BoxSoft Super QuickBooks-Export Templates

Every business works differently, and this is reflected in their operational database system (the one that you probably created for them). In sharp contrast, the accounting functions for most businesses are remarkably similar. Rather than re-invent the wheel, it makes sense to provide a custom solution for operations, and a generic solution for accounting.

Many small-to-medium sized businesses choose QuickBooks for their accounting needs, leaving you to build an interface between the Clarion-based operational system and QuickBooks. Due to the idiosyncrasies of QuickBooks, this job can be long, complex and frustrating. Our *QuickBooks-Export* templates will make that task much easier.

The templates lead you by the hand, to specify were your data is coming from and where it needs to go. It automatically creates the views, ranges, filters, access logic, etc., to fetch the information from the operational system, and it guides you through assigning the data to the various files and fields in QuickBooks. It can even automatically apply date/time stamps to your records when they are exported.

In addition to the developer specifying the file and field assignments, it's up to your users to indicate their corresponding QuickBooks accounts (because every business uses different account names and/or numbers). There is a special lookup function included as an importable TXA, which can read the account names and descriptions exported from QuickBooks. This makes it simple for your users to select the account names, so that all the exported transactions get allocated to the proper accounts.

Once you have setup the interface, your program will create an IIF file (Intuit Interchange Format) for each export session. This format is understood by virtually all versions of QuickBooks, so your users should not need to upgrade to the latest version. (Unfortunately, these IIF files must be manually imported into QuickBooks, as there is no way for an external program to instigate this step. However, with proper planning, implementation and training, this hurdle is only a minor annoyance for your users.)

We also provide a variety of examples, showing you how you might setup your own interface. You might want to export every single transaction with all of its details, or you could export only a monthly summary journal entry. You can fully tailor the settings to suit your own needs and expectations.

The templates are a significant aid, but it can still be a little confusing and always requires some trial and error. We're using Intuit's own IIF standard, but QuickBooks can be frustratingly picky about the information it will import. If you need assistance, we can provide additional consulting services. This can be as little as advice before you start, help with debugging your work, or we can do the whole job for you. How much help you get from us is completely up to you.

As with all BoxSoft products, all the source code is included. There are no black boxes or version-specific DLLs to leave you stranded when Clarion is upgraded. You have little need for worry, though; BoxSoft has been creating tools for the Clarion market since 1986, and we have no intention to quit any time soon!

Installation

Install the Files

Once you've finished running the installation program, you should see the following structure under your Clarion directory:

```
C:\CLARION
+-BIN         ST_*.HLP, ST_*.CNT, STAB_CNV.DLL
+-TEMPLATE    STAB_*.TPL, STAB*.TPW              (ABC templates)
|             STCL_*.TPL, STCL*.TPW              (Clarion templates)
+-LIBSRC      STAB*.INC, STAB*.CLW, STAB*.TRN    (ABC templates)
|             STCL*.INC, STCL*.CLW, STCL*.TRN    (Clarion templates)
+-CONVSRC     STAB_CNV.*
`-SUPER
  +-DOC       *.HST (Revision History), ST_*.PDF (Documentation)
  +-IMAGES    *.ICO, *.CUR, *.WMF
  +-SRC_ABC   *.TXD, *.TXA, *.DCT (Source)       (ABC templates)
  +-SRC_CLA   *.TXD, *.TXA, *.DCT (Source)       (Clarion templates)
  +-EX_ABC    *.DCT, *.APP, *.TPS (Examples)     (ABC templates)
  `-EX_CLA    *.DCT, *.APP, *.TPS (Examples)     (Clarion templates)
```

Product Abbreviations for Filenames

AEQB	Super QuickBooks-Export (i.e. Accounting-Export QuickBooks)
BRW	Super Browse
DIA	Super Dialer
FF	Super Field-Filler
IE	Super Import-Export
INV	Super Invoice
LIM	Super Limiter
PCD	Super Passcode
QBE	Super QBE
SEC	Super Security
TAG	Super Tagging
MHSTF	Super Stuff (a.k.a. The "MikeHanson" Templates)

Update the Redirection File

After the files are installed, you must update your Redirection file to update the `*.*` entry. You can do this using the "Setup / Edit Redirection File" pulldown menu option. The entry should look like this:

```
*.*    = .; %ROOT%\examples; %ROOT%\libsrc; %ROOT%\images;
                %ROOT%\template; %ROOT%\convsrc;
                %ROOT%\super\images
```

There `*.RED` examples in the `SUPER\DOC` directory.

Register the Template

Clarion allows you to have multiple template sets accessible in the same application. It does this with the Template Registry. To use a Super Template, you must register it first.

1. Load Clarion, then select the "Setup / Template Registry" pulldown menu option.
2. Press the [Register] button.
3. Select `C:\CLARION\TEMPLATE\STAB_*.TPL` (ABC) or `STCL_*.TPL` (Clarion). The

directory name may not exactly match your system.

Assuming this all went without a hitch, you're ready to start using the template.

RTFM Warning!!!

It is very important that you read this documentation. If you follow the instructions step-by-step, then the usage is very simple. It is almost IMPOSSIBLE if you try to do it on your own!

QuickBooks Export Procedure

The Super QuickBooks-Export templates are actually a combination of templates and classes. The templates should handle most interfacing with the classes, so you should rarely (possibly never) have to call them yourself. In these situations, you can usually copy some generated code to get you started. (For more information on this, see the Class Reference.)

How will Clarion and QuickBooks Match-up?

Before you add the templates, you should think about how well your Clarion database's organization is going to map into QuickBooks. How much information do you want to export? Do you need all of the customers, products, etc. exported into QuickBooks, or is it sufficient to export only summary transactions with accounting totals for each period? Do you need to export all of the line items from each invoice, or can you export only the "totals" for each invoice?

Logging Exports

How are you going to track which information has been exported to IIF files, so that you don't constantly re-export the same information? Will you use a "date-changed" field? Or perhaps you'll create a log of export sessions, with your data records containing a pointer to this record. (That way you would export only records which are new since the last export.) Perhaps you'll use a combination of these. Also, will you let your users re-export old information? Take a look at the example programs for insight into dealing with this problem.

Handling Updates "After the Fact"

Can your records be changed after they have been exported? If so, you will probably have to deal with this. Some file types (e.g. Customers) are automatically updated when you import them again into QuickBooks. Other types (e.g. Transactions) are always added as new records. (If they have the same "DOCNUM", QuickBooks may display a warning message.) It's up to your user to manually delete any duplicate records that are created. In light of this, you may want to modify your program so that transactions cannot be updated once they've been exported.

Exporting "Accounts" from Clarion is Unadvisable

Although you can use the templates to export new Accounts, it's strongly advised that you setup your chart of accounts in QuickBooks to properly match the business needs. Then specify the QuickBooks destination accounts using configuration files in your Clarion system.

This could be as simple as a field in a pre-existing configuration file to hold the QuickBooks account code for sales totals, and another for cash collected to balance the transaction. If you have different sales accounts for each product, then you might want a field for the QuickBooks account code in your Product file. Similarly, if you have multiple payment types, you could have the QuickBooks account number in the Payment Method file.

There is also a procedure that you can import from `SUPER\LIBSRC\AE\QB\SELECT.TXA` that will enable your users to select an account from their QuickBooks chart-of-accounts. The chart of accounts must be exported from QuickBooks into a regular IIF file in an IIF.

As usual, the amount of flexibility (and consummate complexity) is completely up to you. The examples show a variety of ways to approach these issues.

The Templates

Once you've decided how the two systems will coexist, it's time to get started with adding the templates to your application. You start by creating a new procedure using the *"Super QuickBooks Export Procedure"* from the SuperAccExpQB template class. ***Be sure to select***

the template from the "Defaults" tab. This procedure acts as a container for all of the various files to be exported.

Once you've created the above procedure, you can starting adding instances of the "ExportFile" extension template. You'll need to add one of these for each destination file in QuickBooks (e.g. CUST, VEND, TRNS, etc.). You should add them to ensure proper referential integrity; for example, export the customers, vendors, and products before you export the transactions that refer to those entities.

For more information, study the following topics:

QuickBooks Export Procedure
ExportFile Extension
Support Files

QuickBooks Export Procedure Template

(Procedure Template)

This template acts as a container and controller for the various ExportFile Extension templates. You create one of these, and then add one extension for each accounting entity to to be exported from your Clarion database. There is a progress window that will show which file is being exported, etc.

You'll find the settings for this procedure template under the "QuickBooks Export Properties" button on your Procedure Properties window. The settings are organized as follows:

"General" Tab

Export Filename - Since the records are exported into an IIF file, the name for that file needs to be specified somehow. You can have the user do it at run-time (which may not be safe, but is very flexible). You can hard-code it (which means the new file is always overriding the old file). Or you can specify a variable name and control it with your application.

The filename setting assumes you are entering the actual name of the file as a string constant (i.e. it will automatically add the single quotation marks in the generated code). If it's a variable name or expression, then prefix it with an exclamation point.

For example, if you're using the QBExport log file, then you could write an expression like this:

```
!FORMAT(QBExp:SysID, @N08) & '.IIF'
```

"Stamp Values" Tab

There are countless ways that you could ensure that the proper records are exported. The simplest, for example, is to have a BYTE field that equals "0" before it's exported and "1" after. This isn't very flexible, as it difficult to determine when various records may have been exported. However, it's sufficient in some scenarios, and it's very simple to implement.

If you are recording each export session in a log, then you could use the system ID from the log record. This generally works pretty well, except that it doesn't indicate "when it was exported" without fetching the related log record. Also, if you allow your users to update exported records in your Clarion system, and then re-export them, then the system ID in the exported record would represent only the most recent export operation.

Include when stamp equals - The template can automatically build filters to find records matching a standard stamp value (like "0"). If you intend to include only records with a specific stamp value, specify that value here, and the template will handle the filters for all the files. You'll get more control, though, if you specify the filters for each file yourself.

Apply stamp value - After exporting something, the template can automatically apply a stamp value to the stamp field of the exported record. All exported records from all files will get the same stamp value, which helps you to determine what has been exported (and sometimes "when", if you set it up that way).

Force application of stamp, even if blank - If, for some reason, you want your export procedure to be able to clear the stamp values, then you might specify a value of zero for the prior setting (likely through a variable). The templates try to optimize updates to the disk, so they won't normally update the file unless the stamp value is non-zero. If you need it to write-out the blank stamp value, then turn this switch ON.

Force application of stamp, even if equal - The templates also skip the write operation if the stamp already has the specified value. You may be changing other fields in the record, though,

so this switch forces the write to occur, even if the stamp field's value hasn't changed.

ExportFile Extension Template

After you've created the container procedure (see QuickBooks Export Procedure), you'll use this template to specify what accounting entities you want to export. For each destination, you'll add a separate extension. You can think of each of these extensions as a miniature, specialized Process procedure.

The extension will create one export record for each record processed from your file. (The TRNS type is a special case, which is discussed below.)

The settings are as follows:

"Source" Tab

This tab contains options that control where the data is coming from.

Transaction Name - This descriptive text will be displayed in the progress window during the export operation, so that your users knows which file is being processed. Usually it would be something like "Customers" or "Invoices". It might be the same as the actual database file, although this isn't necessary.

Shortcut to Files Window - Once you've populated the extension template, it requires a primary file to be specified so that you can enter the rest of the settings. Rather than exit from the Extensions window to go to the Tables (a.k.a.) schematic window, you can use this shortcut button here. Although you'll likely select the actual file that you're using as a primary file, this setting has no bearing on code generation.

Record Filter - If you specified the "*Include when stamp equals*" setting back in the QuickBooks Export Procedure settings, then you may be able to ignore this. In most cases, though, you'll be specify a filter expression to determine which records will be exported during the session. If desired, this can be combined with a Range, just like in a normal Process procedure. In example #1, the filter statement is:

```
Inv:QBExportDateTime=0 OR Inv:QBExportDateTime >=
     QBExp:BeginDateTime OR Inv:LastUpdateDateTime >
     Inv:QBExportDateTime
```

In this case, `Inv:QBExportDateTime` is the stamp field in the Invoice file, and it holds the actual DateTime that the record was exported. This will be equal `QBExp:ExportDateTime` (a.k.a. Cutoff) in the QBExport log record associated with that export session. The `QBExp:BeginDateTime` contains the cutoff DateTime for the export session. The `Inv:LastUpdateDateTime` is the DateTime that the record was inserted or last updated.

Export DateStamp Field - This is the field in your primary file that will receive the value of the common Stamp specified back in the procedure settings. For more information, see QuickBooks Export Procedure and DateTime Stamps.

If you don't want to save a stamp (if, for example, you're exporting records by date range), then you can leave this blank. If it's blank, the *Update Method* setting below will be disabled.

Update Method - This indicates how you want to update the primary file during the export process. Depending on your file driver, relationship settings, etc., you may prefer one of these methods over the others. Your choices are:

```
Access:File.Update
Relate:File.Update
PUT(View)
```

`PUT(File)`

NOTE: If you're using the "LastUpdateDateTime" stamp fields in your files, then you should use `PUT(File)` or `PUT(View)`, because `Access:File.Update` will cause the stamp to be updated (which is after the beginning of the export), resulting in the record being re-exported with each subsequent export session.

"Range Limits" Tab

These settings are the same as they are for regular Process and Report procedures. Depending on your circumstances, you may find them useful.

"Destination" Tab

It's here that you specify which QuickBooks entity will receive the exported data. All of QuickBooks entities are here. For a description of them all, see QuickBooks Files & Fields. Once you specify the destination file, you need to map some of the destination fields to your source fields. Some of the fields are required, while others are optional (and very frequently to be left empty). Here are a few hints to help you along:

- Create a sample entry in QuickBooks that looks like you expect, then export it from QuickBooks to an IIF to see how QuickBooks allocates data to its columns. Unfortunately, QuickBooks won't export TRNS records (which tend to be the most difficult), but this method works with most other file types.

- Start by mapping a minimum set of fields. Then test it to be sure it works with QuickBooks. Add a couple fields more, then try it again. If you run into problems (which you probably will the first few times), this approach will save you much time with trying to figure out where the problem is occurring.

- Intuit provides a bunch of examples. See Sample IIF Files from Intuit for more details. Most of these will not be an exact match for your situation, but they'll usually give you some insight to carry on.

TRNS/SPL Destination

Most of the QuickBooks file types are simple flat databases (e.g. lists of customers, vendors, etc.). However the TRNS destination (in conjunction with SPL) represents a parent+child, in the standard database fashion. You will have a single parent record, along with *one or more* SPL records. Depending on how much detail you're planning to export, your method will vary quite a bit. Take a look at the example programs for more guidance in this area.

When you choose QuickBooks fields for the TRNS file, you'll discover that their are both *TRNS:Fields* and *SPL:Fields*. If you specify a TRNS field, it will be exported once as part of the header for that transaction; if you specify an SPL field, it will be exported with each detail record.

The templates use a VIEW to access records, which means that the header fields will have their values with each VIEW record. The template automatically figures out (using the header's primary key fields) when the transaction is changing headers. If you're source is not really a header+child, then there's a setting to tweak this behavior. See *"Primary (a.k.a. header) has all details"* below.

One of the TRNS fields is "TRNS:TRNSTYPE". This must always included, to specify which type of transaction you're importing. Depending on this setting, QuickBooks will allow and/or expect certain fields, and disallow others. (That's why it's important to look at Intuit's sample IIF files for guidance.) We often use `'GENERAL JOURNAL'` for regular accounting entries (especially summaries). It tends to be the most flexible and forgiving. Sometimes we'll use `'INVOICE'` or

'PAYMENT', but these require much more care and experimentation. Which type you choose to use really depends on what you want to see after the records have been imported into QuickBooks. (e.g. If you need to print the Invoices after exporting them to QuickBooks, then you need to use the 'INVOICE' TRNSTYPE.)

Primary (a.k.a. header) has all details - This setting should be used if your primary file record contains all the information to export the details. If so, it will not attempt to "break" as the header changes. This can simplify the code required to create the additional SPL records. The template will automatically write the header and first split, and you can output the rest of the SPLs by overriding various methods named ExportExtraSPLs:*position*.

Destination Fields

For each destination field, you can specify the following settings:

QuickBooks Field - This is the destination field in the QuickBooks file. Your list will include only those fields that are acceptable for the destination file that you declared earlier. For TRNS, you'll see TRNS:Fields and SPL:Fields, where TRNS fields are exported with the header and SPL fields are exported with each detail record.

Database Field/Expr. - This is a normal Clarion expression. You can put the name of a field here, a variable, or any expression. This is an EXPRESSION entry prompt, which means that the lookup button takes you to the field list, but resulting the choice is concatenated onto the end of the existing expression, rather than replacing it. If you need to replace the existing expression with your choice, clear the existing expression first. If you specify a lone field name, it will automatically be handled as a "hot field", and will be projected as part of the VIEW (if applicable). For more complex expressions involving field values, you may need to add these manually to the "Hot Fields" list.

Picture Source - Generally you can leave the Picture Source as "Default", and the system will handle it for you. Most QuickBooks fields use @S pictures. Dates use @D01b. If, for some reason, you want a different picture, you can specify "Dictionary" (to grab the default picture from your Clarion dictionary), or "Custom" (so you can type in any picture right here). One example where you would probably do this is a phone number field, so that it appears correctly formatted in QuickBooks.

When the destination field is **"NAME"**, this is usually the unique identifier within QuickBooks. You may have the inclination to specify a literal "name" for this field value. However, it won't work (unless your names are really unique). It's usually best to use a numeric field (like your database's auto-number field).

Also, QuickBooks shares some pools of unique names. For example, Customers and Vendors must have unique "names" across both record sets. Depending on your users' desires, you may want to preface the customer and/or vendor numbers with something (like "C" for Customer and "V" for Vendor, or "Cust" and "Vend", etc.). Hence, the "name" for customer #1234 could be "C1234" or "Cust 1234", while the "name" for vendor #1234 could be "V1234" or "Vend 1234".

"Hot Fields" Tab

These are the same "Hot Fields" that you would specify for regular Process and Report procedures. If you specify a source field directly when you are matching up fields to the destination, then it will automatically be a hot field. If, however, you have a complex expression, then you'll need to specify that each of the fields in the formula are hot field, so that they are projected in the VIEW structure.

We've often wasted time tracking down a problem, only to discover it was caused by a missing hot field. Therefore, the default is to project *all* of the fields in the VIEW. Once you get your

export working properly, you may want to turn this off and then manually add all of the necessary hot fields. It won't make a huge difference in performance, but it will help a little (for all file drivers that we've tested, including both TPS and SQL).

"Classes" Tab

This is the same as the classes tab on any other ABC template. If you intend to override some of the virtual methods with your own code, it's strongly recommended that you change this from the default to something more recognizable (like "QB_Inv" for the QuickBooks Invoice export class).

If your destination is TRNS, then there will be an extra button entitled "SPL Class Settings", where you can specify the values for the SPL object. If our main object is called QB_Inv, then our SPL object would usually be QB_InvSPL.

Exporting extra SPLs:

In some situations (like when the header has all the details), you will want to export extra SPL records. There are three things you must do:

1. For any SPL fields whose values can come from more than one source (i.e. two different amount fields), you need to use an intermediate variable as the expression for the `SPL:Field`.

2. Those intermediate variables must have their values set in the `CalcFields` method for the SPL class (before the parent call). There is a `CalcBank` parameter for this method. You must create a `CASE` structure to check the `CalcBank` value, and then handle the calculations for each type of SPL. The automatic SPL(s) will call `CalcFields` with `CalcBank=0`. For all other SPLs, you can use any other numbers that make sense to you. You must create calculations for all of the SPLs, including the automatic one(s).

3. As was just mentioned, at least one SPL is written automatically. It's up to you to instigate the writing of the rest. There are three "usual" methods where you would do this:

   ```
   SPLObject.ExportExtraSPLs:BeforeFirst
   SPLObject.ExportExtraSPLs:AfterFirst
   SPLObject.ExportExtraSPLs:AfterLast
   ```

   where "`SPLObject`" is the object name in the "SPL Class Settings" (on the extension's "Classes" Tab). Choose whichever method puts the extra SPL items where you want to see them in the transaction. Use the command:

   ```
   SELF.ExportRecord(CalcBank)
   ```

   where *`CalcBank`* is the bank number to be interpreted by your code in the `CalcFields` method. Remember that at least one SPL is exported automatically, and you don't have to issue the ExportRecord command for that.

Hints for TRNS/SPL:

- The total for all SPL:AMOUNT records within a TRNS must total to the negative value of the TRNS:AMOUNT. It's up to you to ensure that this balance occurs. If your amounts in both the header and child in your database are positive, then you will likely need to negate the expression for the SPL:AMOUNT. If you negate the SPL:AMOUNT, you must also negate the SPL:QNTY (if you're exporting that field). SPL:PRICE (a.k.a. unit price) needs to be positive, regardless of whether the SPL:AMOUNT is negative. (This is demonstrated in the example programs.)

- The TRNS:NAME field is actually the customer ID.

- The TRNS:ADDR*n* fields are purely for display purposes, and don't have to correspond to the same-named fields in the CUST file. You would usually put the customer's textual name in TRNS:ADDR1, the customer's address in TRNS:ADDR2, etc.

- The TRNS:MEMO field is usually displayed in the list of transactions in QuickBooks, so it's good to put some identifiable text here, like the customer's name.

- There are sometimes corresponding fields (like TRNSTYPE and DATE) that will have the same value in both TRNS and SPL.

- Usually the TRNS:TRNSID and SPL:SPLID fields need to be included in the export file as blanks (''). It's not sufficient to omit them.

- If you are not exporting the detail records for your transaction (i.e. the header *is* the transaction), then the templates will generate only *one* SPL record for each TRNS record. You may need to create some additional SPL records yourself. Remember that the transaction **must** balance. Take a look at the examples (specifically the Invoice and Payment portions of example #1, and everything in examples #2 and #3).

- Most of the TRNSTYPE values imply that it's a positive-amount (e.g. INVOICE of $25, PAYMENT of $10, etc.), even though these may be debits or credits. To balance this, the SPL amounts are negative. In the case of a GENERAL JOURNAL transaction, however, the leading amount in the TRNS header *may* need to be negative to create the desired transaction in QuickBooks. The best approach is to try it one way, and if the imported transaction is incorrect, then change it to be the other way.

- When exporting PAYMENT records, the header (i.e. TRNS) must specify a "Bank" or "Other Current Asset" account. For example, you cannot do a deposit directly to "Accounts Receivable". The total applied to AR must appear in a SPL line.

- Some TRNSTYPEs (e.g. `PAYMENT`) require that there is only *one* SPL line accompanying the TRNS header. If your transaction requires a more complex split, then you cannot use that TRNSTYPE, and must resort to `'GENERAL JOURNAL'` instead.

- You cannot export a blank date. If the column is included in the export file, then a date *must* be present.

Examples:

#1 - Customers, Invoices & Payments
#2 - Transactions without "Details"
#3 - Period Summaries

Support Files

You'll find all the TXDs and TXAs in `SUPER\LIBSRC\AE\QB`.

SELECT.TXA

Your users will need to specify the various QuickBooks destination accounts in your Clarion system. Although you could rely on them to type them in properly, there is no way for the system to verify that they are correct. To vastly reduce this possibility, we've provided a sample procedure, **SelectQuickBooksAccount**, to provide a lookup browse for your users. It needs to import the account names from a QuickBooks IIF file, so your user needs to export the chart of accounts (COA) from your QuickBooks company before this procedure will work.

As it's written, SelectQuickBooksAccount loads the name of the IIF file in STABAEQB.INI, in the current directory. The section is called `IIF` and the line is `ChartOfAccounts`. The first time it will prompt for the filename, and once it's been specified by the user, it remembers it for the future. There's a button for the user to re-specify the file as well. If you want to change this to a hard-coded filename, or have it automatically scan the directory for IIFs, or whatever, then all the source is available to you. Due to the unpredictable nature of users, though, I would suggest that you leave it as-is until you decide if there's a better way to handle your users' needs.

The procedure returns either an account name from the IIF file, or an empty string (if the user pressed Cancel).

> ***NOTE:*** If you haven't already included it, you must add the ASCII file driver to your project settings.

EXPLOG.TXD and EXPLOG.TXA

There are many methods that you can choose for controlling when records should be exported to QuickBooks. One method is to keep a log of when export sessions were run, the resulting IIF filename, and their "cutoff date". All records changed after the cutoff will be included in the export. You must maintain a couple of fields in your exportable data files (Customer, Invoice Header, etc.), so that it can be determined if an export is applicable.

EXPLOG.TXD contains a file definition for QBExport. It's a Topspeed file, but you can change the driver and/or path as soon as you import it into your own dictionary. Don't change the file or field labels, though, or the application TXA import won't work.

EXPLOG.TXA includes the procedure **BrowseQBExport**. This displays records from the QBExport log file, and also contains two large buttons for calling your **QBExport** procedure (the one that performs the actual export). The two buttons are:

Export New Only - This will automatically export only those items that have been added or changed since your last export. For this button, it doesn't matter which record is highlighted in the browse.

Export New, plus Re-export any on or after the highlighted "Cutoff Date/Time" - This exports all of the new records (as above). It also exports any records that were exported after the highlighted export session's cutoff date. This is useful if they've forgotten to import their records into QuickBooks, or they've accidentally deleted their IIF file, or any of a million other reasons.

You might be tempted to ask why we can't reproduce a specific IIF file, exactly as it was exported earlier. The main reason is that your users may have changed the records (like Customer), and they will not look exactly the same. The safest approach in this situation is to include anything that's new since the highlighted

export's original cutoff date. If you decide that you prefer another approach, then you're welcome to change this procedure to suit your needs, or to roll-your own from scratch.

EXPLOG.TXA also contains two procedures to handle the packing and unpacking of DateTime Stamp Fields. As they're written, they use 12-digit DateTime values. If you want to use 10 digit versions instead, then edit the source and un/comment the appropriate lines. The two procedures have the following prototypes:

DateTimeStampBuild(LONG p_Date,LONG p_Time),REAL - This function takes Date and Time parameters, and returns a packed DateTime value as a REAL.

DateTimeStampExtract(REAL p_DateTimeStamp,<*LONG p_Date>,<*LONG p_Time>) - This procedure separates the passed DateTime into its constituent Date and Time values. You can pass in either (or both) of the Date and Time LONG parameters. Because these are passed by reference, you must use a LONG versus any other data type.

In addition to importing these files into your dictionary and application, you must also ensure that the data files to be exported have two DateTime stamp fields. The first will be called **`LastUpdateDateTime`**. It must be assigned whenever records are added or changed. The easiest approach is to override the `Insert`, `Update` , `TryInsert`, and `TryUpdate` methods for the files' classes in the global embeds.

The other DateTime field will be called **`QBExportDateTime`**. It holds the actual time that the record was last exported, and it's updated by the *QBExport* procedure (the one you create using the templates).

When you create the **`QBExport`** procedure itself, `BrowseQBExport` assumes that it will be called `QBExport`. If you don't like that procedure name, then you'll have to change the calls in `BrowseQBEExport`.

For each primary file that's processed with the `QBExport` procedure, you must specify the filter statement in each extension. The filter expression will look something like this:

```
Inv:QBExportDateTime=0 OR Inv:QBExportDateTime >=
     QBExp:BeginDateTime OR Inv:LastUpdateDateTime >
     Inv:QBExportDateTime
```

In this case, the prefix of the exported file is `Inv`. The `QBExp:BeginDateTime` field is the cutoff date in the `QBExport` log file. It's calculated by the embed logic in the `BrowseQBExport` procedure that you just imported from the TXA.

If you prefer a different approach to insert/change auditing, then feel free to adapt the above approach to your own circumstances. As we already mentioned, this is only one of countless possible methods, and this one is provided as a model to get you up to speed as quickly as possible.

DateTime Stamp Fields

DateTime Stamp fields should be able to hold the 5 digits returned by Clarion TODAY() function, plus the 7 digits returned by the CLOCK() function (i.e. 12 digits total). Depending on your circumstances, you could use a REAL or DECIMAL(12). The values can be calculated as simply as:

```
Inv:LastUpdateDateTime = TODAY() * 1.0e+7 + CLOCK()
```

If you will usually be doing your exports to QuickBooks during a period of relative inactivity (e.g. after the store has closed), then you may not need a resolution of 0.01 seconds (as above). If you can live with a 10 second resolution, then you could use a LONG instead of a REAL/DECIMAL(12), in which case your code would look like this:

```
!Alternate 10 seconds resolution
Inv:LastUpdateDateTime = TODAY() * 1e+4 + CLOCK() / 1e+3
```

If you import the EXPLOG TXD and TXA from the Support Files into your dictionary and application, there are two procedures for packing and unpacking these DateTime Stamp values. As written they use the high-resolution values, but the alternate code is there to be uncommented.

Sample IIF Files from Intuit

Intuit provides a variety of sample IIF files for your perusal. These are invaluable for determining what and how you need to assign your various Clarion database columns to the QuickBooks equivalents. You'll find the following files in the `SUPER\DOC\QUICKBOOKS` directory:

`IIF.EXE`	Sales tax preference **ON**.
`IIF2.EXE`	Sales tax preference **OFF**.

These are self extracting ZIP files. You can also find sample files on Intuit's web site:

www.quickbooks.com/support/faqs/docs/w_iiffiles.html
www.quickbooks.com/support/faqs/docs/w_iiffiles2.html

Unfortunately, this often will not apply to your QuickBooks company, because of the myriad ways that QuickBooks can handle accounting operations. Regardless, they are a helpful learning tool, and will often come in handy when you're tackling a new scenario.

Example #1: Customers, Invoices & Payments

You'll find this example in the directory `SUPER\EXAMPLES\AE\QB1`. It represents a "normal" export of Customers, Invoices, and Payments. Here are some things you should note:

- This implementation might have a single export done at the end of each day, perhaps before the deposit is made up for the bank.

- There is a sample company QuickBooks company in **`TEST.QBW`**. It has a number of accounts for receiving information from the Clarion database, including "Accounts Receivable", "Sales Inc", "Cash Collected", etc. Note that the account "CC Collected" will be used for the collection of all credit card payments. This isn't necessarily the same way that you should do it in your application, but it demonstrates how multiple sources in your Clarion database can be allocated to a single QuickBooks account.

- The mapping for the accounts must be configurable by your users, so there's a configuration file in the Clarion dictionary to hold some of these settings that are fixed, and there's also a setting in Payment Method file. The example assumes that you are allocating all of your sales to a single sales income account. If, instead, you wanted to map the income from each product to a separate account in QuickBooks, then you would also need a setting field for the account in the Clarion Product file.

- The chart of accounts has been exported from QuickBooks to the accompanying **`COA.IIF`**, for use by thThe Header IS the Transactione **`SelectQuickBooksAccount`** procedure in Clarion. (Recall that this lookup procedure is imported from the Support Files.)

- It's assumed that your inventory management occurs on the Clarion side, which is quite common with a split between a custom operational system and an off-the-shelf accounting package).

- There is an export log file called **`QBExport`**, which acts as a record of when each export was performed. This file was imported from **`EXPLOG.TXD`**. (See Support Files for more information.)

- There are DateTime fields in the `Customer` and `Invoice` files. Even though the `InvItem` file contains the details for the `Invoice`, the DateTime fields are not required in that child file. This is because all exporting is controlled by the parent (i.e. header) file.

- The Customer export is a very simple one, and you should be sure to fully understand it before moving on to the Invoice and Payment exports.

- The `Invoice` records on the Clarion side contain both sales and payment information. To illustrate proper handling of Accounts Receivable, we do two passes of the Invoice file. First we create the `INVOICE` transactions, followed by the `PAYMENT` (actually `'GENERAL JOURNAL'`) transactions. It must be done as two transactions, because the format of the two transaction types can be rather different. Also, if you're really using supporting Accounts Receivable, the payment may come much later than the original Invoice. To facilitate this, there are also two QBExportDateTime fields: one called `QBExpInvDateTime` and the other called `QBExpPmtDateTime`

  The Invoices and Payments were exported as separate transactions, mostly for demonstration purposes. It also showed how you can post information against Account Receivable, while having it apply to a particular customer. In a "real" application, we would probably have exported the entire invoice as a single General Journal transaction, and eliminated the concept of Accounts Receivable in this case (because the invoices in this example are always paid-in-full at the time of the sale). In fact, that's exactly what we do in Example #2.

- To balance the transaction with the extra Invoice header fields (tax amounts for Invoices, and both taxes and payment amounts for the Payments), some extra SPL records will be created using hand-code. Therefore, the SPL destination fields are setup to receive their values from intermediate local variables. These local variables must be calculated with the appropriate values at various points in time, which is done by overriding `QB_InvSPL.CalcFields` (for Invoices) and `QB_PmtSPL.CalcFields` (for Payments). For actual children for the Invoice (and for the one automatic SPL for the Payment), this method is automatically called with `CalcBank=0`. Non-zero values are used to do the extra calculations. The equates for these are hand-coded in an embed at the beginning of the procedure.

  To write the SPL items, a call to *`SPLObject`*`.ExportRecord(`*`CalcBank`*`)` is used (where *`SPLObject`* is either `QB_InvSPL` or `QB_PmtSPL`). Behind the scenes, this method calls the `CalcFields` method with the appropriate *`CalcBank`* parameter, then calls the `WriteRecord` method to perform the actual disk output operation. Note that these calls are in `ExportExtraSPLs:`*`AfterLast`* for the Invoices, and `ExportExtraSPLs:`*`AfterFirst`* for the Payments.

- In the case of the exported Payments, the InvItem file is not included in the schematic. Therefore there are no "children". Instead the setting for *"Primary (a.k.a. header) has all details"* is turned ON. The first payment is written out automatically as a child, and its local values must be calculated in the `QB_Pmt.CalcFields` method. The other, optional payment methods (including change due) are calculated and exported in the `ExportExtraSPLs:`*`AfterFirst`* method.

- QuickBooks will only accept PAYMENT transactions that involve a Bank or Current Asset account (in the TRNS record) and the Account Receivable account (in the single SPL record). Because our example supports multiple payment types for each invoice, along with the option of cash back as change, we can't use the PAYMENT transaction type. Instead, we'll create a GENERAL JOURNAL transaction, which allow more flexibility. Note that the amounts must be negated to properly balance the customer's account in QuickBooks.

Example #2: Transactions without "Details"

You'll find this example in the directory `SUPER\EXAMPLES\AE\QB2`. The Payments processing in Example #1 also demonstrates this technique, so you can look at that for another example.

In most situations you probably don't need to export all of the invoice details (products, quantities, etc.) from your Clarion database into QuickBooks. All it may need it the total breakdown of the sale (i.e. total transaction amount, sub total w/o tax, and taxes collected).

This information might already exist as distinct fields in your transaction's header record, or it could be calculated at run-time by scanning the detail records to get the totals. Regardless of which approach you've used in your Clarion application, you must still export a properly-balanced "split" transaction to QuickBooks.

As with Example #1, you'll find a QuickBooks company in TEST.QBW. In this case we won't be using Accounts Receivable or exporting Customers. We're merely concerned with getting the basic transactions from Clarion into QuickBooks. Most operational systems handle the majority of customer management, sales, payments, etc., and it's often unnecessary to send keep a duplicate list of customers in the QuickBooks system.

We're using the GENERAL JOURNAL transaction type, because it allows the most flexibility with splits, account allocation, etc.

Getting the +/- sign on the amounts can be tricky. Note that in the Destination fields, the TRNS:AMOUNT is negated, whereas the SPL:AMOUNT is left alone. (The two amounts must always be opposite, so that the transaction balances.) In this case the negative is there for the TRNS:AMOUNT, because the destination account ("Sales Income") is a credit account, where negatives are actually positives. When we were building this example, our first attempt produced the wrong result, and we had to switch them to get it right.

There's an additional hitch: Subtotal, Tax1 and Tax2 are "credit" fields, while Pmt1, Pmt2, Pmt3, and Change are "debit" fields. Subtotal appears in the TRNS:AMOUNT header, while Tax1 and Tax2 appear as SPLs (where SPL:AMOUNT is usually negated). Because the TRNS' companion fields are SPLs, they must be double-negated to get it to balance. This can make it look a little funny in QuickBooks, but it is definitely correct. Again, a little experimentation will see you through.

Example #3: Period Summaries

You'll find this example in the directory `SUPER\EXAMPLES\AE\QB3`.

In many situations you won't want every transaction exported individually to QuickBooks. Instead, you want a single transaction representing a period of business activity. It should be a balanced General Journal transaction, with all the necessary accumulated accounts represented. Not surprisingly, the resulting journal entry looks almost the same as the single transaction per invoice demonstrated in example #2, except that the transaction includes the total of multiple invoices, payments, etc.

To achieve this, we use the `QBConfig` file as the primary file for the export. There's only one record in this file, so we'll get only one transaction output in our IIF file. We don't specify the Stamp Field, so that the QBConfig file isn't changed during the export.

To get the desired totals, we use a *Process* procedure called `SumInvoicesForQBExport` to accumulate the totals. It will have the same filters that we used in example #2, and we'll update the stamp field manually. (Rather than using the stamp field, you could use date ranges instead. Use whatever approach makes the most sense for your application.)

During calculations, the total values are stored in global variables, and total payments are stored in a global queue (with one record per payment type).

As usual, the template creates the TRNS and first SPL record, and the rest of the SPL lines are up to you. Note that the `Glo:TotalPmtQueue` is sorted by payment type ID during the accumulation, but is sorted by DisplayOrder after it's done, so that the payments appear in the desired display order within the QuickBooks transaction.

The TRNS fields can be assigned from the explicit fields (`QBCfg:QBSalesAccount` and `Glo:TotalSales`), but the SPL lines need to use intermediate local variables so that you can feed it the various values accumulated in the other global variables and PmtQueue.

Different from the other examples, the `CALCBANK` equates now have a `Pmt0` (i.e. payment-zero) equate. This is used as an offset to determine the actual record to be fetched from the `Glo:TotalPmtQueue`.

Regarding the appearance of the journal transaction in QuickBooks:

- The "DOCNUM" for the transaction is the `QBExport SysID` value, and the memo contains *"Imported 00000000.IIF"*, where "00000000" is the `SysID`.
- The date for the transaction is the date that the export was performed. This date comes from a local variable (`Loc:ExportDate`) that contains the extracted date from the `QBExp:ExportDateTime` stamp field.
- There is no specific customer, so the NAME and ADDR*n* lines are not exported.

QuickBooks Files & Fields

The following list of QuickBooks files and fields was transcribed directly from the QuickBooks help file, and is copyrighted by Intuit Software. You can find it yourself under the topic of "IIF file format". You may want to check your own version of QuickBooks, especially if you're planning to export employee payroll information (which is very region specific).

If you discover that your version of QuickBooks contains fields that are not supported by the Super QuickBooks Export templates, then please let me know so that I can update the templates.

ACCNT (Chart of Accounts)
BUD (budgets)
CLASS (Class list)
CTYPE (Customer Type list)
CUST (Customer:Job list)
EMP (Employee list)
HDR (header information)
INVITEM (Item list)
INVMEMO (Customer Message list)
OTHERNAME (Other Names list)
PAYITEM (Payroll Item list)
PAYMETH (Payment Method list)
QBP EMPLOYEE (Employee payroll records)
SHIPMETH (Ship Via list)
SPL (Distribution lines)
TERMS (Terms list)
TIMEACT (activites timed with the Timer)
TIMERHDR (Header for Timer data)
TODO (To Do list)
TRNS (Transactions)
VEND (Vendor list)
VTYPE (Vendor Type list)

ACCNT (Chart of Accounts)

Import and export files can contain complete information about your Chart of Accounts. The keyword for the Chart of Accounts is ACCNT.

Column headings

The following table shows the column headings for a Chart of Accounts. The NAME and ACCNTTYPE columns are required.

NAME	(Required) The name of an account in your Chart of Accounts. If the account is a subaccount, the account's name includes the names of the parent accounts, beginning with the highest level account. If you are creating an import file, use a colon (:) to separate subaccount names.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Chart of Accounts.
REFNUM	(Export files only) A unique number that identifies the account in the list.
ACCNTTYPE	(Required) The type of account. If you are creating an import file, use one of the keywords below to indicate the account type:

AP	Accounts payable
AR	Accounts receivable
BANK	Chequing or savings
CCARD	Credit card account
COGS	Cost of goods sold
EQUITY	Capital/Equity
EXEXP	Other expense
EXINC	Other income
EXP	Expense
FIXASSET	Fixed asset
INC	Income
LTLIAB	Long term liability
NONPOSTING	Non-posting account
OASSET	Other asset
OCASSET	Other current asset
OCLIAB	Other current liability

OBAMOUNT	The opening balance of the account. Leave the amount blank and import the account without an opening balance. If the account must have an opening balance, use the TRNS keyword to create an opening balance transaction for the account.
DESC	A brief description of the account.
SCD	The tax line assigned to the account. To specify the tax line, enter its number (tax line numbers come from the file BUSTAX.SCD). If you don't know the correct number, leave this field blank. You can always assign the tax line later, after you import the Chart of Accounts into QuickBooks.
ACCNUM	The account number of the account.
EXTRA	Identifies an account as one of the special balance sheet accounts that QuickBooks automatically creates when the need for the account arises. If you are creating an import file, use one of these keywords to identify the account:

OPENBAL	Opening Balance Equity
RETEARNINGS	Retained Earnings
GST	GST Payable
PST	PST Payable
UNDEPOSIT	Undeposited Funds
INVENTORYASSET	Inventory Asset

	UNCATINC	Uncategorized Income
	COGS	Cost of Goods Sold
	UNCATEXP	Uncategorized Expense
PURCHORDER	Purchase Order	

BUD (budgets)

Import and export files can contain complete information about your budgets. The keyword for a budget is BUD.

Column headings

The following table shows the column headings for a budget. Only the ACCNT and AMOUNT columns are required.

ACCNT	(Required) The name of the account to which this budget applies. If the account is a subaccount, the account's name includes the names of the parent accounts, beginning with the highest level account. If you are creating an import file, use a colon (:) to separate subaccount names.
PERIOD	A keyword that identifies the time intervals within the budget. Because QuickBooks only allows one month intervals in a budget, an export file always shows the keyword MONTH. If you are creating an import file, type MONTH.
AMOUNT	(Required) The budget amounts. Because a QuickBooks budget covers an entire year, an export file shows 12 amounts, one for each month of the year. If you are creating an import file, each month's amount must have its own column.
STARTDATE	The starting date of the budget. If you are creating an import file, enter the date in MM/DD/YY format. For example, 1/30/94.
CLASS	The name of the class that the budget applies to. If the class is a subclass, the class name includes the names of the parent classes, beginning with the highest-level class. If you are creating an import file, use a colon (:) to separate subclass names.
CUSTOMER	The customer to which the budget applies. If you are applying the budget to a specific job for the customer, the customer name includes the job name. The customer's name comes first, then a colon (:), then the job name.

CLASS (Class list)

Import and export files can contain the list of classes you use in QuickBooks. The keyword for the Class list is CLASS.

Column headings

The following table shows the column headings for the Class list. Only the NAME column is required.

NAME	(Required) The name of the class. If the class is a subclass of another class, the name includes the name of the parent classes, beginning with the highest level class. If you are creating an import file, use a colon (:) to separate subclass names.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Class list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.

CTYPE (Customer Type list)

Import and export files can contain the list of customer types you use in QuickBooks. The keyword for the Customer Type list is CTYPE.

Column headings

The following table shows the column headings for the Customer Type list. Only the NAME column is required.

NAME	(Required) The customer type. If the customer type is a subtype of another customer type, the name includes the name of the parent types, beginning with the highest-level type. If you are creating an import file, use a colon (:) to separate subtype names.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Customer Type list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.

CUST (Customer:Job list)

Import and export files can contain all the information on your Customer:Job list. The keyword for the Customer:Job list is CUST.

Column headings

The following table shows the column headings for the Customer:Job list. Only the NAME column is required.

NAME	(Required) The name of the customer. If a job name is included, the customer name appears first. If you are creating an import file, use a colon (:) to separate the customer name from the job name.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Customer:Job list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
BADDR1	The first line of the customer's billing address.
BADDR2	The second line of the customer's billing address.
BADDR3	The third line of the customer's billing address.
BADDR4	The fourth line of the customer's billing address.
BADDR5	The fifth line of the customer's billing address.
SADDR1	The first line of the customer's shipping address.
SADDR2	The second line of the customer's shipping address.
SADDR3	The third line of the customer's shipping address.
SADDR4	The fourth line of the customer's shipping address.
SADDR5	The fifth line of the customer's shipping address.
PHONE1	The customer's phone number.
PHONE2	The customer's alternate phone number.
FAXNUM	The customer's FAX number.
NOTE	The name or number of the account you want to associate with this customer.
CONT1	The name of your primary contact with the customer.
CONT2	The name of an alternate contact with the customer.
CTYPE	Your classification for the customer (QuickBooks calls this a "customer type"). If you import a customer type that is not on your Customer Type list, QuickBooks adds the new customer type to the list.
TERMS	The customer's payment terms with your company.
TAXABLE	Indicates whether you assigned a customer tax code. If you are creating an import file, enter one of these keywords in the TAXABLE field: **Y** Yes. You have assigned a customer tax code. **N** No. You have not assigned a customer tax code.
TAXCODE	The tax code assigned to this customer.
LIMIT	The customer's credit limit with your company. If you are creating an import file, enter the dollar amount.
RESALENUM	The customer's resale number.
REP	The initials of the sales representative who deals with the customer. **Name:** (Required) First and last name of the sales representative. **ListID:** (Required) Number of the list to which the sales representative should be added. **2** Vendor **3** Employee **4** Other Names

	Initials: (Required) Initials of the sales representative.
NOTEPAD	Your notes about the customer. If you are creating an import file, the notes appear in the Notepad window for the customer.
SALUTATION	The customer's salutation, or title (Mr., Ms., Doctor, etc.).
COMPANYNAME	The name of the customer's company.
FIRSTNAME	The customer's first name.
MIDINIT	The customer's middle initial.
LASTNAME	The customer's last name.
CUSTFLD1 **CUSTFLD2** **...** **CUSTFLD 7**	The custom field entries for the customer (you can have up to 7 custom field entries). Custom fields let you track special information about the customer, such as the customer's birthday or email address. What you use custom fields for is entirely up to you.
JOBDESC	(QuickBooks Pro only) A short description of the job you a performing for the customer.
JOBTYPE	(QuickBooks Pro only) A word or phrase you want to use to categorize the job you are performing for the customer. You can create reports based on job types.
JOBSTATUS	(QuickBooks Pro only) The status of the job you are performing for the customer. QuickBooks Pro uses these terms to indicate job status: **Pending** **Awarded** **In progress** **Closed** **Not awarded**
JOBSTART	(QuickBooks Pro only) The starting date of the job you are performing for the customer.
JOBPROJEND	(QuickBooks Pro only) The projected ending date of the job you are performing for the customer.
JOBEND	(QuickBooks Pro only) The ending date of the job you performed for the customer.
NOTE	The name or number of the account you want to associate with this customer. QuickBooks requires an account number if you want to set up this customer as an online payee. The payee uses this number to identify you.

EMP (Employee list)

Import and export files can contain all the information in your Employee list. The keyword for the Employee list is EMP.

Column headings

The following table shows the column headings for the Employee list. The NAME and INIT columns are required.

NAME	(Required) The name of the employee.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Employee list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
INIT	(Required) The employee's initials.
ADDR1	The first line of the employee's address.
ADDR2	The second line of the employee's address.
ADDR3	The third line of the employee's address.
ADDR4	The fourth line of the employee's address.
ADDR5	The fifth line of the employee's address.
SIN	The employee's Social Insurance number.
EMPN	The employee's Employee Number
PHONE1	The employee's phone number.
PHONE2	An alternate phone number for the employee.
NOTE	The name or number of the account you want to associate with this employee.
NOTEPAD	Your notes about the employee. If you are creating an import file, the notes appear in the Notepad window for the employee.
FIRSTNAME	The employee's first name.
MIDINIT	The employee's middle initial.
LASTNAME	The employee's last name.
SALUTATION	The employee's salutation, or title (Mr., Ms., Doctor, etc.).
CUSTFLD1 **CUSTFLD2** . .**CUSTFLD7**	The custom field entries for the employee (you can have up to 7 custom field entries). Custom fields let you track special information about the employee, such as the employee's birthday or email address. What you use custom fields for is entirely up to you.
HIDDEN	This list entry is currently hidden.

HDR (header information)

An export file always begins with a header that shows the version and release numbers of the QuickBooks software used to create the file. The keyword for the header is HDR.
If you are creating a file to import, you do not need to include a header.

Column headings

The following table shows the column headings that appear in the header:

PROD	Identifies the Intuit product that created the export file.
VER	The version number of the QuickBooks software that created the export file.
REL	The release number of the QuickBooks software that created the export file.
IIFVER	The version number of the export file format (IIF).
DATE	The date the export file was created.
TIME	The time the export file was created.
ACCNTNT	Is there an Accountant's Review Copy in effect at this time?
ACCNTNTSPLITTIME	

INVITEM (Item list)

Import and export files can contain all the information on your Item list. The keyword for an entry in the list is INVITEM.

Column headings

The following table shows the column headings for the Item list. The NAME, INVITEMTYPE, and ACCNT columns are required.

NAME	(Required) The name of the invoice item.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Item list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
INVITEMTYPE	(Required) Indicates the type of invoice item. If you are creating an import file, use one of these keywords to indicate the item type. **DISC** Discount item **GRP** Group item (groups several invoice items into a single item) **INVENTORY** Inventory part item **OTHC** Other charge item **PART** Non-inventory part item **PMT** Payment item **SERV** Service item **SUBT** Subtotal item
DESC	A description of the item as you want it to appear in the Description column on invoices, credit memos, and sales receipts.
PURCHASEDESC	(Inventory part items only) A description of the item as you want it to appear on purchase orders.
ACCNT	(Required) The name of the income account you use to track sales of the item. The type of this account should be INC.
ASSETACCNT	(Inventory part items only) The name of the asset account you use to track the value of your inventory. The type of this account should be OASSET.
COGSACCNT	(Inventory part items only) The name of the account you use to track the cost of your sales. The type of this account should be COGS.
QNTY (K)	Quantity of inventory on hand
QNTY (L)	Value of inventory on hand
PRICE	(All item types except group, payment, and subtotal) The rate or price you charge for the item. If you are creating an import file, add a percent sign (%) if the amount is a percentage.
COST	(Inventory part items only) The unit cost of the item.
STAXCODE	Indicates this item's tax code when you sell it
PTAXCODE	Indicates this item's tax code when you purchase it.
PAYMETH	(Payment items only) The payment method customers use (cheque, Visa, etc.).
PREFVEND	(Inventory part items only) The name of the vendor from whom you normally purchase the item.
REORDERPOINT	(Inventory part items only) The minimum quantity you want to keep in stock at any given time. When your inventory reaches this level, QuickBooks informs you that it is time to reorder the item.
EXTRA	Adds additional information about the invoice item. These keywords can appear in the EXTRA field: **REXPGROUP** Indicates that the item is a group of reimbursable expenses that you included on

		the invoice.
	REXPSUBTOT	Indicates that the item is the subtotal amount for a group of reimbursable expenses you included on the invoice.
CUSTFLD1 **CUSTFLD2** **...** **CUSTFLD5**	The custom field entries for the item (you can have up to 5 custom field entries). Custom fields let you track special information about the item, such as colour, unit or measure, or size. What you use custom fields for is entirely up to you.	
DEP_TYPE	(Payment items only) Indicates how you want QuickBooks to handle deposits of the payment item.	
	1	You want QuickBooks to deposit the payment in the bank account of your choice when you record the payment. The payment does not go into the Undeposited Funds account, and you do not have to use the Make Deposits window to deposit the payment.
	0	You want QuickBooks to "hold" all the payments in a special account named Undeposited Funds. To move the payments to a bank account, you must use the Make Deposits window to group the payments into one deposit.
ISPASSEDTHRU	(QuickBooks Pro) (Service, non-inventory part, and other charge items) Indicates whether you pass the item through as an expense to the customer.	
	Y	Yes. You pass the item through as an expense.
	N	No. You do not pass the item through as an expense.

INVMEMO (Customer Message list)

Import and export files can contain the list of customer messages you use in QuickBooks. The keyword for the Customer Message list is INVMEMO.

Column headings

The following table shows the column headings for the Customer Message list. Only the NAME column is required.

NAME	(Required) The text of the customer message.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Customer Message list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
HIDDEN	This list entry is hidden. Y/N

OTHERNAME (Other Names list)

Import and export files can contain all the information in your Other Names list. The keyword for the Other Names list is OTHERNAME.

Headings for the Other Names list

The following table shows the column headings for the Other Names list. Only the NAME column is required.

NAME	(Required) A name or entry on the list.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Other Names list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
BADDR1	The first line of the person's address.
BADDR2	The second line of the person's address.
BADDR3	The third line of the person's address.
BADDR4	The fourth line of the person's address.
BADDR5	The fifth line of the person's address.
PHONE1	The person's phone number.
PHONE2	The person's alternate phone number.
FAXNUM	The person's FAX number.
NOTE	The name or number of the account you want to associate with this person.
CONT1	The name of your primary contact.
CONT2	The name of your secondary contact.
NOTEPAD	Your notes about this person. If you are creating an import file, the notes appear in the Notepad window for this person.
SALUTATION	The person's salutation, or title (Mr., Ms., Doctor, etc.).
COMPANYNAME	The name of the person's company.
FIRSTNAME	The person's first name.
MIDINIT	The person's middle initial.
LASTNAME	The person's last name.
HIDDEN	This list entry is hidden. Y/N
DELCOUNT	

PAYITEM (Payroll Item list)

Import and export files can contain all the information on your Payroll Item list. The keyword for an entry in the list is PAYITEM.

Column headings

The following table shows the column headings for the Payroll Item list. The NAME and PAYITEMTYPE columns are required.

NAME	(Required) The name of the payroll item.
DESC	Description of the payroll item
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Payroll Item list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
PAYITEMTYPE	(Required) Indicates the type of payroll item. If you are creating an import file, use one of these keywords to indicate the item type: **SALARY** For salaried wages. **HOURLY** For hourly wages. **COMMISSION** For commissions. **ADD** For bonuses or other additions to gross or after-tax pay. **DEDUCT** For union dues, loan repayments, employee-paid insurance, employee deductions for a pension plan. COMPANY For company-paid contributions associated with each pay period.
SICK	Sick pay.
LUMPVAC	Lump vacation pay
VACATION	Vacation pay
FEDTAX	Federal Income Tax
CPP	Canada Pension
EI	Employment Insurance
QPP	Quebec Pension
PROVTAX	Quebec Income Tax
DEFAULTAMT	The rate or amount of the item.If you are creating an import file, be sure to include a percent sign if the item is based on a percentage rate. If the item is to be subtracted from employees' pay, include a minus sign before the rate or amount.
LIMIT	The annual limit, in dollars, of the payroll item. For deductions from pay, this is the maximum amount that can be deducted from an employee's pay in one year. For additions to pay, this is the maximum amount that can be added to an employee's pay in one year.If you are creating an import file, be sure to enter an minus sign for all deductions.
EXPACCT	The name of the expense account you use to track company expenses related to payroll.
LIABACCNT	The name of the liability account you use to track money you deduct, but then owe as a liability to the federal, a state, or a local government.
TAXAUTHORITY	The agency to whom you make this payment
T4TRACKING	The box this item goes to on the T4 slip.
ADDJUSTGROSS	(ADD and DEDUCT payroll item types) Indicates whether the item affects gross pay. If you are creating an import file, enter one of these keywords in the ADDJUSTGROSS column: **Y** Yes.

N No.

BASEDONQUANTITY (one word) (ADD, DEDUCT, COMPANY payroll item types). Indicates that you want QuickBooks to calculate the amount of the item by multiplying its default rate or amount by a quantity you specify whenever you use the item on a pay cheque.If you are creating an import file, enter one of these keywords in the BASEDONQUANTITY column:

Y Yes.

N No.

SPECIAL Adds additional information that QuickBooks needs to know about the payroll item.If you are creating an import file, the keywords you can use are listed below. If none of the keywords apply, enter NO. Several of these items relate to Quebec, Canada. You should check that your customer's region doesn't have additional special fields.

SALARY	Indicates that the item tracks salary.
SICKSALARY	Indicates that the item tracks sick salary pay.
VACSALARY	Indicates that the item tracks vacation salary.
SICKHOURLY	Indicates that the item tracks sick hourly pay.
VACHOURLY	Indicates that the item tracks vacation hourly pay.
SICK	Indicates that the item tracks sick hours.
LUMPVAC	Indicates that the item tracks accrued vacation pay paid out.
VACRETAINED	Indicates that the item tracks accrued vacation pay.
VACPAIDOUT	Indicates that the item tracks vacation paid out.
FEDTAX	Indicates that the item tracks federal income tax.
COCPP	Indicates that the item tracks company CPP.
EECPP	Indicates that the item tracks employee CPP.
COEI	Indicates that the item tracks company EI.
EEEI	Indicates that the item tracks employee EI.
COQPP	Indicates that the item tracks company QPP.
EEQPP	Indicates that the item tracks employee QPP.
QUETAX	Indicates that the item tracks Quebec tax.

ISEMPREC Indicates that the payroll item tracks the employee share of a tax that both the employee and the company pay. If you are creating an import file, enter one of these keywords in the ISEMPREC column:

Y Yes.

N No.

ALLOCOST

Y Yes.

N No.

HIDDEN Is this payroll item hidden?

Y Yes.

N No.

DELCOUNT

Y Yes.

N No.

USEID

PAYMETH (Payment Method list)

Import and export files can contain the list of payment methods you use in QuickBooks. The keyword for the Payment Method list is PAYMETH.

Headings for the Payment method list

The following table shows the headings that QuickBooks recognizes in the Payment Method list. Only the NAME column is required.

NAME	(Required) The name of the payment method.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Payment Method list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
HIDDEN	This list entry is currently hidden.

WARNING: It's been reported that the 2003 version no longer allows a payment method to reduce a liability account. What this means is if you are using a payment item to such as 'GiftCardPmt' to track payments made by gift cards and reducing a GiftCardLiability account with this payment method, it will no longer work in QB 2003. You will have to use a Discount item instead of a payment item for the gift cards. The error QB import will give is "The Posting Account is Invalid".

QBP EMPLOYEE (Employee payroll records)

Import and export files can contain the payroll records for your employees. Because of the complexity of payroll records, there are four keywords for payroll records: QBP EMPLOYEE, CUSTOMPI, HOURLYPI, LOCALPI. Each keyword has its own column headings. For each employee record, there is one QBP EMPLOYEE line (or row). Immediately following, there can be up to 10 CUSTOMPI lines, up to 8 HOURLYPI lines, and up to 2 LOCALPI lines. The keyword ENDQBPEMP (in a line by itself) indicates the end of an employee's record.

Columns for QBP EMPLOYEE

The following table shows the column headings for QBP EMPLOYEE. The NAME and PAYPERIOD columns are required.

NAME	(Required) The name of the employee.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Employee list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
SALARY	The employee's yearly salary. Omit this field if you pay the employee an hourly wage.
PAYPERIOD	(Required) A keyword that identifies the pay period for the employee.If you are creating an import file, use one of these keywords: **DAILY** You pay the employee each business day. **WEEKLY** You pay the employee once a week. **BIWEEKLY** You pay the employee every two weeks (26 times a year). **SEMIMONTHLY** You pay the employee twice each month (24 times a year). **MONTHLY** You pay the employee once a month (12 times a year). **QUARTERLY** You pay the employee once a quarter (4 times a year). **YEARLY** You pay the employee once a year.
CLASS	The name of the class that applies to the employee. If the class is a subclass, the class name includes the names of the parent classes, beginning with the highest-level class. A colon (:) separates each class name.
PROVINCE	Province where this employee lives.
NUMCUSTOM	The number of "custom" payroll items in the employee's payroll record. A custom payroll item can be either a deduction, addition, comission, or company contribution payroll item. If there are no custom payroll items in the employee's record, enter 0.
NUMHOURLY	The number of hour wage payroll items in the employee's payroll record. If there are none, enter 0.
HIREDATE	The date you hired the employee. The date is always in MM/DD/YY format. For example, 1/30/94.
RELASEDATE	The date you released the employee. The date is always in MM/DD/YY format. For example, 1/30/94.
SICKACCRL	The sick time accrual period. **SVA_PERIOD** Every pay period **SVA_YEAR** Beginning of year
SICKRATE	The rate of pay for sick leave.
SICKACCRD	The amount of hours the employee can accrue for sick leave.
SICKLIMIT	The maximum number of hours the employee can accrue for sick leave.
VACACCRL	Vacation Accrual

	SVA_PERIOD	Every pay period
	SVA_YEAR	Beginning of year
VACRATE	The rate of pay for vacation leave	
VACACCRD	The amount of vacation pay the employee has accrued.	
DOB	The employee's date of birth.	
HIREDATE	The date the employee was hired.	
RELEASEDATE	The date the employee was released from the company.	
TD1CLAIM	The employee's federal tax claim (form TD1).	
QUEALLOW	The employee's Quebec tax claim.	
FEDEXTRA	Extra income tax deduction from this employee's pay.	
EIFACTOR	The employee's EI factor.	
FEDTAX	Is this employee subject to Federal Income? Y/N	
CPP	Is this employee subject to CPP? Y/N	
QPP	Is this employee subject to QPP? Y/N	
EI	Is this employee subject to EI? Y/N	
QUETAX	Is this employee subject to Quebec tax? Y/N	
TIMECARD	Do you use time datA to create this employee's pay cheques? Y/N	
CARRYSICK	Carry accrued sick time over to next year? Y/N	
CARRYVAC	Carry accrued vacation time over to next year? Y/N	
SICKPERPAY	Is sick time accrued every pay period? Y/N	
VACPERPAY	Is vacation pay paid out every pay period? Y/N	

Columns for CUSTOMPI

The following table shows the column headings for CUSTOMPI (custom payroll items). Each employee record can have up to 10 custom payroll items. The PAYITEM column is required.

PAYITEM	(Required) The name of the deduction, addition, commission, or company contribution payroll item.
AMOUNT	The rate or amount of the payroll item.If you are creating an import file, be sure to include a percent sign is the item is based on a percentage rate. If the item is to be subtracted from employees' pay, include a minus sign before the rate or amount.
LIMIT	The annual limit, in dollars, of the payroll item. For deductions from pay, this is the maximum amount that can be deducted from an employee's pay in one year. For additions to pay, this is the maximum amount that can be added to an employee's pay in one year.If you are creating an import file, be sure to enter an minus sign for all deductions.

Columns for HOURLYPI

The following table shows the column headings for HOURLYPI (hourly wage payroll items). Each employee record can have up to 8 hourly wage payroll items. Both columns are required.

PAYITEM	(Required) The name of the hourly wage payroll item.
AMOUNT	(Required) The hourly rate for the employee.

SHIPMETH (Ship Via list)

Import and export files can contain the list of shipping methods you use in QuickBooks. The keyword for the Ship Via list is SHIPMETH.

Column headings

The following table shows the column headings for the Ship Via list. Only the NAME column is required.

NAME	(Required) The name of the shipping method.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Ship Via list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
HIDDEN	This list entry is currently hidden. **Y/N**

Transactions (TRNS and SPL)

Import files can contain information about transactions. Note that you cannot export transactions from QuickBooks.
The keyword for a transaction is TRNS; the keyword for each distribution line is SPL.
The distribution lines appear immediately below the transaction. The keyword ENDTRNS appears after the last distribution line. In an import file, the first column for a transaction that has three distribution lines would look like this:

```
TRNS
SPL
SPL
SPL
ENDTRNS
```

Column headings for transactions (TRNS)

The following table shows the column headings for transactions. The TRANSTYPE, ACCNT, and AMOUNT columns are required.

TRNSID	A unique number that identifies the transaction.
TIMESTAMP	A second unique number that works with TRNSID to identify the transaction.
TRNSTYPE	(Required) A keyword that identifies the type of transaction. These keywords can appear in the TRNSTYPE field:
	BEGINBALCHECK Transactions that create a beginning balance in a balance sheet account
	BILL Bills from vendors
	BILLCCARD Bill payments you make by credit card
	BILLPMT Bills payments you make by cheque.
	BILL REFUND Refunds from a vendor
	CASH REFUND Cash refunds you give to customers
	CASH SALE Sales receipts
	CCARD REFUND Refunds you receive on credit card charges
	CHEQUE Cheques
	CREDIT CARD Charges you make on a credit card
	CREDIT MEMO Credit you give to customers for merchandise they return
	DEPOSIT Bank or money market deposits
	ESTIMATE (QuickBooks Pro only) Estimates or bids
	GENERAL JOURNAL General journal transactions
	ITEMADJ Adjustments for inventory.
	LIABADJ Adjustments for the amount you owe for a payroll liability
	LIABCHEQUE Payroll liability cheques
	INVOICE Invoices
	PAYCHEQUE Pay cheques
	PAYMENT Customer payments
	PURCHORD Purchase orders
	TRANSFER Transfers of funds from one balance sheet account to another
DATE	The date of the transaction. The date is always in MM/DD/YY format. For example, 1/30/98.
ACCNT	(Required) The name of the account assigned to the transaction.

NAME	The name of the customer, vendor, payee, or employee.
CLASS	The name of the class that applies to the transaction. If the class is a subclass, the class name includes the names of the parent classes, beginning with the highest level class. A colon (:) separates each class name.
AMOUNT	(Required) The amount of the transaction. Debit amounts are always positive, credit amounts are always negative.
DOCNUM	The number of the transaction. For cheques, the number is the cheque number; for invoices, the number is the invoice number; etc.
MEMO	The memo text associated with the transaction.
CLEAR	Indicates whether the transaction has cleared. These keywords can appear in the CLEARED field: **Y** Yes. The transaction has cleared. **N** No. The transaction hasn't cleared.
TOPRINT	Indicates whether a cheque, invoice, credit memo, or sales receipt has been marked as "To be printed." These keywords can appear in the TOPRINT field: **Y** Yes **N** No
ADDR1	The first line of the customer's, vendor's, payee's, or employee's address.
ADDR2	The second line of the customer's, vendor's, payee's, or employee's address.
ADDR3	The third line of the customer's, vendor's, payee's, or employee's address.
ADDR4	The fourth line of the customer's, vendor's, payee's, or employee's address.
ADDR5	The fifth line of the customer's, vendor's, payee's, or employee's address.
DUEDATE	(Bills and invoices only) The due date of the bill payment or invoice payment. The date is always in MM/DD/YY format. For example, 1/30/94.
TERMS	(Invoices only) The terms of the invoice.
PAID	(Invoices only) Indicates whether an invoice has been paid in full. These keywords can appear in the PAID field: **Y** Yes. The invoice has been paid in full. **N** No. The invoice is either partially paid or unpaid.
PAYMETH	(Sales receipts only) The method your customer used to pay for the merchandise.
SHIPVIA	(Invoices and sales receipts only) The method you used to ship the merchandise.
SHIPDATE	(Invoices and sales receipts only) The shipping date. If you are creating an import file, enter the date in MM/DD/YY format. For example, 1/30/94.
REP	(Invoices and sales receipts only) The initials of the sales representative or employee who made the sale.
FOB	(Invoices, credit memos, and cash sales only) The location where you delivered the merchandisefree of chargeto a carrier for shipment.
PONUM	(Invoices and credit memos only) The customer's purchase order number.
INVTITLE	(Invoices, credit memos, and sales receipts only) The title that appears on the invoice, credit memo, or sales receipt.
INVMEMO	(Invoices, credit memos, and sales receipts only) Your message to the customeras it appears on the invoice, credit memo, or sales receipt.

SADDR1 (Invoices and sales receipts only) The first line of the customer's shipping address.

SADDR2 (Invoices and sales receipts only) The second line of the customer's shipping address.

SADDR3 (Invoices and sales receipts only) The third line of the customer's shipping address.

SADDR4 (Invoices and sales receipts only) The fourth line of the customer's shipping address.

SADDR5 (Invoices and sales receipts only) The fifth line of the customer's shipping address.

TAXCODE (Invoices and sales receipts only) Indicates whether the customer whose name appears in the transaction has a customer tax code.

Y Yes.

N No.

PAYITEM (Payroll) The name of the payroll item in a transaction.

YEARTODATE (Payroll) The amount of a year-to-date adjustment transaction for an employee.

WAGEBASE (Payroll) The total amount of employee wages or earnings on which a payroll tax is calculated in a transaction.

Column headings for distribution lines (SPL)

The following table shows the column headings for distribution lines:

SPLID (Required) A unique number that identifies the distribution line in the transaction.

TRNSTYPE A keyword that indicates the type of transaction. The keyword in this field must match the keyword in the TRNSTYPE field for the transaction. The keyword will be one of the following:

BILL Bills from vendors

CHEQUE Cheques

DEPOSIT Bank or money market deposits

INVOICE Invoices

PAYMENT Customer payments

DATE The date of the transaction. The date in this field must match the date in the DATE field for the transaction.

ACCNT (Required) The income or expense account to which you assigned the amount on the distribution line.

NAME The name of the customer, vendor, payee, or employee.

CLASS The name of the class that applies to the distribution amount. If the class is a subclass, the class name includes the names of the parent classes, beginning with the highest level class. A colon (:) separates each class name.

AMOUNT (Required) The distribution amount. Credit amounts are negative.

DOCNUM The number of the transaction. For cheques, the number is the cheque number; for invoices, the number is the invoice number, etc.

MEMO The memo text associated with the distribution line.

CLEAR Indicates whether the distribution amount has cleared. These keywords can appear in the CLEARED field:

Y Yes. The amount has cleared.

N No. The amount hasn't cleared.

PRICE The unit cost of the item.

QNTY The number of items sold. This value is part of a line item on an invoice, credit memo, or sales receipt.

INVITEM The type of items sold. This value is part of a line item on an invoice,

credit memo, or sales receipt.

PAYMETH On a sales receipt, indicates the method of payment (cheque, Visa, etc.) that the customer used.

VALUEADJ (Inventory) Indicates whether the amount in a detail line is an inventory value adjustment.

- **Y** Yes. The amount is an inventory value adjustment.
- **N** No. The amount is not an inventory value adjustment

REIMBEXP Indicates the status of the distribution amount as a reimbursable expense. These keywords can appear under REIMBEXP:

- **NONEED** Indicates that the distribution amount does not qualify as a reimbursable expense.
- **NOTHING** Indicates that the distribution amount can be billed to a customer as a reimbursable expense.
- **THISWAS** Indicates that the distribution amount is a reimbursable expense that appears on an invoice or sales receipt.
- **HASBEEN** Identifies a distribution amount on a cheque that has been billed to a customer as a reimbursable expense.

TAXCODE Indicates the tax code assigned to this line item.

OTHER1 Indicates the custom fields you may have added to your sales form.
OTHER2
OTHER3

PAYITEM (Payroll) The name of the payroll item in a transaction.

YEARTODATE (Payroll) The amount of a year-to-date adjustment transaction for an employee.

WAGEBASE (Payroll) The total amount of employee wages or earnings on which a payroll tax is calculated in a transaction.

EXTRA Adds additional information about the distribution line. These keywords can appear in the EXTRA field.

- **ENDGRP** (Invoices, credit memos, and sales receipts only) Indicates that the distribution line is the last item of an invoice item group.

GST or **PST** Identifies a sales tax code applied to this transaction.

TERMS (Terms list)

Import and export files can contain all the information in your Terms list. The keyword for the Terms list is TERMS.

Headings for the Terms list

The following table shows the column headings for the Terms list. Only the NAME column is required.

NAME	(Required) Your name for the payment terms.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Terms list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
DISCPER	The discount percentage earned for early payment. If you are creating an import file, enter the percent number.
DISCDAYS	The number of days by which an early payment can earn the discount specified by DISCPER.
TERMSTYPE	Indicates the type of terms you are setting up. Type 0 for "standard terms" (payment to be made within a specific number of days), or type 1 for "date driven" terms (payment to be made by a certain date of the month).
STDDUEDAYS	Due in this number of days (Standard –driven by number of days).
STDDISCDAYS	Discount if paid in this number of days.
DAYOFMONTHDUE	Due this day of month (Date Driven).
DISCDAYOFMONTH	Discount if paid by this day of the month.
DATEMINDAYS	Due the next month if issued within this many days of due date.
HIDDEN	This list entry is hidden. **Y/N**

TIMEACT (activites timed with the Timer)
(QuickBooks Pro only)

You can import data about the activities you time with the Timer to QuickBooks Pro. The keyword for activity data is TIMEACT. Timer activity data must begin with the Timer header block (TIMERHDR).

Column headings

The following table shows the column headings for activity data:

DATE	The date the activity was performed. Enter the date in a MM/DD/YY format. For example, you would enter June 21, 1997 as 06/21/97.
JOB	The name of the customer (or job) for whom the activity was performed. If you're entering the name of a job, enter the customer's name followed by a colon followed by the name of the job. Both the customer and the job names must also be on your Customer:Job list (CUST).
EMP	The name of the employee who performed the activity. The employee's name must also be on your Employee list (EMP).
ITEM	The name of the service item assigned to the activity (a service item indicates the type of work performed). The service item must also be on your Item list (INVITEM).
PITEM	The type of payroll items. Leave blank. This field is not used in QuickBooks 5.0.
DURATION	The duration of the activity in hours and minutes. Enter the duration in an HH:MM format. For example, if the activity lasted 4 hours and 15 minutes, you would enter 4:15. The duration cannot be greater than 23:59.
PROJ	The QuickBooks Pro class assigned to the activity (classes give you a way to group activities in meaningful ways in time reports). The Class must also be on your Class list (CLASS)
NOTE	Your notes about the activity. You may enter up to 1000 characters, but the characters must be on a single line within the field. To indicate a line break in the note, enter \n.
BILLING STATUS	Indicates the billing status of the activity. Enter one of these values: **0** The activity is not billable to the customer or job specified in the Job field. Enter this value if the activity is part of your company overhead. **1** The activity is billable but has not yet been billed to the customer or job specified in the Job field. **2** The activity has already been billed to the customer or job specified in the Job field.

TIMERHDR (Header for Timer data)

(QuickBooks Pro only)

Data you export from or import to the QuickBooks Pro Timer always begins with a header block. The keyword for the header is TIMERHDR. The header block can be followed by either activity data (TIMEACT) you are importing from the Timer, or list data (CUST, EMP, INVITEM, VEND, OTHERNAME, and CLASS) you are importing to the Timer.

Column headings

The following table shows the column headings that appear in the header:

VER	The version number of the QuickBooks Pro Timer.
REL	The release number of the QuickBooks Pro Timer.
COMPANYNAME	The name of your QuickBooks Pro company. Enter the name as it appears in the Company Information window in QuickBooks Pro.
FROMTIMER	Indicates whether you exporting data from the Timer or importing data to the Timer. Enter one of these values: **Y** If you are importing activity data from the Timer to QuickBooks Pro. **N** If you are exporting list data from QuickBooks Pro to the Timer.
COMPANYCREATETIME	A unique number that comes from your QuickBooks Pro company file when you export lists to the Timer. The number identifies which company file the lists came from. You can see an example of this number by exporting your lists to the Timer and then opening the .IIF file that QuickBooks Pro creates.

TODO (To Do list)

Import and export files can contain all the information on your To Do list. The keyword for the To Do list is TODO.

Column headings

The following table shows the column headings for the To Do list.

REFNUM	(Export files only) A unique number that identifies an entry in the list.
ISDONE	Indicates whether the task is complete. If you are creating an import file, enter one of these keywords in the **ISDONE** column: **Y** Yes. **N** No.
DATE	The date you want QuickBooks to place the task on your Reminders list.
DESC	Your notes about the task.
HIDDEN	This list entry is currently hidden. **Y/N**

VEND (Vendor list)

Import and export files can contain all the information in your Vendor list. The keyword for the Vendor list is VEND.

Column headings

The following table shows the column headings for the Vendor list. Only the NAME column is required.

NAME	(Required) The name of the vendor.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Vendor list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
PRINTAS	The name you would like cheques to be made out to. This field allows you to make checks out to a different name than the name that appears on your Vendor list.
ADDR1	The first line of the vendor's address.
ADDR2	The second line of the vendor's address.
ADDR3	The third line of the vendor's address.
ADDR4	The fourth line of the vendor's address.
ADDR5	The fifth line of the vendor's address.
VTYPE	Your classification for the vendor (QuickBooks calls this a "vendor type"). If you import a vendor type that is not on your Vendor Type list, QuickBooks adds the new vendor type to the list.
CONT1	The name of your primary contact with the vendor.
CONT2	The name of your secondary contact with the vendor.
PHONE1	The vendor's phone number.
PHONE2	The vendor's alternate phone number.
FAXNUM	The vendor's FAX number.
NOTE	A short note or phrase you want to associate with the vendor. QuickBooks automatically puts your note in the Memo field of cheques you send to the vendor.
TAXID	The vendor's tax identification number.
LIMIT	Your credit limit with the vendor. If you are creating an import file, enter the dollar amount.
TERMS	Your payment terms with the vendor.
NOTEPAD	Your notes about the vendor. If you are creating an import file, the notes appear in the Notepad window for the vendor.
SALUTATION	The vendor's salutation, or title (Mr., Ms., Doctor, etc.).
COMPANYNAME	The name of the vendor's company.
FIRSTNAME	The vendor's first name.
MIDINIT	The vendor's middle initial.
LASTNAME	The vendor's last name.
CUSTFLD1 **CUSTFLD2** **...** **CUSTFLD7**	The custom field entries for the vendor (you can have up to 7 custom field entries). Custom fields let you track special information about the vendor, such as the vendor's birthday or email address. What you use custom fields for is entirely up to you.
HIDDEN	This list entry is currently hidden. Y/N
DELCOUNT	

VTYPE (Vendor Type list)

Import and export files can contain the list of vendor types you use in QuickBooks. The keyword for the Vendor Type list is VTYPE.

Column headings

The following table shows the column headings for the Vendor Type list. Only the NAME column is required.

NAME	(Required) The vendor type. If the vendor type is a subtype of another vendor type, the name includes the name of the parent types, beginning with the highest level type. If you are creating an import file, use a colon (:) to separate subtype names.
TIMESTAMP	(Export files only) A unique number that identifies the company file from which you exported the Vendor Type list.
REFNUM	(Export files only) A unique number that identifies an entry in the list.
HIDDEN	This list entry is currently hidden. **Y/N**

Class Reference

There are several classes that are used by the QuickBooks-Export templates:

STAB::AE::QB::ExportManager

There is one of these in each QuickBooks Export procedure, to manage the multiple export extensions. The methods are as follows:

Init (SHORT ProgressStringControl, |
SHORT ProgressThermometerControl, |
SHORT ProgressPctTextControl)

This method initializes the object. The three parameters are the field equates for the procedure's progress window.

AddFile (STAB::AE::QB::FileClass FileObject)

This method will be called once for each export object managed by the procedure. The parameter is the export object itself.

Kill ( )

This method shuts down the object after everything is done.

TakeTimer ( )

This method is called whenever the EVENT:Timer fires. It calls the various methods in the export objects, and updates the progress window.

STAB::AE::QB::FieldClass

This class is responsible for the QuickBooks destination fields. There will be one field for each field of each destination. Each export extension has its own collection of field objects.

Init (STRING Name, *? Field, <STRING Picture>)

This method initializes the object with the QuickBooks name of the field, a reference to the source variable, and an optional picture (if its not to be formatted as a simple string).

Kill ( )

This method shuts down the field class.

GetValue ( ),STRING

This function returns the current value of the field, formatted appropriately.

STAB::AE::QB::FileClass

This class represents a non-TRNS QuickBooks destination, as well as the SPL portion of a TRNS. There will be one per export extension.

Init (STRING Name, | *!QuickBooks Name (e.g. CUST)*
STRING Description, | *!Description for Display*
<*ViewManager VM>, | *!View Manager to read records (opt.)*
*FILE OutputFile, | *!Pointer to OutputFile*
BYTE UpdateMethod) *!Update Method Equate*

This method initialize the object. The ViewManager parameter is omitted for an SPL file (because the source file is already being accessed by the TRNS' view).

Kill ( )

This method shuts down the object.

AddField (STRING Name, *? Field, <STRING Picture>)

This method is called to add each field to the destination object. The method, in turn, calls the **Init** method for it's own internal collection of **STAB::AE::QB::FieldClass** objects.

AssignStampField ( ),BYTE,VIRTUAL

This is a virtual function to assign the stamp value to the primary file. It doesn't do anything in the base class, and relies entirely upon the instantiating procedure to derive this method (if required). The derivation is handled automatically by the template, although you can certainly "roll your own".

CalcFields (<BYTE CalcBank>),VIRTUAL

This virtual method is responsible for performing expression calculations. If all of your expressions for destination fields are simple fields (i.e. no complex assignments), then this method may not even be used. However, virtually all applications will have at least one calculated expression.

CalcBank will be zero for normal calculations (for regular, TRNS and SPL destinations). If you're manually adding extra SPL lines, then you'll need to override this method (with code before the PARENT call), to interpret your own CalcBank numbers.

ExportRecord (<BYTE CalcBank>),BYTE,VIRTUAL

This method calls CalcFields(CalcBank) to finalize calculations, then calls WriteRecord to output the record to the disk.

This is the method that you'll call if you have to manually exporting extra SPL records. In those situations, be sure to pass a non-zero CalcBank, and to interpret that appropriately in your overridden CalcFields method.

Reset ( ),BYTE,VIRTUAL,PROC

This method calls the WriteHeader method, followed by the Reset and Open methods for the ViewManager. It's usually called by the ExportManager object.

TakeRecord ( ),BYTE,VIRTUAL,PROC

This method is called to read another record from the view, and respond appropriately. If a record is available, it calls the necessary methods to have it exported.

UpdateFile (<STRING ErrorMessage>),BYTE,VIRTUAL,PROC

This method is called to physically update the file after applying the stamp. If you want to update the record in some other manner, you could override this method to prevent the parent call. (If the update method is anything put PUT(VIEW), then a REGET view has already been called to properly align the record position.)

ValidateRecord ( ),BYTE,VIRTUAL

This method is called by TakeRecord as a final check to ensure that the record should be

exported. The base class merely returns LEVEL:Benign. If you have additional validation that cannot be handled by the filter, then override this method. If you want to skip a particular record, just return LEVEL:Notify.

WriteFooter (<STRING Prefix>),BYTE

For non-TRNS destinations, there is no footer required in the IIF file. This method is an empty virtual, waiting to be overridden by a derived class object.

WriteHeader ( ),BYTE,VIRTUAL

This method writes the destination header into the IIF file, based upon the field list specified during initialization.

WriteRecord ( ),BYTE,VIRTUAL

This method formats and writes the record out to the IIF file.

STAB::AE::QB::ParentFileClass

This class is derived from the **STAB::AE::QB::FileClass**. It handles TRNS destinations, for a parent/child scenario.

Init (STRING Name, | *!Name of QuickBooks destination*
STRING Description, | *!Description for Display*
*ViewManager VM, | *!View Manager to read records (opt.)*
*FILE OutputFile | *!Pointer to OutputFile*,
BYTE UpdateMethod, | *!Update Method Equate*
BYTE HeaderHasDetails) *!Header has details (i.e. no break)*

This method initializes the object. Notice that the parameters are almost the same as the **STAB::AE::QB::FileClass**, with the exception of the extra "HeaderHasDetails" (which would only apply for a parent/child situation).

Kill ( )

This method shuts down the object.

AddChildFile (*STAB::AE::QB::FileClass File)

This method is called with the name of the SPL export object.

AddFileManager (*FileManager FM)

This method is called to inform the object of the file buffers that must be saved and restored during header breaks.

ExportExtraSPLs:BeforeFirst ( ),BYTE,VIRTUAL,PROC
ExportExtraSPLs:AfterFirst ( ),BYTE,VIRTUAL,PROC
ExportExtraSPLs:AfterLast ( ),BYTE,VIRTUAL,PROC

These empty virtual methods are called from within the class, so that you can easily export your own extra SPL records. Recall that at least one SPL is automatically exported for each TRNS. You may want to export your extra SPL(s) before this first SPL, immediately after, or after *all* the SPL lines for that TRNS.

RestoreFileBuffers (BYTE Bank, BYTE DoRestore=1)
SaveFileBuffers (BYTE Bank)

These two methods are used by the class to save and restore buffers during header breaks. It gets its list of files from the calls to the **AddFileManager** method. If the **HeaderHasDetails** parameter was TRUE in the call to the **Init** method, then these methods will not be used.

TakeRecord ( ),BYTE,DERIVED,PROC

This method overrides the equivalent in the **STAB::AE::QB::FileClass**. Instead of just exporting the record, it automatically watches for header breaks (unless **HeaderHasDetails=FALSE**). If its the first record of the transaction, it outputs the TRNS line. If it's a child, it out the SPL line. After a header break (or at end-of-file), it calls the **WriteRecordFooter** method to end the transaction.

WriteHeader ( ),BYTE,DERIVED

This method overrides the equivalent in the **STAB::AE::QB::FileClass**. Instead of exporting a single-line header to mark the beginning of a group of records, it outputs the description of both TRNS and SPL lines.

WriteRecordFooter (<STRING Prefix>),BYTE,VIRTUAL

This method outputs an "ENDTRNS" marker after the TRNS and SPL lines.

Interface Modification and Translation

All user-viewable elements of the templates are available as regular window elements and template settings. Therefore, there is no TRN provided with this template set.

License Agreement

This product is comprised of the templates, default applications, libraries, source code, documentation and help files provided with the package. You must have a separate registered copy for each person using it.

You are allowed to use the Super Templates for any programs that you create, and you may distribute the generated source code. You may not, however, distribute any portion of the product in its original or modified form without the prior written consent from BoxSoft Development Inc. One exception to this is the example programs provided with the Super Templates or separately from BoxSoft or its agents: these may be distributed without penalty, whether in their original state or modified.

Contacting Technical Support

If you have any troubles with this product, then please contact:

Mitten Software
1865 West Wayzata Blvd
Suite 218
Long Lake, MN 55356

Voice: (952) 745-4941
Fax: (952) 745-4944

Internet: www.MittenSoftware.com
Mitten@MittenSoftware.com

www.boxsoft.net
mike@boxsoft.net

Troubleshooting

Problem: *QuickBooks won't import (and the error message isn't entirely helpful).*

Solution: You're probably including something that's unacceptable to QuickBooks. Here's what you should do to can find the cause:

- Take a look at your IIF file in Excel. Check that everything looks good. In particular, be sure to check that transactions balance, so that the sum of the detail amounts is the negative of the amount in the header.
- Simplify your export. Remove field assignments until you get something that does work. This will often clarify what's causing the problem. Then you can re-add the fields one (or a few) at a time, with the problem area fixed.
- Take a look at the sample files from Intuit. We've included some of these in the `SUPER\DOC\QuickBooks` directory, and you can find more on the QuickBooks web-site. Here are two useful links that we have found:

  www.quickbooks.com/support/faqs/docs/w_iiffiles.html
  www.quickbooks.com/support/faqs/docs/w_iiffiles2.html

- QuickBooks won't export Transaction records, which are usually the toughest to create. For all other file types, however, you can create a sample in QuickBooks of what you would like to see. Export this into an IIF file. Now you know which QuickBooks fields to use to get your data into the desired locations, and how that data needs to be formatted.

Problem: *You're importing an IIF file, and QuickBooks complains that it's got a duplicate customer or vendor.*

Solution: QuickBooks uses the same name space to identify Customers and Vendors. Therefore, the names (which are usually the customer and vendor numbers/codes in your Clarion database) must be unique across both sets of entities. If your Clarion database uses a similar numbering/coding scheme for both customers and vendors, then you will need to make the number unique for the export to QuickBooks.

The easiest way to do this is to preface the customer number with "C" and/or vendor number with "V". You may want to change only one entity, and leave the one with the same number/code. It's entirely up to you.

By the way, you must remember to alter the customer identifier for all other record types (e.g. Transactions) that refer to these customers and/or vendors. Otherwise, the related files will not be properly connected to the customer/vendor in QuickBooks.

Problem: *You've created a new QBExport procedure, but it doesn't have a window.*

Solution: Delete the procedure and re-create it while selecting on the "Default" tab on the procedure templates window.

Problem: It's been reported that the 2003 version no longer allows a payment method to reduce a liability account. The error QB import will give is "The Posting Account is Invalid". What this means is if you are using a payment item to such as 'GiftCardPmt' to track payments made by gift cards and reducing a GiftCardLiability account with this payment method, it will no longer work in QB 2003.

Solution: You will have to use a Discount item instead of a payment item for the gift cards.

Disclaimer

BoxSoft Development Inc. does not warranty this software for any use. Any expenses or lost time due to errors in this product are not the responsibility of BoxSoft Development Inc. We will attempt to fix any errors that we are made aware of, but we are not legally liable for any lack of correctness of the product.

✌■♎♏☒

✌

✌♍♍□◆■♦⬩

✌♋♋♓■♑

✌▱✌

✌☼♏➶♏❒♏■♍♏

✌

✌❒□⬩⬩♏

✌□■♦❒□●

✌✞☜

☎♌◆♎♑♏♦⬩◑

☝

☝☹✌♦♦

☎☝●♋⬩⬩

●♓⬩♦◑

☝□■♦♋♍♦♓■♑

❄♏♍♒■♓♍♋●
♦◆□□□❒♦

☝❄✡▱☜

☎☝◆⬩♦□❍♏❒

❄♲□♏
●♓⬩♦◑

☝✞♦❄

☎☝◆⬩♦□❍♏❒💻☺□♌

●♓⬩♦◑

☟

☟♋♦♏❄♓❍♏

♦♦♋❍□⬩

☟♓♍♦♓□■♋❒♲

☟♓⬩♍●♋♓❍♏❒

☟♏❍□⬩

☟♓❒♏♍♦□❒♓♏⬩

☜

☜💣▱

☎☜❍□●□♲♏♏

●♓⬩♦◑

☜☒♋❍□●♏⬩

☞

☞♓●♦♏❒

❄♋♑♑♏♎

☼♏♍□❒♎⬩

☞◆■♍♦♓□■

☼♏➶♏❒♏■♍♏

☝

☝●□♌♋●

☜☒♦♏■⬩♓□■
❄♏❍□●♋♦♏

☟

☟☟☼

☎♒♏♋♎♏❒
♓■➶□❒❍♋♦♓□■◑

✋

✋♍□■⬩

✋■⬩♦♋●●♋♦♓□■

✋❍□□❒♦

✋■♦♏❒➶♋♍♏

❄❒□◆♌●♏⬧♒□□♦♓■♑  ❄✠☜ 

✞

✞♋❒♓♋♌●♏ ☼♏➚♏❒♏■♍♏  ✞☜☠☜ ☎✞♏■♎□❒ ●♓⬧♦◑ 
✞❄✡☜ ☎✞♏■♎□❒ ❄⊠◘♏
●♓⬧♦◑