profiles**), and may be edited at any time in the future, and/or I may create other profiles for other types of WebImport pages. In my opinion, this is a great time saver, hence, a great feature to GoldMine.

As the E-mail clearly states:

**Attached are your GoldMine generated script and HTML for your web import.
****You WILL have to modify the script in order to use these pages.******

These modifications should be made within the guidelines of your service provider and the server side scripting technology.

Live within that, and you should have the basis for your WebImport.

DJ's Registration WebImport

Well, for a couple of books now I said that one could easily convert the **Contact_Info.html/Process.asp** into a Web Site document/script. I thought that, in this edition of the book, I would include a themeless version of my very own GoldMine Support Registration document and accompanying script. The templates have been included in your eBook download, and I will document it somewhat here. A full detailed description should not be necessary as we have already described this script previously.

For this exercise the templates are **GoldMine_Support_Registration.html** & **GoldMine_Support_Registration.asp** respectively. About the only thing that I can say on the GoldMine_Support_Registration.html document is that, as we don't have anything from within GoldMine on the Web Site for the user to select from, the **E-mail Address** field is a **Required** field. Since we use the default unique E-mail Address in GoldMine, we will be able to use the E-mail Address for duplicate checking via the WebImport. Also, a few of the fields are prepopulated with the most common value, for instance: **Country**: is prepopulated with **USA**, yet is editable. The **POST** method of this form points to the GoldMine_Support_Registration.asp on my web site, and that's pretty much all that I need to address on the HTML document.

Now let's look at the actual script, and you will find this is nothing more then a simplified form of the previously discussed Process.asp.

This begins the script section:

```
<%
```

This Creates the WebImport Object for use by this Web Page. Remember that I previous stated that I used the CDO SMTP capabilities:

```
Set WebImport = Server.CreateObject("CDO.Message")
```

Here is where you would put what you will want as the **Subject:** of the e-mail message that is being sent to your GoldMine:

```
WebImport.Subject = "<< GoldMine Support Registration >>"
```

This sets the **From:** value of the e-mail message that is being sent to GoldMine. Notice that I am stuffing the From value with the E-mail Address from the Web Page form that was submitted by the registrant (the **Required** field on the Form). This way the e-mail message appears to GoldMine as if it were sent directly by the registrant.

```
WebImport.From = Request.Form("EMAIL")
```

Here we set the **To:** value of the e-mail message. Please notice that it has the special coding that we will set in the GoldMine E-mail Rule that will designate this as a WebImport message. The actual E-mail Address should be that of the auto-retrieved e-mail account set up within your GoldMine for the WebImport process. Notice that I have remarked out the second or alternate recipient.

```
WebImport.To = "{$SGM-WEBIMPORT$}<DJ@DJHunt.US>"
'WebImport.To = "{$SGM-WEBIMPORT$}<Waldo@DJHunt.US>"
```

Here we begin the construction of the **Body:** value which includes both the **Instruction Set** and the **Data Set**.

Create the **Instruction Set** section of your WebImport:

```
WebImport.TextBody = "[Instructions]" & vbCRLF
```

Define what GoldMine is to use for duplicate checking:

```
WebImport.TextBody = WebImport.TextBody & "DupCheck=Email" & vbCRLF
```

Define the GoldMine Message to be created when GoldMine retrieves the WebImport with the syntax of <UserID, ActvCode, Reference>:

```
WebImport.TextBody = WebImport.TextBody & "OnNewSendEmail=DJ, NEW, New Web Import"&vbCRLF
WebImport.TextBody = WebImport.TextBody & "OnDupSendEmail=DJ, OLD, Old Web Import" & vbCRLF
```

Create the password for this specific WebImport:

```
WebImport.TextBody = WebImport.TextBody & "Password=GoldMine" & vbCRLF
```

Attach an Automated Process, in this case only for those WebImports that are deemed New:

```
WebImport.TextBody = WebImport.TextBody & "OnNewAttachTrack=Observer, DJ" & vbCRLF
```

Create a History record in GoldMine that documents the contents of the WebImport message:

```
WebImport.TextBody = WebImport.TextBody & "SaveThis=WebImport File" & vbCRLF
```

Here is the **Data Set**. These are the **Contact1/Contact2** fields which are to be updated or populated:

```
WebImport.TextBody = WebImport.TextBody & "[Data]" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Contact="&Request.Form("NAME") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Source="&Request.Form("Source") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Company="&Request.Form("COMPANY") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Key2="&Request.Form("Industry") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Key3="&Request.Form("Market") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Address1="&Request.Form("ADDRESS1") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Address2="&Request.Form("ADDRESS2") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "City="&Request.Form("CITY") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "State="&Request.Form("STATE") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Zip="&Request.Form("ZIP") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Country="&Request.Form("COUNTRY") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Phone1="&Request.Form("DAYPHONE") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Phone2="&Request.Form("NIGHTPHONE") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Ext1="&Request.Form("EXT1") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Fax="&Request.Form("FAXPHONE") & vbCRLF
WebImport.TextBody = WebImport.TextBody & "Email="&Request.Form("EMAIL") & vbCRLF
```

As this is an **ini** type of WebImport, we are using these functions to divide the user input Web Form **Comments** into statements existing of 255 characters in length each:

```
If Len(trim(Request.Form("COMMENTS"))) > 0 then
    WebImport.TextBody = WebImport.TextBody & "Notes=" & mid(Request.Form("COMMENTS"),1,255)
    & vbCRLF
End If
```

Note

Placing an apostrophy (') as the first character of a statement is equivalent to **remarking** out that statement (turning the statement into a Comment line) in the script, hence, the color change.

```

If Len(trim(Request.Form("COMMENTS"))) > 255 then
    WebImport.TextBody=WebImport.TextBody&"Notes1="&mid(Request.Form("COMMENTS"),256,255)
    & vbCRLF
End If
If Len(trim(Request.Form("COMMENTS"))) > 511 then
    WebImport.TextBody=WebImport.TextBody&"Notes2="&mid(Request.Form("COMMENTS"),512,255)
    & vbCRLF
End If
If Len(trim(Request.Form("COMMENTS"))) > 766 then
    WebImport.TextBody=WebImport.TextBody&"Notes3="&mid(Request.Form("COMMENTS"),767,255)
    & vbCRLF
End If

```

These fields we are stuffing with fixed data, and this information **does not** come from the Web Form:

```

WebImport.TextBody = WebImport.TextBody & "uWebImp=Y" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "MergeCodes=ME" & vbCRLF
WebImport.TextBody = WebImport.TextBody & "CreateOn=" & Date() & vbCRLF
WebImport.TextBody = WebImport.TextBody & "udRun=" & Date() & vbCRLF
WebImport.TextBody = WebImport.TextBody & "ucRunTime=06:00" & vbCRLF

```

Insert a blank line after the end of the **Data Set**:

```

WebImport.TextBody = WebImport.TextBody&" " & vbCRLF

```

Here is the start of the Contact Supplemental Data. You need to number each of the following of each section. For example, all of the cs1_ would go together and then cs2, etc...

```

if Len(trim(Request.Form("URL"))) > 0 then
    WebImport.TextBody = WebImport.TextBody & "[ContSupp]" & vbCRLF
    WebImport.TextBody = WebImport.TextBody & "cs1_RecType=P" & vbCRLF
    WebImport.TextBody = WebImport.TextBody & "cs1_Contact=Web Site" & vbCRLF
    WebImport.TextBody = WebImport.TextBody & "cs1_ContSupRef=" & Request.Form("URL") & vb-
    CRLF
end if

if Len(trim(Request.Form("MSN"))) > 0 then
    WebImport.TextBody = WebImport.TextBody & "cs2_RecType=P" & vbCRLF
    WebImport.TextBody = WebImport.TextBody & "cs2_Contact=E-mail Address" & vbCRLF
    WebImport.TextBody = WebImport.TextBody & "cs2_ContSupRef=" & Request.Form("MSN") & vb-
    CRLF
end if

'WebImport.TextBody = WebImport.TextBody & "cs3_RecType=P" & vbCRLF
'WebImport.TextBody = WebImport.TextBody & "cs3_Contact=Beyond Gold" & vbCRLF
'WebImport.TextBody = WebImport.TextBody & "cs3_ContSupRef=This is a test" & vbCRLF
'WebImport.TextBody = WebImport.TextBody & "cs3_Title=President" & vbCRLF
'WebImport.TextBody = WebImport.TextBody & "cs3_Dear=Dear" & vbCRLF
'WebImport.TextBody = WebImport.TextBody & "cs3_Notes=These are the notes for the special Detail"
& vbCRLF

ON Error Resume NEXT
WebImport.Send
IF Err.number <> 0 THEN Response.Write( "There was an error while sending your message" & err.
message )END IF
%>

```

Compare this with the previously mention Process.asp, and you'll notice that there is very little difference with even less difference between the two HTML forms where one is for an e-mail message, while the other is for a Web Site.

Go for it!

