

What's in Ffenics v 1.23 Quickfix

Introduction

This fix is mainly concerned with Memo and Long Text fields and a variety of obscure gpf's (particularly involving Subforms on Tabs).

About Memo and Long Text fields.

Some people have expressed puzzlement about why we have both memo and long text fields, so here is a short explanation.

In most computer database systems, since the vast majority of fields are quite short, they are optimised for dealing with these short fields. The most common maximum size for the largest 'normal' field is 254 characters long, 255 being a convenient number internally. Numbers below 255 can be processed much faster than larger numbers.

Ffenics shares this characteristic, so we need special field types for 'long' data, and these tend to have restrictions and features different to 'normal' fields.

Ffenics Memo fields and Ffenics Long Text fields both handle text fields greater than 255 characters in length but have quite different purposes, even though both appear in record entry as a multiline edit control.

Memo fields are not capable of being processed. You can type them in, edit them, do text searching on them, display them and print them, but you cannot do anything useful with the contents from any kind of script. Memo fields are truly variable length - up to 64k - and stored in supplemental .CLO (Character Large Object) files rather than integral to the form.

Long Text fields, while offering the same facilities as memo fields, can additionally be processed by scripts. They are limited to 4k. Long Text fields are stored in fixed length like all other fields (except images, which are also 'long' and are stored 'externally' somewhat like Memos.).

Each long text 'field' will actually be stored as one or more 255 char fields and an additional odd length field to make up the length you specify in the field definition. You can see these supplemental fields in the pick lists in the script editor - it is only when writing scripts you actually need to know about them.

So, Memo fields are literally that – they are meant for comments etc. that are not to be regarded as true process-able data. Long text fields are meant for textual data that you need to manipulate, e.g. in order to create text intensive output like contracts.

In addition, having the two types gives us better options when mapping to external data sources, which also frequently have multiple types of “over 255 character fields”, equally with a variety of restrictions and abilities.

Changes

Memo and Long Text Fields

- Long Text was refusing to display more than 255 characters after a save (they were there, just not being shown on screen – but it meant they could be lost if the record was saved or modified again).
- Memo fields were losing data on a reorganisation or a restore, or if the form definition was changed enough to require reformatting the data file.
- Memo fields no longer lose internal chunks of data when edited.
- .CLO file handling revised to be more robust.
- Long Text fields have a default initial length of 255 in field definition.

Input Using:

- Not a fix, a reminder – if you are using the data from an input using document as ‘record entry’ you should make sure that if the user types ‘esc’ to escape from the IU that you do not input the data on the IU form as a record because it will may not have been validated.

Exception Handler:

- The Exception Handler has been enhanced to provide more useful information, specifically last Ffenics action executed and last Windows message received. These codes will help considerably in tracing any problem that arises.
- The Exception Handler now performs retries on ‘recoverable’ exceptions.
- It also now recognises newer versions of Windows. (2003, Vista, 7)

Other Fixes:

- You can no longer close a document from an Execute ELF or script using the DocumentClose() action function if the document is still in the process of saving data (caused an intermittent gpf). In practice this only affects the case where the user has tried to save data with errors. The result will now be that the document will return so that the errors can be amended and the user will have to attempt the save again.
- We no longer update the wrong set of document flags when returning from a script that closed a document. This caused another intermittent gpf depending on what flags were set – could also prompt to save changes when no change made, etc. etc.
- A complex set of circumstances caused a gpf in Sum Of.

The fix is not a full cure – it means that in some cases (where an Aspect has a subform on a tab, and a virtual field on the underlying form is defined which does a sum of using the same relationship as the subform, BUT the field being summed is not on the subform) the sum of will not sum. But it won't cause an (apparently) random gpf either.

There are two workarounds – define a second relationship for the subform with the same fields but a different unique name, or make sure the field being summed is included in the subform. (It can be hidden.)

- Fixed another 'apparently random' gpf when changing focus on a subform on a tab – the 'change focus' routine was incorrectly identifying the subform's parent. As with the other intermittent gpf's, these were involving 'pointers to pointers' - as long as the place being pointed to was 'clean' (NULL) then no problem occurred, but if there was data in the 'wrong' place being pointed to a gpf would happen.
- Scroll bars on subforms on tabs should be displayed properly and no longer 'bleed through' to the wrong tab.
- Fixed memory leak when using a selection criteria in a MultiBox (i.e. using relationship as opposed to direct to the form).
- Field definition now correctly defaults to 7 digits for numbers and 255 for Long Text (see above). This has required the default length for an ordinary text field to be increased to 16 characters.
- User help information on the status bar was unnecessarily being cleared or overwritten due to confusion over the correct buffer to use.

Release Notes:

The following known problems remain:

- The cutting and pasting of image fields is broken.
- The concatenate option in Document Print Options for Converter output remains unavailable.
- The Relationships form needs adjustment to fit better on an 800x600 screen.
- The parameter to Help Search doesn't work – see MS article KB241381
- The default delimiter is not set correctly for CSV type on import/export screen (when first in).
- The 2000 table limit in OLEDB / ODBC remains. The process for creating a native form from a remote table needs to be simplified.
- Re-layout – there is still work in progress to improve the decision making in this complex process – please report anything you feel is an anomaly and we will continue to refine this feature.
- Convert to DQL does not always produce DQL that conforms to best-practice syntax rules – in rare cases this results in scripts that need editing before they produce correct results.