

## A Strategy Discussion: GoldBox for Imports, Updates and Import/Updates

This document is a summary of the strategy that I use when designing GoldBox Import/Update procedures. A second document that goes more into the nuts and bolts details is in the works; it will be an extension of this one. Watch my "Getting GoldBox" page for its availability, or E-mail me for notification.

---

**Imports** create **NEW** GoldMine Contact records; or **NEW** GoldMine Tab records. Much of this paper could be applied to Tab records; but I will focus on Main Contacts. The **fields** of a **Source file** are **Mapped** to the appropriate **GoldMine fields**, using the **Mapping** screen. Expressions (which may be Mapped in lieu of Source fields) are also controlled on the Mapping screen. "Pure" Imports do not require Matching selections.

**Updates** use Matching techniques to probe for the existence of GoldMine records that **ARE FOR THE SAME PERSON** as the Source record (in the case of Main Contact records); or are **FUNCTIONALLY IDENTICAL** to the Source record (in the case of Tab records). When a match is found, the Mapped Source field(s) or Expression normally over-writes the GoldMine data in the field to which it has been Mapped. However, **Update expressions** can (optionally, more later) be applied to determine what data will remain in GoldMine after the Update is complete (i.e. data that was already in GoldMine; Mapped data from the Source file; or some other, perhaps translated, form of data). Also optionally, a History record can be created whenever a change is made to an existing GoldMine record. In that case, **ALL** the changes made to each record's GoldMine data are included in History records. "Pure" Updates do not create new records.

**Import/Updates** function **BOTH** as Imports and as Updates; if Matching fails for a Source record, it is added as new. GoldBox's Setups are designed to be defined as Imports; as Updates; or as Import/Updates. If you set up a procedure without defining matching to your Target table, it will be a pure Import. If you set up a procedure with matching, but without checking the **ADD No-matches** box, it will be a pure Update. If you set up matching **and** check the **ADD No-matches** box, it will be an Import/Update.

---

In this paper, we'll be discussing the following major topics:

1. **Matching** – the most important topic of all
  2. **Mapping** – the other most important topic
  3. **Expressions** – get to know GXTranslate!
  4. **Stacking Setups** – automating for best results
- 

1. **Why is Matching the most important topic of all?** It doesn't matter at all for a "pure" Import, of course. But if you are doing an **Update**, your Matching Expression must be **safe** enough (an explanation of "safe" in a moment), or one or both of the following will happen:

- you'll get duplicate records (which can cause long term problems); and/or
- you may inadvertently combine the records of two or more different people (a real disaster). Or, for Tab records, you may mix the contents of two different kinds of records (different Detail Types, say).

All this can happen even if you get the Mapping perfectly; if all your Expressions are correct; and, of course, it can happen at amazing speed, and without immediate detection, via automation. In addition to taking proper care in constructing your matching scheme, the most powerful way to fight such problems is to use **multiple Custom matching schemes** (by running multiple Setups).

**Why multiple Custom matching schemes?** A primary objective of any Update is to find the **greatest possible number of matches** with GoldMine, while **NEVER** finding even one **FALSE** match. **False matches are disasters that must be avoided at all costs**, so all Matching Expressions used must be as "**safe**" as possible. "**Safe**" means the Match Expression calls for enough specific information that there is no possibility the records of different people would have exactly the same Match data.

In the real world, that means that almost **ANY** single (safe) Matching scheme will inevitably miss some matches, and (if run alone, as an Import/Update) ultimately create some dupes. This is due to the fact that the fields used in **ANY** matching scheme are almost never 100% populated in both the Source file and in GoldMine...and that's what's required for "safe and perfectly efficient" Matching.

The next best thing is to make multiple passes of the database, **with each pass trying a different safe combination of fields for matching**. This way, a record missing data in one combination of fields may have full data in another. Each of these matching schemes must be as safe as you can possibly make it, which means each will miss some matches. So, to get the highest quality matching, multiple matching schemes are often required.<sup>1</sup>

And why **Custom** matching? GoldBox has two other kinds of matching: **SQL query matching** (which is useful when the Source file is small relative to a large GoldMine SQL database); and **GoldMine Index matching**, which is only useful when a "perfect matching field" (like GoldMine's Accountno) is available in the Source file, and can be mapped to a **GoldMine Indexed field** with corresponding data.

**Custom matching** is best in all other situations, because it enables using GoldBox's "Smooth" functions. These table-based functions virtually convert certain fields of both the Source data, and the GoldMine data, into simplified versions of the original data. Then, GoldBox attempts to find a match. The purpose of the Smooth functions is to remove (or convert to abbreviations) verbiage that often results in matching failure. But they do this **only** virtually; none of the simplifications ever make it into the actual data<sup>2</sup>. Because these functions are table-based, we retain ultimate control over how they work. **GoldBox's Custom matching with Smooth functions is the safest and most effective matching you will find for GoldMine data.**

As we proceed, we'll be looking at different kinds of Expressions. A **Custom Matching Expression** is **ALWAYS** a **literal** expression. That means that it must evaluate to a literal string of characters:

**SmoothComp(TGT->Company, "C") + SmoothName(TGT->Contact) + PhoneStrip("TGT->Phone1")**

evaluates (for a well known Demo record) to the literal expression:

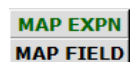
FRONTRANGE SOLUTIONS INC                      LAUREN HALL                      8007867889

2. **Mapping** is the other most important topic. It is the actual linkage between your Source file's field data, and GoldMine's fields. Without it, nothing happens. You can direct link field-to-field; you can modify values via Expressions; even fabricate them entirely. More about Expressions in the next section.

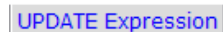
The Mapping screen is a busy place, filled with columns and rows and buttons. I'll give a detailed account of that screen; and a typical work flow outline; in the Details document mentioned earlier. Even with these, a good knowledge of GoldBox's 200+ function library may be necessary to know how to handle every situation that may arise.

I don't want to mislead you; there are a lot of details involved with the Mapping screen. But if you want to begin creating you own GoldBox Import/Update Setups, keep this in mind: if you'll allow yourself ample time to get the job done, you'll probably be able to do it. Most questions can be answered via Help; or by the GoldBox Forum, if you can state your question clearly enough. So, if you have a big Import or Update or Import/Update coming up, allow yourself at least a week or two to get the Setup done. If you have less time than that, it would be best to hire a GoldBox pro, if possible.

Before discussing Expressions further, here's a key thing to understand about the Mapping screen: it contains 1 kind of button for Mapping Source fields, and 4 kinds of buttons for Mapping Expressions.



The bottom of these two Mapping screen buttons is the **only** button that deals with actually mapping a Source field. The top button allows Mapping an Expression in place of Mapping a field. You use only one of these two buttons for any given field.



When you want to create and enter an Expression that will modify how or whether the data that was mapped in the previous paragraph will go into GoldMine **when an Update match has been recognized**, you can enter that expression here.



These two buttons are pre-configured Update Expressions. **Top:** GoldMine will only be Updated if the GXI field contains a non-empty value. **Bottom:** GoldMine will only be Updated if the GoldMine field is currently empty.

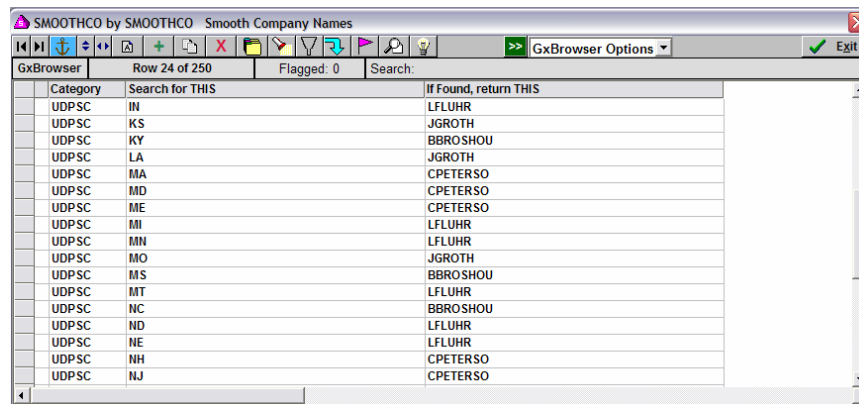
<sup>1</sup> The exception is when you have a perfect matching field, like GoldMine's Accountno.

<sup>2</sup> But see my Data Conditioning paper; and the bottom of the next page; for an exception.

3. **Expressions** fall into one of several categories:

- **Data preparation:** perhaps you have a Source phone field with data that comes in a variety of different phone number formats, and you'd like for them all to be entered as GoldMine's USA format. GoldBox has a function that will ensure that you get what you want. Example: **FmtPhoneToUSA(Source->Phone1)**. Another example: dates that are received as Character Type data. GoldBox has special programming to help you make the correct selection so GoldBox will convert and import the Character data correctly as Date Type data.
- **Literal Data:** You saw a Custom Match Expression earlier. Another example: for a given Import or Import/Update, you may want to map the Source field with a specific value, like "**Cherry Tree Trade Show**". It's worth noting that if you have an Update involved, you may want to **Shield** the Mapped field, so that GoldMine's value for existing records will never be changed; but the Expression will still be applied to any new records created by the Setup. Or, you may want to use an Update Expression that protects the Source field if and only if it is non-empty. There are buttons for several such options.
- **Conditional data:** the previous example mentions a conditional Update Expression. But you may want to apply conditional expressions to the basic Import part of the Mapping, also. For example, you may have a field named **Unoemail** that you want to fill with "**Note: Check for E-mail Address**" if the Source file's **Emailaddr** field is empty. So, you'd Map the Expression **IIF(Empty(Source->Emailaddr), "Note: Check for E-mail Address", "")**
- **Translated Data:** For simple translations, you may be able to use a nested conditional IIF Expression. But when there are many possible translation values to be applied, a table-based translator function is the way to go. GoldBox's **SMOOTHCO** table was originally designed for the **SmoothComp** function. But it soon became apparent that GoldBox's original translator function, **GetCustom**, could be replaced with a new, more powerful translation function, **GXTranslate**; and that the **SmoothComp** table was already perfectly configured for it.

Here is a segment of a translator list within the SmoothComp table:



Category	Search for THIS	If Found, return THIS
UDPSC	IN	LFLUHR
UDPSC	KS	JGROTH
UDPSC	KY	BBROSHOU
UDPSC	LA	JGROTH
UDPSC	MA	CPETERSO
UDPSC	MD	CPETERSO
UDPSC	ME	CPETERSO
UDPSC	MI	LFLUHR
UDPSC	MN	LFLUHR
UDPSC	MO	JGROTH
UDPSC	MS	BBROSHOU
UDPSC	MT	LFLUHR
UDPSC	NC	BBROSHOU
UDPSC	ND	LFLUHR
UDPSC	NE	LFLUHR
UDPSC	NH	CPETERSO
UDPSC	NJ	CPETERSO

It assigns records to a specific Counselor (**Key4**) by State. The actual function call used for Mapping is **GXTranslate(GXI->State, "UDPSC")** Why **GXI**? The GXI table is a temporary table that GoldBox creates as part of the first stage of an Import/Update. In that stage, all the initial field Mappings and Expressions are evaluated (that is, the ones controlled by the first 2 buttons shown on the previous page). So, even if you have to convert the "full word" State value to the two character abbreviation (via another GXTranslate) first, you can still count on having the correct State value on hand when you need it for the Counselor translation. (That's because the State field is processed before the Key4 field.)

One of the things that makes the **GXTranslate** function so powerful is that, by adding a 3<sup>rd</sup> parameter, you can have it **examine every word within a field individually**; and apply the translation that exists in a designated list within the **SMOOTHCO** table for each word. This gives you the potential for "data conditioning" your database; that is, applying rules to fields, in particular Address fields, so abbreviations are used consistently, and so forth. You can do this to your entire database as a batch process Global Replace; then keep it that way by using such Expressions when you do your Imports.

**There are other, special purpose kinds of Expressions.** For example, GoldBox's Plug-Ins of Tab records make it possible to add a variety of Tab records based on information in the Source file, or other places. History, Activities, Document Links, Details, Group Membership...and more...can be added this way. As noted earlier, a History Log record that can be added that will track every change made to GoldMine data via an Update (or Import or Import/Update).

A particularly important special expression is the **Write-Back**. It can be used to write the value of any GXI field back to the Source file. In particular, writing the Accountno (and Setup name, when multiple Setups are used) back to each record in the Source file. This gives the most complete audit trail you can get for an Import/Update.

4. **Stacking Setups** has already been introduced. To wrap up this paper, I'll give you an overview of what's typically involved. I'll present more specific information in the forthcoming Details paper.

- We'll start with the **receipt of a Source file** from, say, a vendor. It's common to receive data in an Excel (\*.XLS) file. Because of some nasty things that Excel does to Zip Codes and certain other kinds of fields, I recommend immediately saving the .XLS file as a .CSV file, which is safer. Then, GoldBox can convert that to a dBase file. There's a manual way to do that for one-time Import/Updates. There are also automated ways to do it, for repetitive Import/Updates.

- Once the Source file has been converted to dBase, you will probably want to **add one or two fields** to it, so you can **Write-Back the GoldMine Accountno** (and maybe the Setup name under which a match was made, or a new record was added). Again, there are both manual and automated ways to do this.

- Next, **decide what matching expressions are appropriate for the data**, based on an analysis of the Source file and GoldMine. GoldBox has tools for this. Also, decide **how many Setups will be required** to ensure **SAFE** matching, but also find the maximum possible number of accurate matches.

- **Make the first Setup.** Take all the time required getting the Mapping screen right, because you'll only have to do it once (the other Setups will be copies). None of the Setups except the last will add new records. Be sure to **filter out Source records where ANY of the fields used in Matching are empty**. Also **filter out any Source record where Accountno is not Empty** (we are Writing-Back the Accountno, remember; so any Source record with an Accountno has already been processed). The Accountno part of the filter is irrelevant for the first Setup, of course; but putting it there should ensure that it's in all the others. You will probably also want to set up **LOG Fid Changes**.

- **Copy the first Setup as many times as required.** Change the matching and the filter for each copy.

- **The last Setup will be different.** It will effectively be a straight Import of the remaining, unmatched records. But, in order for Write-Back to work, it will have to be set up as if it were an Update. You'll use the **Accountno Index** for Matching. (You are filtering out records with non-empty Accountnos, and all the Source records that you want to add will fail the Match Expression because of empty Accountnos). Check **ADD No-Matches**, and **BYPASS Matches** (a time saver).

- **Assemble the Q-file.** Even if you are always going to run this manually, you'll want a Q-file to handle certain housekeeping issues. The folder structure shown is from an actual installation. Using it, the Q-file can accomplish the following:



- BACKUP** the GoldMine database. Depending on the kind of database you have, you may use GoldBox to create the backup; or have GoldBox run an executable or Windows Task that will accomplish it.
- Copy a dBase template that's stored in the **Template** folder to the **Working** folder.
- Convert an ASCII CSV file (with the standardized name of TargetX.CSV) that's been copied to the **NewCSVIn** folder, using the template that's been copied to **Working**. This gives us a dBase file containing the Source data in the **Working** folder.
- Add 2 fields to that dBase file: **Accountno** and **Setupno**. This enables us to Write-Back the **Accountno** of found matches (or, if none found, of the new records created). We can also Write-

Back the Setup ID of the Setup that was running when the Write-Back was done. This tells us the Matching scheme that was used to find a matching GoldMine record, if there was one.

- e. Run (in this particular case) 3 Import/Update Setups in series. The first two have actual matching criteria, and so function as Updates. The last one functions as an Import of all the Source records that were not Matched by the other two Setups.
- f. Finally, the Q-file **renames** the dBase file in **Working** to a date-determined name in the **TargetX** root; and renames the CSV file in **NewCSVIn** to a date-determined name in **SourceFileArchive**. This leaves the folders ready for the next run of the procedure.

**The take away:**

- GoldMine's Import utility is good for the simplest of situations. However, most real-world Imports require more, **especially** when Updating existing GoldMine records is involved. Better, more sophisticated Matching is usually required.
- GoldBox's Custom Matching is as good as it gets. The Smooth functions offer fuzzy logic capabilities that go beyond the "black box" approach. You can actually control what GoldBox does, and that can be invaluable.
- Applying multiple Updates by running multiple Setups (only the last of which adds new records) makes the most of GoldBox's superior Matching, providing the ultimate in safe and effective Updates.
- GoldBox's library of 200+ functions and sophisticated Mapping screen enable processing of the incoming data in just about any way imaginable. Many of the functions were created specifically for GoldMine.
- Automation and manual operation are both readily available. Complete file management is included, so once you drop in the Source file and kick off the process, nothing else is required, except for possibly one exception: GoldMine's Lookup.ini. In most cases, actions equivalent to Lookup.ini AutoUpdates can be accomplished by GoldBox, by including the logic right on the Mapping screen. But for extremely complex Lookup.ini files, it may be necessary to run a GoldMine Global Replace of certain "trigger" fields, replacing them with themselves, after the GoldBox procedure has been run. This is so Lookup.ini's functioning will be activated. Naturally, this may present a problem if you want to do a GoldBox automated procedure nightly. If you have a complex Lookup.ini file, contact a GoldBox pro for evaluation of the situation before starting to build a GoldBox procedure.