Global 3000 Applications

Version 6.0

Presentation Standards

August 2005

Contents

Introduction

This document is divided into two sections.

General Guidelines

This section contains general rules that must be applied consistently across all window and application types.

For example, a label is always displayed with the same case rules regardless of the type of window it is contained in.

Application Specific Guidelines

This contains guidelines that are specific to the application being developed.

For example transaction entry programs always display the list of batches window with a specific window layout and use a common set of buttons. Reports always prompt for selection dialogue in the same way.

General Guidelines

Windows Screen Resolution and Fonts

The minimum recommended screen size for Global 3000 V6.0 is 1024 x 768.

The recommended fonts are:

- Label font - Tahoma, Regular 10 pt
- Data font - Tahoma, Regular, 10 pt
- Button font - Tahoma, Regular, 10 pt

Other fonts and screen sizes can be used but these are the fonts that have been used in house for all our own development and QC.

Speedbase Window sizes and layout

The default size of windows no longer needs to be restricted to 80 x 24. The default size we have assumed is approximately 128 x 33, but this is only limited by screen size and fonts used.

When laying out new windows or updating existing ones, the following should be considered.

- Increase the depth of scrolled areas. The speedbase default when creating a window is eight but this should be increased if space permits.
- Use the FMT "," option to display values with comma separators in all windows.
- Use the FMT "L" option to display all dates in long format.
- Where we display a "code" show the description next to it.
- Do not use the LINE statement in any windows. Any divisions of fields should be achieved using group boxes.

Buttons

General

Try to keep the button names as meaningful as possible without getting too wordy and never use abbreviations. If in doubt the thesaurus in Word can usually supply good alternatives.

Just as a Pop-menu has a single letter selection character for each line, so a button can be defined with an [Alt]-[Char] selection letter, where the [Char] is the first upper case letter in the button text. This selection character is known as the buttons "Accelerator Key".

Regardless of where, or if, a caption has an upper case character, the caption is always converted to sentence case by GX before being displayed, so a caption of "aCtion" would appear on the button as "Action", the "c" is underlined to indicate that <Alt><C> can be used as the accelerator key to activate it.

Do not allow duplicate accelerator keys.

Where a buttons caption is a data item, leaving it blank indicates that the button should be removed from the window. This functionality should always be used before a window is displayed, rather than having buttons appearing and then vanishing.

Remember that to get buttons to the correct width, the button text should be padded with leading spaces. The Captions are always centred by GX when the button is displayed.

The GXSBT$ routine can be used to either alter the caption of existing buttons, or to remove a button completely.

Remember that many people will still use the keyboard for a lot of input so it is important to ensure that the dialogue dialogue remains fluent in keyboard, mouse or a combination of both and at the risk of stating the obvious, both implementation & review should check for fluency in the new presentation.

Navigation Buttons

These are buttons that return $FUNC values in the range 1-21, no additional code is required to process one of these buttons as the Window Manager already knows what the required action for each button is.

By default if there are no user defined buttons, GX will generate an OK and a Cancel navigation button at the bottom of window, right aligned.

- Size – eight characters.
- Only items listed in the table of standard navigation buttons should be used.
- Cancel or Close are always the last button.
- Next and Back do not have a gap between them.
- Amend, Apply and Close do not have accelerator keys.
- The “OK” button must be all in upper case and not specified as “Ok”.
- Always displayed on the right hand side of the window.

Standard Navigation Buttons			
Amend	Places the cursor into the first maintainable field on the currently highlighted record. This is only used on scrolled windows.	U21	-
Apply	Becomes enabled once the Amend function has been selected. It confirms acceptance of all maintainable fields in the current window. This is only used on scrolled windows.	U20	-
Close	Close this window.	U11	$F-BCK
Next >	Accept all values shown and proceed to the next window in the chain. This button is usually only used on EDT windows. If it is required on a window that also has a maintainable scrolled record set (and therefore an Apply button that already uses U20). Use U04 and make the window a REPEAT UNTIL NXT.	U20 U04	- $F-NXT
< Back	Return to the previous window in the chain.	U11	$F-BCK
OK	Confirm the details in the current window and continue.	U20	-
Cancel	Terminate the current process. This button should only be used where you are sure that the data will be in the same state after cancel as it was prior to entering the process. If you are cancelling a single process e.g. the selection of an item from a search window, use U11 ($F-BCK) If you are cancelling an entire process, e.g. the running of a report, use U10 ($F-ABO)	U11 U10	$F-NXT $F-ABO
Select	Used where the user is being asked to select from a list of records.	U21	-

Application Buttons

These are buttons that perform tasks specific to the application. They do not (usually) correspond to existing speedbase function keys and return $FUNC values in the range 50-99.

For each application button defined, the programmer will need to add code to the R-FUNC routine that performs the required processing when the button is clicked.

Application buttons when added to a window are positioned left aligned. This means that a typical window could have two banks of buttons. If there are sufficient buttons on a window it is possible to end up with one contiguous bank of buttons or possibly even multiple rows of button (the 32-bit debugger being a prime example).

- Size - The width of the application buttons should ideally also be eight characters but may be dictated by the length of the text that needs to go on it. All buttons in the application bank must be the same size, i.e. sized to the largest button text, shorter captions will need to be padded out with leading spaces to bring them up to the required size.

- Always displayed on the left hand side.

Standard Application Buttons	
Delete	Used where a high-level delete function is available (i.e. not F7). For example delete batch, delete customer etc.
Drill Down	Used where a further level of enquiry drill-down is available.
Find	Used to invoke the standard find processing dialogue.
Lines	Use on windows that list batch / header details. The lines button takes the user down to the next level.
Notes	This is used to access any additional notes that relate to the current transaction, e.g customer notes, supplier notes etc.
Post	Use on any list of batches windows to invoke the post batch process on the current item.
Print	Any print functions available from within maintenance / enquiry applications should be invoked using this function. If more than one print function is available, selecting this option should display a radio button menu listing the available options.
Search	This should only appear on windows that have search options (UF1) on fields in the scrolled area of the window. Do not add a search button if the only fields with UF1 options are in the non-scrolled area, or contained within an EDT window. When it is used, this should always be the first application button.
Views	Used where a window supports multiple views of the data. Using this function will present a further window listing the available views. If there are only two views then use the button to toggle between them changing the button label as appropriate.

Labels

A label is any item of text that appears within a speedbase window. They are normally used to identify an associated item of data.

Scrolled labels

- Left aligned above text data items
- Right aligned above numeric data items. You should also add the RJF attribute so that the label correctly aligns with the end column when using proportionally spaced fonts.

Non-scrolled labels

- First letter of first word is capitalised, remainder all in lower case
- Columns of labels are left aligned
- Allow at least one space between the end of a label and it's associated data item
- Data items are displayed to the right of their labels
- Do not use abbreviations in labels

Tick Box labels

- Always to the right of the tick box.
- Label should be phrased as a short question and end with a question mark.

Group Boxes

A standard windows group box can now be drawn around related fields and given a title. This should be introduced to as many windows as possible as it not only dramatically improves the look of the product, but also allows related fields to be grouped for greater clarity.

Detailed programming notes can be found in Appendix B

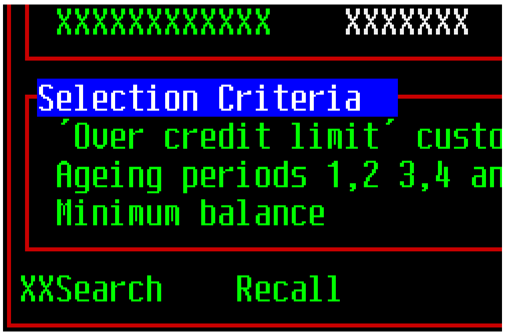
Definitions

- There should be no blank lines before or after a group box.
- Where possible a group box should only contain related fields.
- A group box should have a title that relates to the fields contained within the group.
- Each word of a group box title should be capitalised.
- Where multiple group boxes are present on the same window, they should be symmetrical. i.e. do not have groups boxes lined up vertically of differing widths.
- Where group boxes are displayed next to each other, they should be the same depth. In extreme cases it is fine to have two or more group boxes aligned vertically that in total are the same depth as another single group box displayed next to them.
- When a group box is displayed below a scrolled are, the group box(s) must be the width of the scrolled area.
- When a group box is displayed in a window with no scrolled area, the width of the group box(s) defines the width of the window.
- Group boxes always start in column 2.

Data items contained within Group boxes

- All non-scrolled labels and data items must be contained within a group box.
- With the exception of System Parameter windows, do not leave blank lines at the start of a group box.
- Blank lines should only be left at the end of a group box if it is to make the group box symmetrical with other group boxes displayed on the same window.
- Labels should be positioned directly next to the left hand edge of the group box.
- Data items should be left aligned for text items, and right aligned at the decimal point for numeric items. When there are a mixture of data items in the same group a certain amount of artistic licence might need to be used to determine the most suitable alignment and position with regards to the group box border.
- Fields only need to be aligned within their own group box, not across all group boxes.
- Try to avoid having blank lines between related prompts as visually this implies fields are not related.
- Where display only fields of varying widths are displayed, you should consider displaying fields with the same picture clause. E.g. a Column contains a mixture of 9(6,2) and 9(7,2) values should all be displayed as 9(7,2) to give a uniform look.

Buttons positioned below a Group box



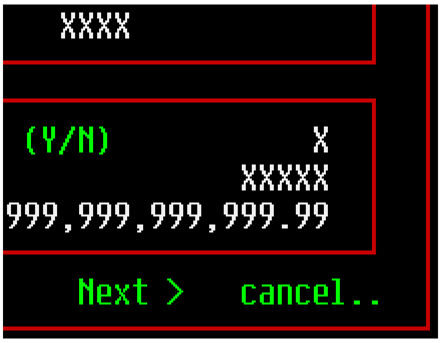
```
  XXXXXXXXXXXX    XXXXXXX

 Selection Criteria
   'Over credit limit' custo
  Ageing periods 1,2 3,4 an
  Minimum balance

XXSearch     Recall
```

- Buttons are positioned immediately below the group box.

- First button appears at Column 2, the same column as the group box starts.



```
   XXXX


 (Y/N)               X
                 XXXXX
999,999,999,999.99

    Next >    cancel..
```

- The last buttons last character appears two characters before the end of the group box.

  This ensures that the window displays with symmetrical margins between the edge of the group box and the edge of the window on the left and right side.

Combo Boxes

Fields that allow one item to be selected from a simple list of valid inputs should be processed as a combo box. A lot of prompts ask you to enter a single character to represent a particular option and then (sometime) display a bit of narrative to the right expanding on the name of the option selected.

Replacing these with combo boxes allows the user to select from a list of the descriptions rather than cryptic one character inputs.

- A combo box should always be given a default value.
- You need to allow 3 characters between the end of the group box description and any other field or group box border to allow GX room to add the combo box drop down button.
- It is recommended that a combo box list is limited to a maximum of twenty items. More than that and it could become difficult for a user to find the selection they want as there are no options to re-sequence a combo list or do partial key searches etc like you can with a speedbase window.
- A combo box is always the width of the longest description.
- Where multiple combo boxes appear above one another, they should if possible be the same width. Do not however inflate for example a 10 character combo box to match a 40 character one above it.

Tick Boxes

A tick box can be used in place of YN handling in most places, but is especially relevant to system parameter windows, report selection options etc

- Tick boxes are always displayed to the left of the label describing the option they enable/disable.
- Try to place all tick boxes that exist within a group box in a vertical group.
- Any other non-tick box prompts within the same group box should be separated from them by a blank line.
- The prompts should be phrased as a short impersonal question. i.e. instead of "Do you want a summary report?", the prompt should be "Summary Report?". Do not include any (Y/N) text in the prompt.

Radio Buttons

Radio buttons can be used when there is a sequence of options and only one can be selected. It can also be used to replace pop-menus added in earlier versions of the software, though many pop-menus are better replaced by placing applications buttons onto the target windows.

- Radio buttons look best lined up vertically (like a menu), but can also be displayed horizontally.

- Radio button options can be "greyed-out" in the same way that a pop-menu line can. Unlike a pop-menu, you cannot select the greyed out option to receive an error message.

- When replacing a pop-menu add "Next" and "Cancel" buttons and remove the "Exit" line from the list if present. Programmers should ensure that the "Cancel" processing follows the same pathway as the removed "Exit" option.

- Always set a default option when first displaying a window containing a set of radio buttons.

Message Boxes and Baselines

There are a number of warning/message box type standards.

Warning Message

Displayed where ever the software needs to put out a single warning message before proceeding with a process.

- Use the “exclamation” icon in the call to MBOX$.
- The text should start with “Warning: “.
- Use a single OK button.

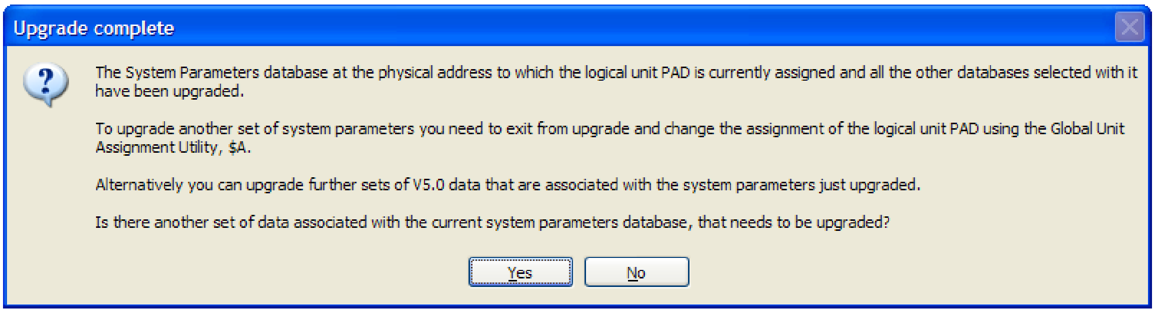
Error Message

Displayed where an error situation has occurred which prevents a process from continuing, in most cases control is returned to the user to allow any invalid input to be corrected. The ERROR statement already generates messages in a similar format.

- Use the “exclamation mark” icon in the call to MBOX$ if an error is correctable.
- Use the “stop” icon in the call to MBOX$ if the process is being terminated.
- Use a single OK button.

Confirmation Window

Where a user is being asked a question that requires a response of “yes” or “no”, use an MBOX$ call for this rather than a speedbase window.



The System Parameters database at the physical address to which the logical unit PAD is currently assigned and all the other databases selected with it have been upgraded.

To upgrade another set of system parameters you need to exit from upgrade and change the assignment of the logical unit PAD using the Global Unit Assignment Utility, $A.

Alternatively you can upgrade further sets of V5.0 data that are associated with the system parameters just upgraded.

Is there another set of data associated with the current system parameters database, that needs to be upgraded?

- Use the question mark icon

A standard call (GX%YN) has been provided for this. Pass the routine a pointer to a #00 terminated block of text, and a pic x(132) title field. A response of “No” returns an exception.

Baseline Messages

Use the GXMSG$ routine if you need to display the equivalent of a baseline message for a particular field.

The routine should be invoked from the B- of the relevant field when not in display mode.

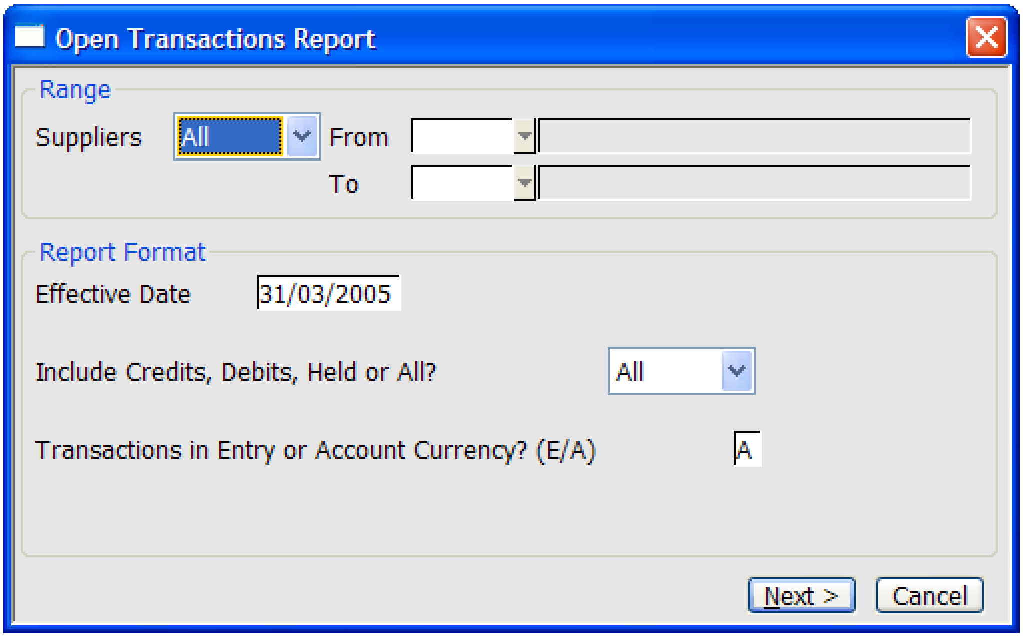
Remove the baseline message window in the B- of the next field. In addition it may be necessary to add calls to remove the baseline message window in the B- routines of preceding fields in case the user moves the cursor to an earlier field in the window.

Range Selection

The following apply mostly to report selection criteria but is applicable to anywhere the user is prompted to enter a range of items.

Range, Single or All

Used where you can enter both a “From” and a “To” selection.



Suppliers: All
From:
To:
Effective Date: 31/03/2005
Include Credits, Debits, Held or All?: All
Transactions in Entry or Account Currency? (E/A): A

Label

This should state the title of the data item whose range is being prompted for.

- Use the plural form of the item being selected.

Range Combo box

Allows a selection from “All”, “Single” or “Range”.

- Default selection to All.
- Allow two spaces between the corresponding label and the combo field.
- Allow five spaces between combo field and first range selection field.

From

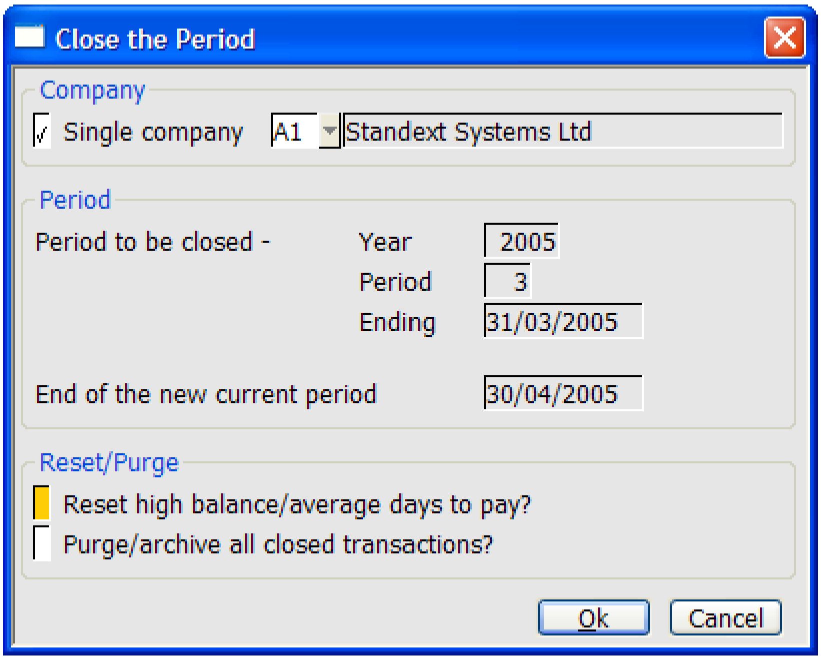
- Blank indicates first on file.
- Field is blanked and skipped if range is “All”.
- Field may not be left blank if range is “Single”.
- Entering a partial or invalid code sets the field to the first valid code after the invalid entry.

To

- Only prompted for when range is “Range”.
- Field should be blanked and skipped if range is “All”.
- If range is “Single” this field is set to the “From” code.
- Blank indicates last on file.

Single or All

Used where you can either enter a single code or leave the field blank to indicate all items.



Company

☑ Single company A1 Standext Systems Ltd

Period

Period to be closed - Year 2005

Period 3

Ending 31/03/2005

End of the new current period 30/04/2005

Reset/Purge

□ Reset high balance/average days to pay?

□ Purge/archive all closed transactions?

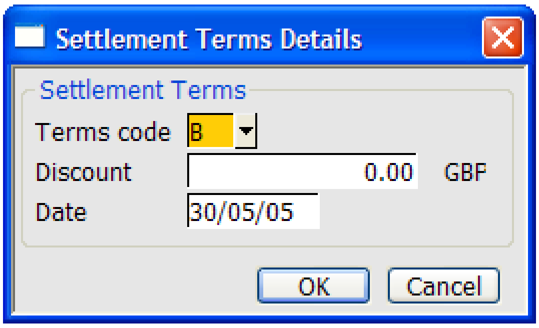
Tick box

Ticking the box allows you to enter the code of the single item required, Leaving the box un-ticked indicates that all items are required.

- The label should reflect that you are selecting a single item, eg “Single company”, “Single location” etc.
- The field used to display/enter the selected code should be to the right of the tick box label.
- If there is only one item of this type then tick the option, default the code to the one on file and suppress input of the field(s).

Dialogue windows

The most common type of window is known as an EDT window. It displays and accepts a series of prompts within one or more group boxes.



Terms code B

Discount 0.00 GBP

Date 30/05/05

- Stand alone windows should use OK and Cancel navigation buttons.

- Windows that are the first window in a series should use Next and Cancel navigation buttons.

- Windows that are in the middle of a series of connected windows should use Back, Next and Cancel navigation buttons.

- Windows that are the last window in a series should use Back, Ok and Cancel navigation buttons.

There are a number of specific types of EDT window that are described in more detail further on, these include.

- Report selection criteria and options windows.

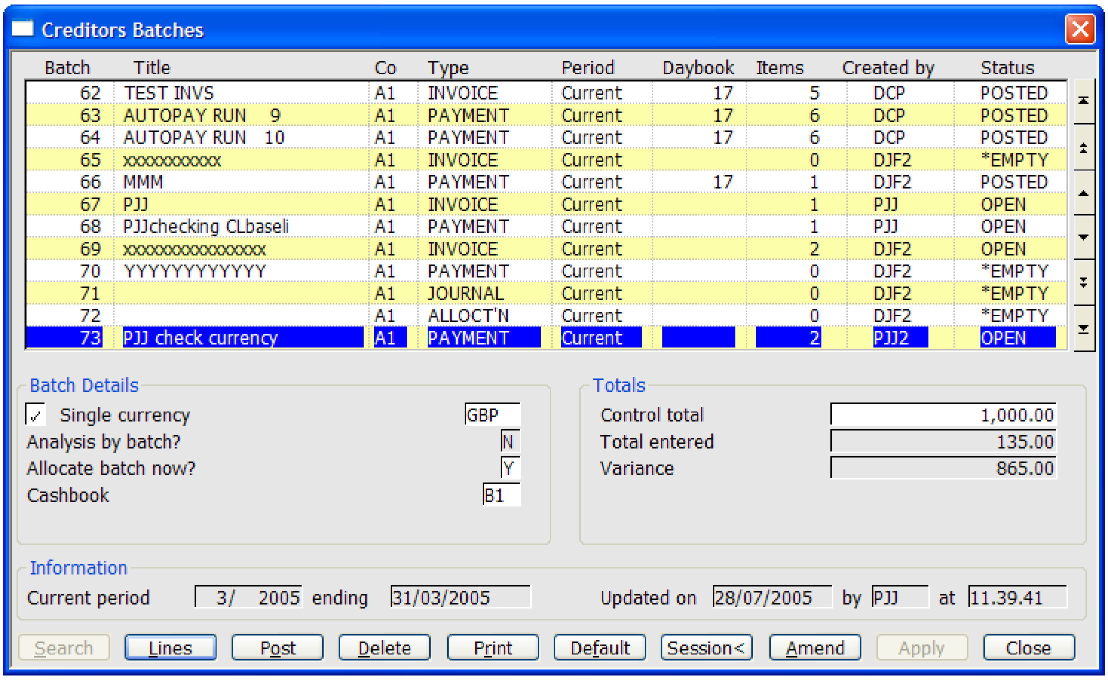
- System parameter windows.

Application Specific Guidelines

The following items describe techniques and layouts that should be used on all applications of the same type, not just those represented by the example screen shots.

Transaction Entry

List of batches window



Creditors Batches

Batch	Title	Co	Type	Period	Daybook	Items	Created by	Status
62	TEST INVS	A1	INVOICE	Current	17	5	DCP	POSTED
63	AUTOPAY RUN 9	A1	PAYMENT	Current	17	6	DCP	POSTED
64	AUTOPAY RUN 10	A1	PAYMENT	Current	17	6	DCP	POSTED
65	xxxxxxxxxxx	A1	INVOICE	Current		0	DJF2	*EMPTY
66	MMM	A1	PAYMENT	Current	17	1	DJF2	POSTED
67	PJJ	A1	INVOICE	Current		1	PJJ	OPEN
68	PJJchecking CLbaseli	A1	PAYMENT	Current		1	PJJ	OPEN
69	xxxxxxxxxxxxxxxx	A1	INVOICE	Current		2	DJF2	OPEN
70	YYYYYYYYYYYY	A1	PAYMENT	Current		0	DJF2	*EMPTY
71		A1	JOURNAL	Current		0	DJF2	*EMPTY
72		A1	ALLOCT'N	Current		0	DJF2	*EMPTY
73	PJJ check currency	A1	PAYMENT	Current		2	PJJ2	OPEN

Batch Details

☑ Single currency GBP

Analysis by batch? N

Allocate batch now? Y

Cashbook B1

Totals

Control total 1,000.00

Total entered 135.00

Variance 865.00

Information

Current period 3/ 2005 ending 31/03/2005 Updated on 28/07/2005 by PJJ at 11.39.41

Group Boxes

Definition

- Batch Details

  This group box contains all the batch specific items that are neither a batch total nor static information relating to the batch.

- Totals

  Contains any value that could be classed as a total relating to the items within the batch, e.g. batch totals, number of transactions, variance etc.

- Information

  Contains static information pertaining to the batch. Last updated, current period end dates, operator details etc.

Layout

Batch details and Totals window are displayed side by side using half the total width of the window each. The information group is to be displayed below these two group boxes and extends the entire width of the window.

Buttons

Where space permits, buttons should be declared as a single line of buttons in the following sequence.

- Search

  Standard UF1 search button.

- Lines

  Takes the user to the transaction details window. If this is a closed batch, the transaction window will be presented in enquiry only mode.

- Post

  Standard post batch processing invoked.

- Delete

  Standard delete batch processing invoked.

- Print

  Any print functions are accessed via this option. If more than one type of print is available (i.e. CM110 has Print Batch and Print Cheque) then display a radio button menu to select the type of print required.

- Amend

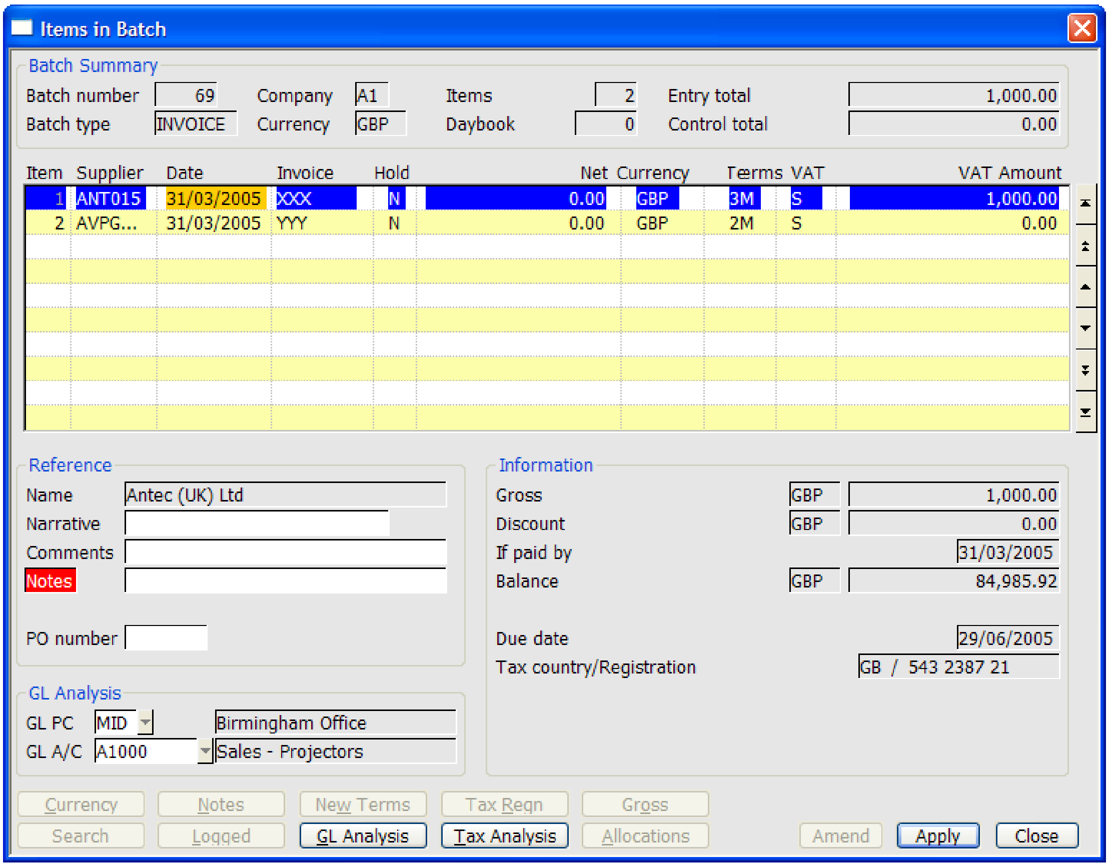
  This options allows the operator to amend the batch header fields displayed within the current window. Accepting the last field updates the batch details and remains in the current window with the same record highlighted.

  It does not automatically drop you into the items in batch window.

- Apply / Close

  Standard processing.

Items in batch window



Items in Batch

Batch Summary

Batch number	69	Company	A1	Items	2	Entry total	1,000.00
Batch type	INVOICE	Currency	GBP	Daybook	0	Control total	0.00

Item	Supplier	Date	Invoice	Hold	Net	Currency	Terms	VAT	VAT Amount
1	ANT015	31/03/2005	XXX	N	0.00	GBP	3M	S	1,000.00
2	AVPG...	31/03/2005	YYY	N	0.00	GBP	2M	S	0.00

Reference

Name	Antec (UK) Ltd
Narrative	
Comments	
Notes	
PO number	

Information

Gross	GBP	1,000.00
Discount	GBP	0.00
If paid by		31/03/2005
Balance	GBP	84,985.92
Due date		29/06/2005
Tax country/Registration		GB / 543 2387 21

GL Analysis

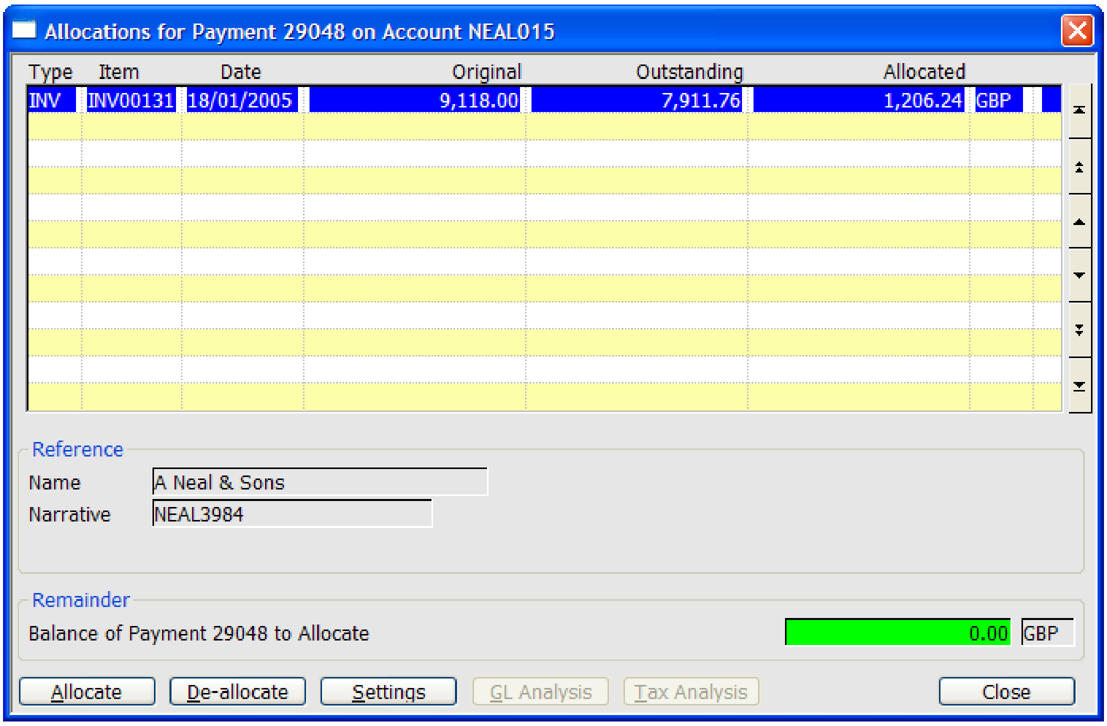
GL PC	MID	Birmingham Office
GL A/C	A1000	Sales - Projectors

- Batch header summary should appear as a group box at the top of the items in batch window, rather than as a separate window that is artificially positioned above the items in batch window.

- Subsequent windows need to be accessible via a button from this window when amending, eg GL Analysis, Tax Analysis.

Remaining balance window

Typical examples are the GL Analysis, Tax analysis and Allocation windows.



Allocations for Payment 29048 on Account NEAL015

Type	Item	Date	Original	Outstanding	Allocated	
INV	INV00131	18/01/2005	9,118.00	7,911.76	1,206.24	GBP

Reference

Name: A Neal & Sons

Narrative: NEAL3984

Remainder

Balance of Payment 29048 to Allocate: 0.00 GBP

Remaining Balance Display

- Balance remaining must be displayed at bottom of window normally right aligned within a group box titled "Remainder". If possible, align the field with the field being totalled from the scrolled area.

- You will need to use the GXLFI$ sub-routine to ensure that any remainder fields remain displayed even when you step onto an empty line

- Whilst the remainder is non-zero, this value should be displayed using the error attribute (B$ERR). Once the remainder reaches 0, display using the warning attribute instead (B$WARN). This will have the effect (with default GX settings) of the values changing from red whilst a balance remains, to green once it reaches 0.

Buttons

In addition to any standard navigation and application buttons, the following are also required.

- Reset

  This option removes all analysis/allocations present and starts the process again.

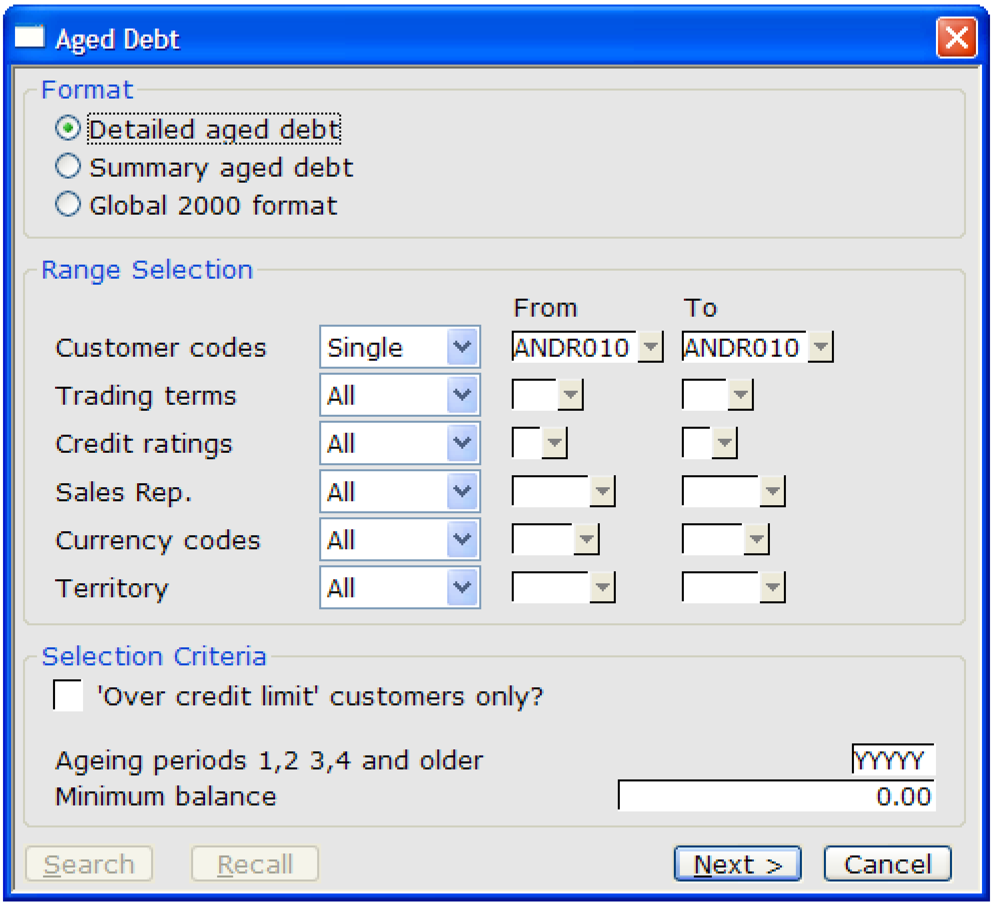
Reporting

Reports usually consist of a series of windows that prompt for the ranges of items to include, the sequence in which to print them and the type of information to be printed.

- All windows use “Next” and “Back” to navigate between windows.
- The “Cancel” button aborts the report and returns to the application menu.

Windows (where used) should appear in the following sequence.

Selection criteria / Report format



Aged Debt

Format

● Detailed aged debt
○ Summary aged debt
○ Global 2000 format

Range Selection

		From	To
Customer codes	Single	ANDR010	ANDR010
Trading terms	All		
Credit ratings	All		
Sales Rep.	All		
Currency codes	All		
Territory	All		

Selection Criteria

□ 'Over credit limit' customers only?

Ageing periods 1,2 3,4 and older YYYYY
Minimum balance 0.00

The initial window(s) prompt the user for range selection and report content. Standard group boxes could include the following.

- Format

  Contains a radio button menu allowing the user to select from different formats or styles of the same report.

- Range selection

  Contains all the range selection prompts that determine the set of records selected for the report.

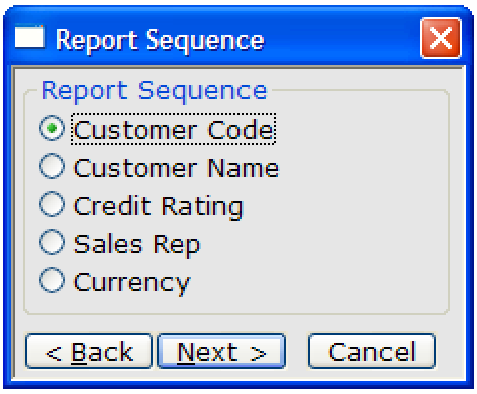
- Selection criteria

  Additional items that can be specified to further restrict the items selected for the report.

- Report Content

  Used to specify options that affect the detail and layout of items printed on the report.

Report Sequence



Report Sequence

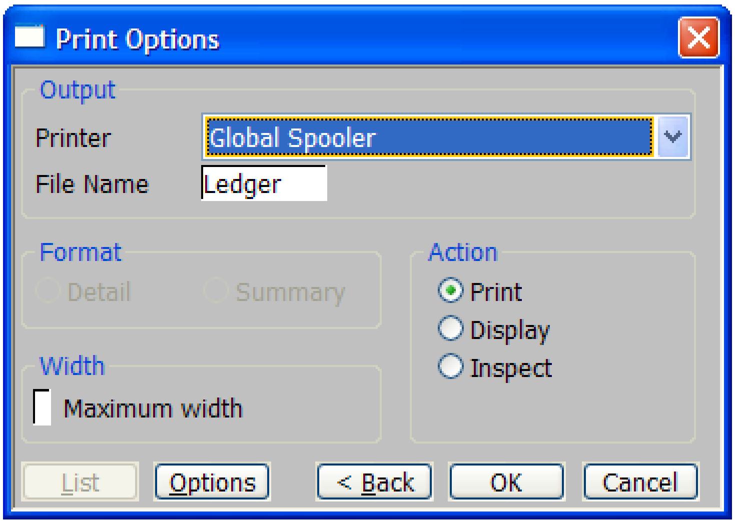
Report Sequence

● Customer Code
○ Customer Name
○ Credit Rating
○ Sales Rep
○ Currency

Used to select the sequence in which a report can be printed.

Print Options window

All reports must end with the print options dialogue window



Print Options

Output

Printer Global Spooler

File Name Ledger

Format

○ Detail ○ Summary

Width

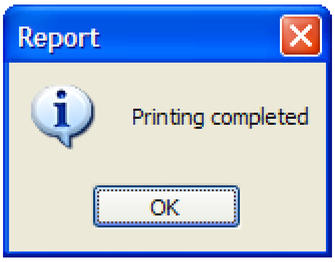
□ Maximum width

Action

● Print
○ Display
○ Inspect

For details of the programming interface please see document gx15722.doc

Report Completed

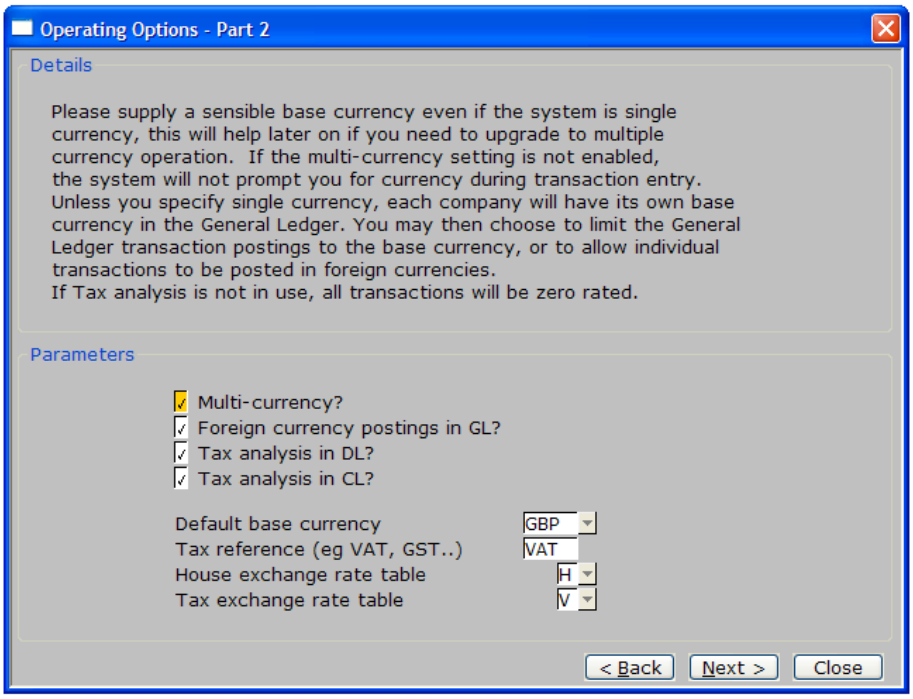


Printing completed

After printing, if a “Report Completed” message is required, then use `CALL GX%PRC` to initiate the above dialogue.

Parameter Entry

Window



Operating Options - Part 2

Details

Please supply a sensible base currency even if the system is single currency, this will help later on if you need to upgrade to multiple currency operation. If the multi-currency setting is not enabled, the system will not prompt you for currency during transaction entry.
Unless you specify single currency, each company will have its own base currency in the General Ledger. You may then choose to limit the General Ledger transaction postings to the base currency, or to allow individual transactions to be posted in foreign currencies.
If Tax analysis is not in use, all transactions will be zero rated.

Parameters

☑ Multi-currency?
☑ Foreign currency postings in GL?
☑ Tax analysis in DL?
☑ Tax analysis in CL?

Default base currency: GBP
Tax reference (eg VAT, GST..): VAT
House exchange rate table: H
Tax exchange rate table: V

Group Boxes

- Details

  Contains the text that describes the parameters maintained by this window. Text should start in Column five and a blank line should appear within the group box before and after the text.

  If you find that when adding new parameters to an existing window, you need to add entire new paragraphs, then it may be that the new parameter should be in a separate parameter window.

- Parameters

  Contains the parameter labels and data items to be maintained.

Buttons

The following navigation buttons should be used where appropriate. In addition some parameter windows have additional application buttons .

- Back

  Return to the parameter window immediately prior to this one in the chain.

- Next

  Accept all values shown and display the next parameter window in the chain.

- Close

Accept all values shown, write them back and terminate the chain of parameter windows, returning to the high level menu from which they were selected.

Appendix A – Glossary Of Terms

Application Button A button that invokes an application function specific to the current window or data item being processed.

e.g. Find, View, Allocations

Combo Box A standard windows control that presents a list of values from which one value must be selected.

Group Box A standard windows construct that is used to enclose a series of related fields within a titled frame.

Label A text string that identifies an associated item of data.

MBOX$ A routine that allows us to generate standard windows dialogue windows for display messages and asking simple questions.

Navigation Button A button that is used to navigate between windows or to initiate/confirm processing of a record.

e.g. Amend, Apply & Close

Radio Button Menu A list of options displayed as a set of standard windows radio buttons.

Appendix B – Reference Documents

The following technical notes can be found in \gsm documentation\devsystem on the Reseller CD.

acmbo$.doc	Add string to GX combo box.
ccmbo$.doc	Clear combo box.
dcmbo$.doc	Display combo box.
gxboxes.doc	The BOX clause and GX group boxes.
gxlfi$.doc	Set/Reset Linked Field Index.
gxmsg$.doc	Display simple display only GX Message Box.
gxradiobuttons.doc	GX Radio buttons.
gxsbt$.doc	Set Button Text on Current GX Window
mbox$.doc	Display simple message dialogue box.

Appendix C – Programming Details

Buttons

A button is defined within the window structure itself and in its simplest form, looks very much like an ordinary label.

```
li cl Button-Label   BTN Unn
```

Where *li* is the Line number where the button is to appear, *cl* is the column number where the button is to appear, *Button-Label* is a text literal or a text variable to be displayed on top of the button, BTN is a clause indicating the segment is a button, and U*nn* indicates the function number the button is to generate, where *nn* is in the ranges 1-21, and 50-99 inclusive.

The function number is the value returned in system variable $FUNC when the button is clicked. Each button should therefore have a separate function number to allow the button click to be identified. Numbers in the range 1-21 are Standard System Functions such as <DEL> ($FUNC = 13), and <INS> ($FUNC = 17). Functions in the range 50-99 are Applications Function, which allow you to define up to 50 further functions for any given window.

Window Manager automatically manages Buttons for System Functions (U01-U21). If, for example, you code an explicit button for Delete (using the BTN U13 Clause>, then the user will be able to delete a record by clicking on your button, by using the <DEL> key (Usually mapped to F7 on the keyboard), or by clicking on the Toolbar Delete Button. As Window Manager already knows how to process the <DEL> function, you do not need to write any other code. Note however that you cannot use the ENABLE or DISABLE verbs to enable or disable system functions, as these at all times operate directly under Window Manager's control.

Taking the F1 search function as our example again, if you add a button (titled Search) that maps to U01 then GX will automatically enable it on fields with the UF1 attribute, and disable it when you move off of that field.

When you create a button for an Application function (U50-U99), then you will need to write some code in the new R-FUNC entry-point to process your function (See later in this document). You will probably also want to use the new R-START entry-point (Also see later in this document) to enable or disable the buttons depending on the state of processing. When an Application Button is clicked, Window Manager simply calls the R-FUNC routine within the routines section to give the programmer the opportunity to deal with the operation. If no R-FUNC routine exists, or no code has been written to deal with the function, then no action occurs.

Enabling and disabling

When a window is first displayed, all buttons are enabled by default. An application button in the range U50-U99 may be disabled or enabled using the ENABLE and DISABLE verbs. A disabled button is shown grayed out, and cannot be clicked on.

The syntax of the verbs is as follows:

```
ENABLE [ WINDOW Window-ID ] BUTTON nn [TEMPORARY | DEFAULT]

DISABLE [ WINDOW Window-ID ] BUTTON nn [TEMPORARY]
```

Where Window-ID identifies the affected window, and nn is a numeric literal or a 9(4) Comp integer in the range 50-99 inclusive, giving the button's Function number to be enabled or disabled. Window ID may be omitted when the instruction is coded within a Window's Routines-Section.

The TEMPORARY clause causes the button to be enabled or disabled for the current accept operation only, after which the button returns to it's preceding state. Such a temporary change can only be made within a fields B-Routine, and only persists for the duration of the Accept Operation for that field. When TEMPORARY is omitted, the buttons state is altered for as long as the window remains displayed (or until a further ENABLE or DISABLE again alters its state).

The DEFAULT clause, which is valid only for the ENABLE statement, causes the specified button to be enabled and set as the default button. The default button is displayed with a deepened black surround, and is automatically selected when the <ENTER> key is typed by the operator. Note that this feature requires the legacyTabCRProcessing to be disabled.

Processing buttons

R-START

The R-START entry is called whenever Window Manager begins processing for a given operating mode ($MODE). The entry-point has been provided to allow you to Enable or Disable buttons appropriate for the current mode using the ENABLE and DISABLE statements. Your code should simply test $MODE, and then enable the buttons you want enabled, and disable the buttons that are invalid for the mode. In general, you should code all of your non-Temporary ENABLE and DISABLE statements in this routine.

The Entry-point is called immediately before the first accept operation takes place as processing begins in all modes. For symmetry and completeness, the R-START entry-point is called in Delete mode even though no accept operation normally takes following on a record deletion request.

Note that unlike the R-FETCH routine, the entry-point is NOT called each time a record is read. You cannot therefore use this entry point to set field display attributes, except for the record that is currently being processed.

You can use the R-START Routine even when using a mixture of both navigation and application Button functions. Immediately before calling the R-START routine, Window Manager will already have enabled the appropriate navigation buttons, and will have set the default navigation button according to its rules. You can then use the ENABLE .. DEFAULT statement to over-ride the standard Window Manager default button, which then remains in force until R-START is next called.

R-FUNC

The new R-FUNC Routines Section Entry-point allows ALL functions to be intercepted and processed by the application program. It is used to process application buttons, and to augment or over-ride the processing of standard navigation button functions.

The R-FUNC routine is called by Window Manager whenever any navigation or application function is requested. The routine is called irrespective of the source of the request and returns all functions generated by all Key-stroke, Button-click, Toolbar, TYP$ and Mouse operations.

The requested function number is returned in System Variable $FUNC. Thus when the button coded as BTN U78 is clicked, $FUNC will contain the value 78, or following a <PGE> keystroke, the value 5. Window Manager supplies the Symbol name of the Variable being accepted when the function was requested in Variable $VARNAME. Thus if the button was clicked while on field TRNAME, system variable $VARNAME will contain the value "TRNAME".

This can be extremely useful when the action to be taken depends on the field being processed. For example, you may want to define a "Search" button, but what is searched may depend on the field being processed. Note that when the R-FUNC routine is called in DSP mode, no particular field is in the process of being accepted, and $VARNAME is therefore null.

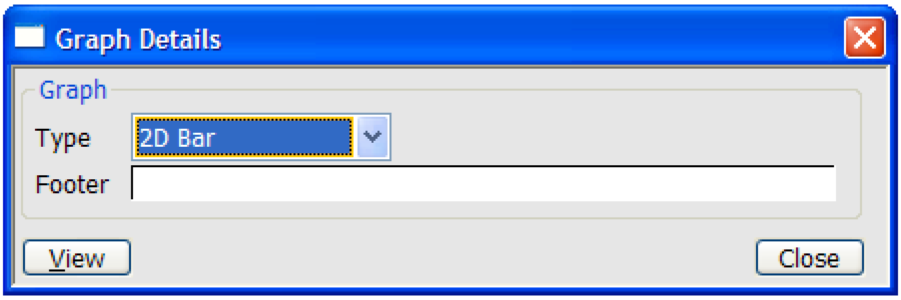
Having identified the requested function, the R-FUNC routine may immediately perform the requested action, which might involve invoking another window or some other process. Alternatively the routine may make use of the TYP$ interface to cause window manager to take some action, such as deleting a series of records, or forcing an exit from the window.

An EXIT WITH 1 condition returned by the R-FUNC routine will cause Window Manager to ignore the function, and resume processing at the current field. This allows you to suppress an inappropriate function at any time. An EXIT condition > 1 causes the Window to be terminated immediately, with the Exit code passed back to the calling ENTER WINDOW statement.

It should be noted that the processing to be performed for an application button is entirely in the hands of the application programmer. Window Manager simply passes the event to the R-FUNC routine, after which it simply resumes the last accept operation.

Example

The following example is taken from the current Global 3000 V6.0 Stock Control.



Type 2D Bar

Footer

```
WINDOW W1
EDT
SBOX
SEQUENCE EXIT CLW, EXIT CLW
BOX 03 02 60 03 "Graph"
BASE AT  4 12
02 02  "Graph Details"                          A12
04 03  "Type"
                04 10  W1TYPX      X(15)
04 10  "~C106"                                  BTN U51
05 03  "Footer"
                05 10  W1FOOT      X(50)        NUL
07 02  "    View"                               BTN U50
07 54  "   close"                               BTN U11
```

One application button has been defined and one standard navigation button. The "Cancel" button will generate function 11 <BCK>.

The remainder of the code to process the buttons is as follows.

```
R-FUNC.
        IF $FUNC = 50
           MOVE W1TYPX TO M1OPT
           PERFORM DISPLAY-GRAPH
        END
EXIT
```

Group Boxes

The following describes how group boxes have been implemented in the speedbase language and the standards for their use in Global 3000.

Introduction

A group box is used to enclose related fields on a window. It's only purpose is to improve the presentation of the dialogue.

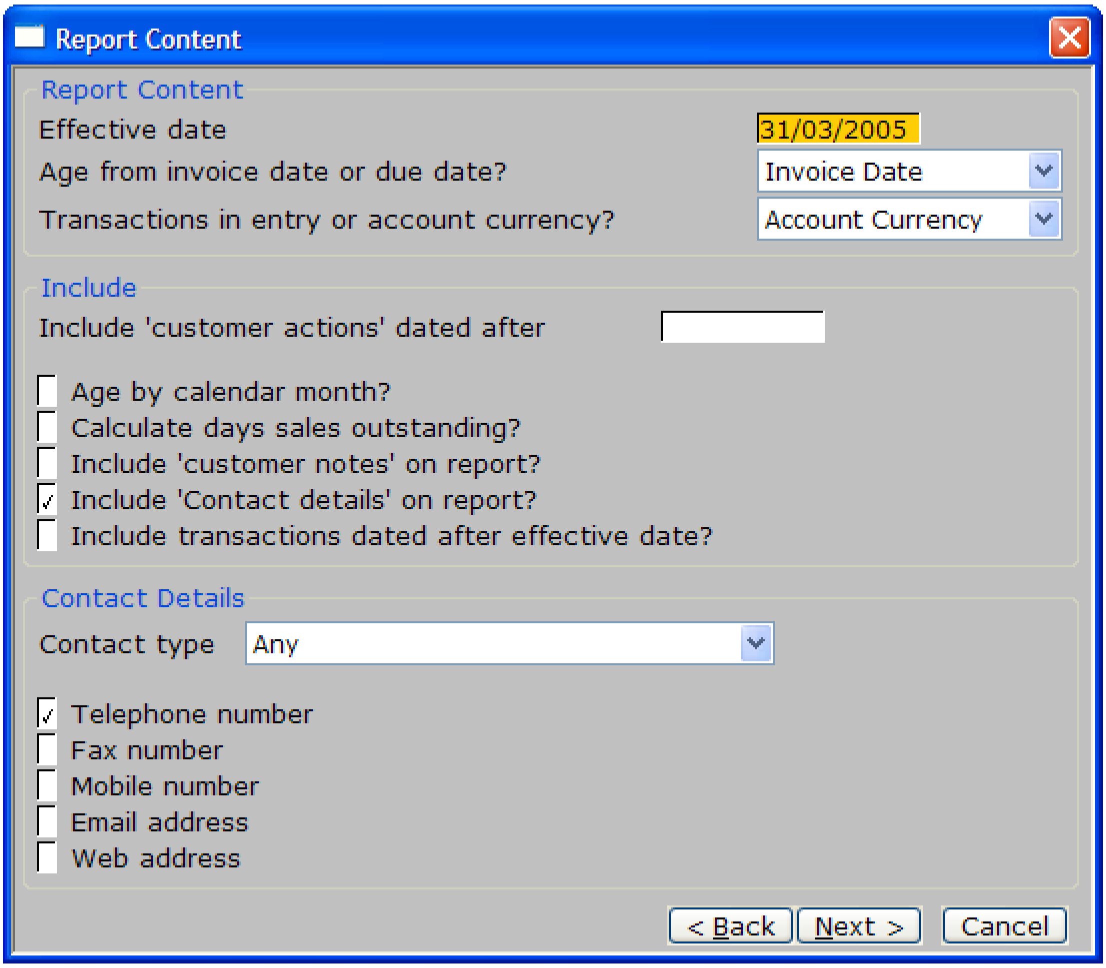
Defining a group box

A group box is defined using the new BOX statement in the window division. A window may contain any number of group boxes.

```
BOX row column width depth caption
```

Example

The following example is taken from the Debtors Aged Debt Report in V6.0



Effective date 31/03/2005

Age from invoice date or due date? Invoice Date

Transactions in entry or account currency? Account Currency

Include 'customer actions' dated after

☐ Age by calendar month?
☐ Calculate days sales outstanding?
☐ Include 'customer notes' on report?
☑ Include 'Contact details' on report?
☐ Include transactions dated after effective date?

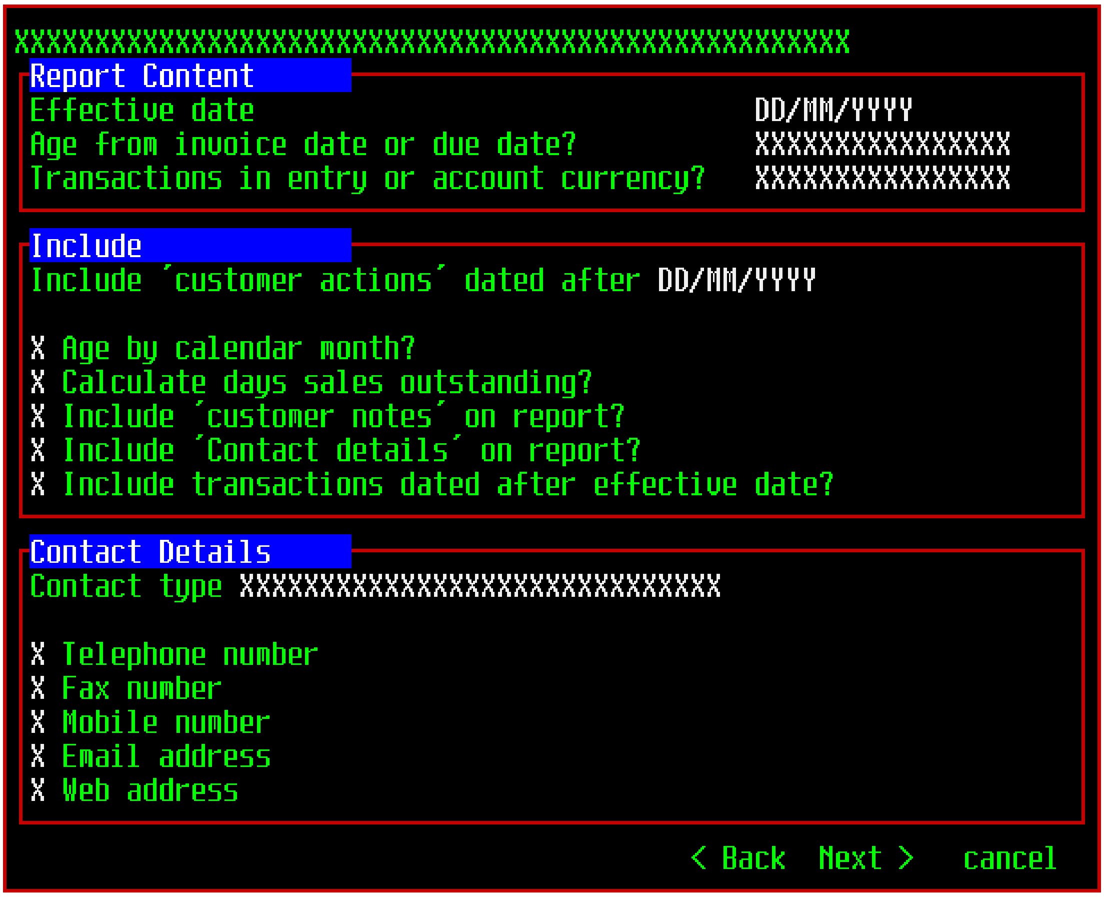
Contact type Any

☑ Telephone number
☐ Fax number
☐ Mobile number
☐ Email address
☐ Web address

WINDOW DIVISION

```
WINDOW W1
SBOX
DISABLE CLR
ENABLE ABO
EDT
POP-UP
BOX 03 02 66 04 "Report Content"
BOX 08 02 66 08 "Include"
BOX 17 02 66 08 "Contact Details"
BASE AT  5 15
```

The preview function in $SDE (i.e. F1 when in F9-Q mode) shows you the approximate position of the boxes to allow checking and fine adjustments.



```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 Report Content
 Effective date                              DD/MM/YYYY
 Age from invoice date or due date?          XXXXXXXXXXXXXXXX
 Transactions in entry or account currency?  XXXXXXXXXXXXXXXX

 Include
 Include ´customer actions´ dated after DD/MM/YYYY

 X Age by calendar month?
 X Calculate days sales outstanding?
 X Include ´customer notes´ on report?
 X Include ´Contact details´ on report?
 X Include transactions dated after effective date?

 Contact Details
 Contact type XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

 X Telephone number
 X Fax number
 X Mobile number
 X Email address
 X Web address

                                          < Back  Next >   cancel
```

Combo Boxes

The following describes how comb boxes have been implemented in the speedbase language and the standards for their use in Global 3000.

Introduction

A Combo box is used where the user needs to select one option from a limited list of options. This will typically replace some of the pop-menus that we use and also some of the search windows where you select from a limited number of records (Sales reps, country codes etc).

The technique where we prompt the user for an option via a one or two letter prompt and then display a long description of that option are easily replaced by this technique.

Defining a combo box

A combo box is a combination of an input field and a button. The input field is the field in which the selected option is passed back, and the button is converted to the drop down arrow within the combo box control. Clicking the button returns the assigned buttons function value (U50-U99) and is processed in the R-FUNC routine. You may define up to 9 combo boxes on the same window.

As well as defining the field and button within the window, there are three system calls that control the operation of the combo box.

```
CALL CCMBO$ USING number
```

Empty the list of values from the passed combo box

```
CALL ACMBO$ USING number display_text [return_text]
```

Add an entry to the combo box passed. If the optional third parameter is specified, this return_text is passed back to the input field in place of the display_text if that option is selected.

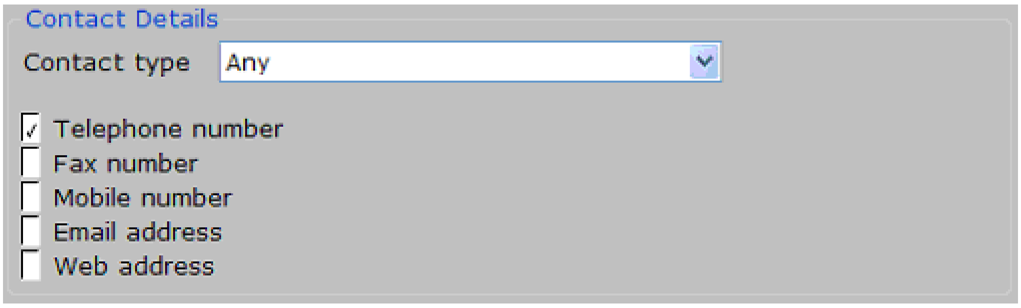
```
CALL DCMBO$ USING number
```

Always called from the R-FUNC routine in response to the button being pressed. This routine triggers the display of the combo box. GX then handles the user selection and the selected option is passed back to the input field whose V- routine is then invoked.

Example

The following example is taken from the Debtors Aged Debt Report in V5.1, and replaces the input of a one character "code" that represented the required contact type.

Aged Debt Window



Contact Details

Contact type: Any

☑ Telephone number
□ Fax number
□ Mobile number
□ Email address
□ Web address

```
77 W1CTYP       PIC X      * Selected Contact Type
01 X-TEXT
   03 C-TEXT    PIC X(30)  * low-values terminated display_text
   03 FILLER    PIC X
                VALUE #00
01 X-CODE
   03 C-CODE    PIC X      * low-values terminated return_text
   03 FILLER    PIC X
                VALUE #00
WINDOW DIVISION

18 03 "Contact Type"
                        18 16 W1CTYP X(30)
18 16 "~C108" BTN U50
```

The field W1CTYP is declared as an X(30) in the window as that is the maximum size of the values that will be displayed in the combo list. To the rest of the speedbase program, the field is still the original 1 byte input field and the existing code should not need amending.

The button U50 must be positioned at the same position as the input field and must have the caption in the shown format. The positioning is important as this is how GX associates the combo box with it's input field. The ~C tells GX that this is a combo box, 1 is the combo box number and 08 defines that the drop down list will be 8 entries deep (windows adds scroll bars if there are more than eight entries).

Within the B-W1CTYP

```
IF W1C1LO ZERO
   MOVE 1 TO W1C1LO          * Only build list once
   CALL CCMBO$ USING 1       * Clear down list of entries for combo 1
   MOVE "N" TO K-CTYP        * Contact types are held on the
   FETCH FIRST SDCDEX KEY K-CTYP NOLOCK * SD record set
   ON NO EXCEPTION
      DO
         MOVE SDTEXT TO C-TEXT      * Move text to the display field
         MOVE SDCODE TO C-CODE      * The code we want to pass back
         CALL ACMBO$ USING 1 X-TEXT X-CODE   * Add it to the list
         FETCH NEXT SDCDEX KEY K-CTYP NOLOCK
         ON EXCEPTION FINISH
      ENDDO
   END
END


R-FUNC.
      IF $FUNC = 50           * User has pressed the combo button
         CALL DCMBO$ USING 1  * display comb box 1
         IGNORE EXCEPTION
      END
EXIT
```

After a value is selected the V- routine is processed just as if W1CTYP had been keyed in as a standard input field.

It is important that we only build the list once, as clearing the list causes it to be blanked out on screen. The list remains valid whilst the window is on screen, even if it is not the current active window. Once the window is cleared, the list will need to be rebuilt if the window is subsequently re-displayed.

If W1CTYP contains a valid entry from the combo list when the window is first displayed, GX will display the corresponding display text as the default value.

Radio Buttons

Introduction

Radio buttons can be used when there is a sequence of options of which only one can be selected. They are used to replace pop-menus used in earlier versions of Global 3000.

Defining Radio Buttons

A radio menu is simply a series of specially formatted text labels.

"~R1CCustomer Code"

~R Specifies that this button group has the label to the right of the radio button. If you use ~L, the labels are displayed to the left of the button. The standard in Global 3000 is to always use ~R.

1 Identifies this as button group 1. All items to be treated as part of this button group must be declared with the same group number.

C Is the character returned in the input field if this option is the one selected.

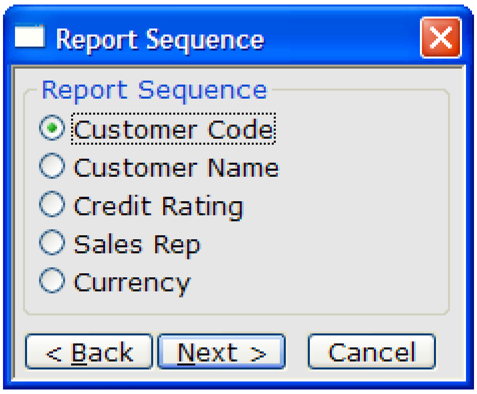
Customer code This is the text of the label.

Radio button options can be "greyed-out" in the same way that a pop-menu line can. Unlike a pop-menu, you cannot select the greyed out option to receive an error message. To grey out an option, replace the character to be returned with an "*".

e.g. "~R1*Customer Code"

Example

```
WINDOW M1
SEQUENCE EXIT CLW, EXIT
POP-UP
SBOX
EDT
ENABLE ABO
BOX 03 02 26 06 "Report Sequence"
BASE AT 20 58
02 02  "Report Sequence"
A12
04 03  "~R1CCustomer Code"
05 03  "~R1NCustomer Name"
06 03  "~R1RCredit Rating"
07 03  "~R1SSales Rep"
08 03  "~R1YCurrency"
                04 03  Z-MOPT  X
10 02  "  < Back"            BTN U11
10 10  "  Next >"            BTN U20
10 19  "  cancel"            BTN U10
ENDWINDOW
```



The input field (Z-MOPT), must be position at the same co-ordinates as any of the labels within the group.