Vitalware®
Vital Records Management

The World's Premier Vital Records Software

Vitalware Release Notes

Vitalware 2.4

Document Version 1

Contents

Here you will find collected together the Release Notes for Vitalware 2.4, alongside all documents referenced in the notes. These release notes and documents are also available on the Vitalware website.

This PDF document brings together a number of individually published documents: please note that page numbering below refers to this combined PDF document and not to the page numbers printed at the bottom of pages, as each individual document has its own internal numbering:

Release Notes: Vitalware 2.4
Release Date: 08 October 2013

Requirements

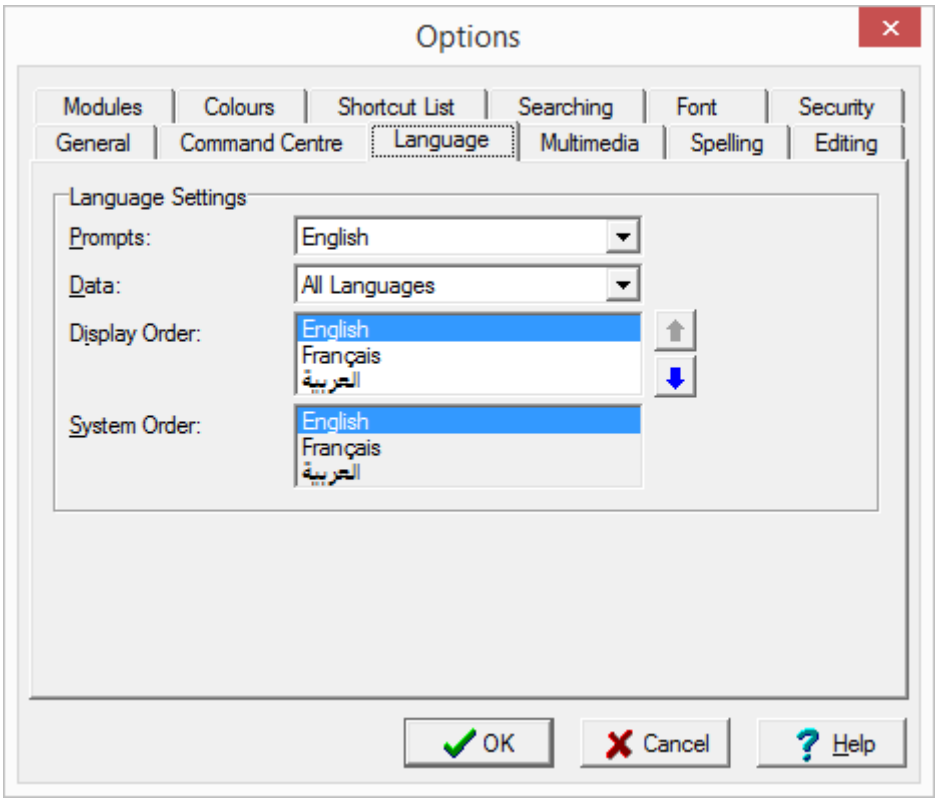
- For Windows 2000, XP, 2003, Vista, Windows 7, Windows 8
- Texpress 8.3.012 or later
- TexAPI 6.0.011 or later
- Perl 5.8.8 or later

New Features

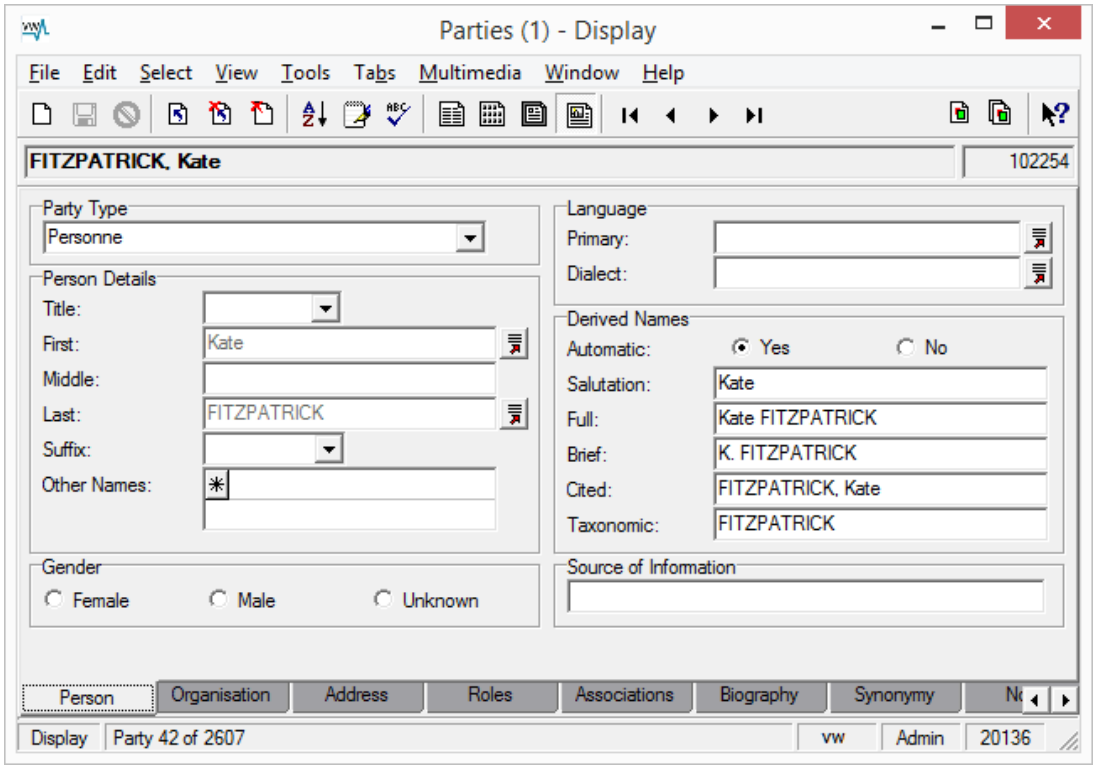
Edit in a Single Language

Edit in a single language functionality allows users with a multi-lingual version of Vitalware to use the system in either a single language or all languages. Full functionality is provided regardless of the language(s) selected. A number of extensions have been added that provide a more flexible mechanism for displaying and altering data within a single language:

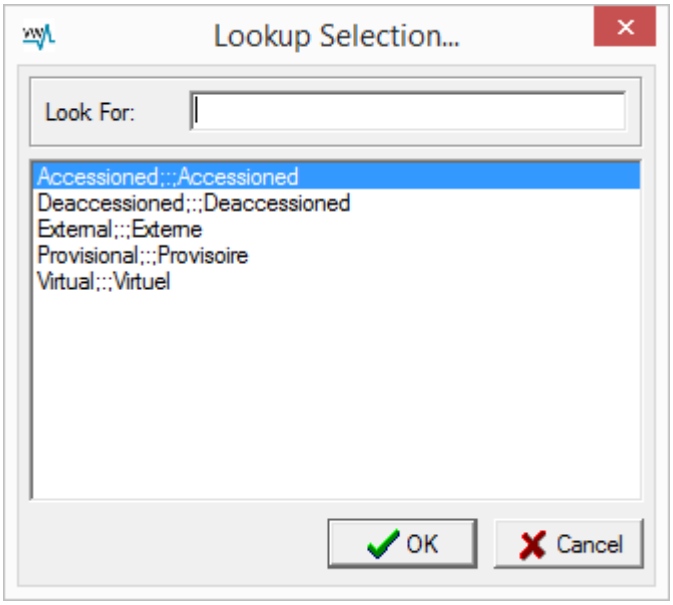
- When the *All Languages* option is selected, users may now select the order in which languages are displayed. This allows users who are dominant in a particular language to display this language first with other languages following:



- Users may now alter data while a single language is displayed. Any data entered into a field will only update the value of the language selected.
- When displaying data in a single language and where a value does not exist in that language for a given field, the value of the first filled language may be displayed. The text is shown in a user selectable colour (grey by default) to indicate that the value is not associated with the current language. When the field is entered, the text is removed, allowing data to be input:



- A number of data entry helpers always display values in all languages in the user defined order to assist data entry. For example, if a Lookup List is displayed, all languages are always shown so that users may select the correct entry based on context. This is particularly important where one term in a given language has multiple terms in another language:

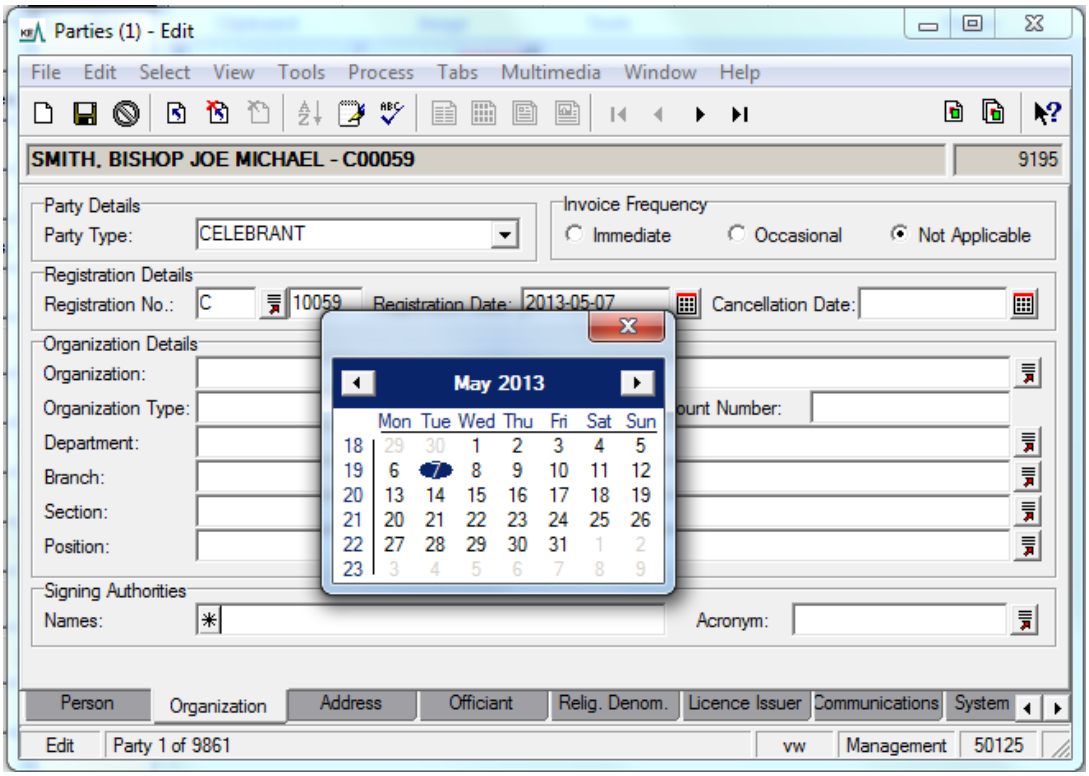


A complete description of the support for editing in a single language can be found in the How to edit in a single language documentation.

Improvements

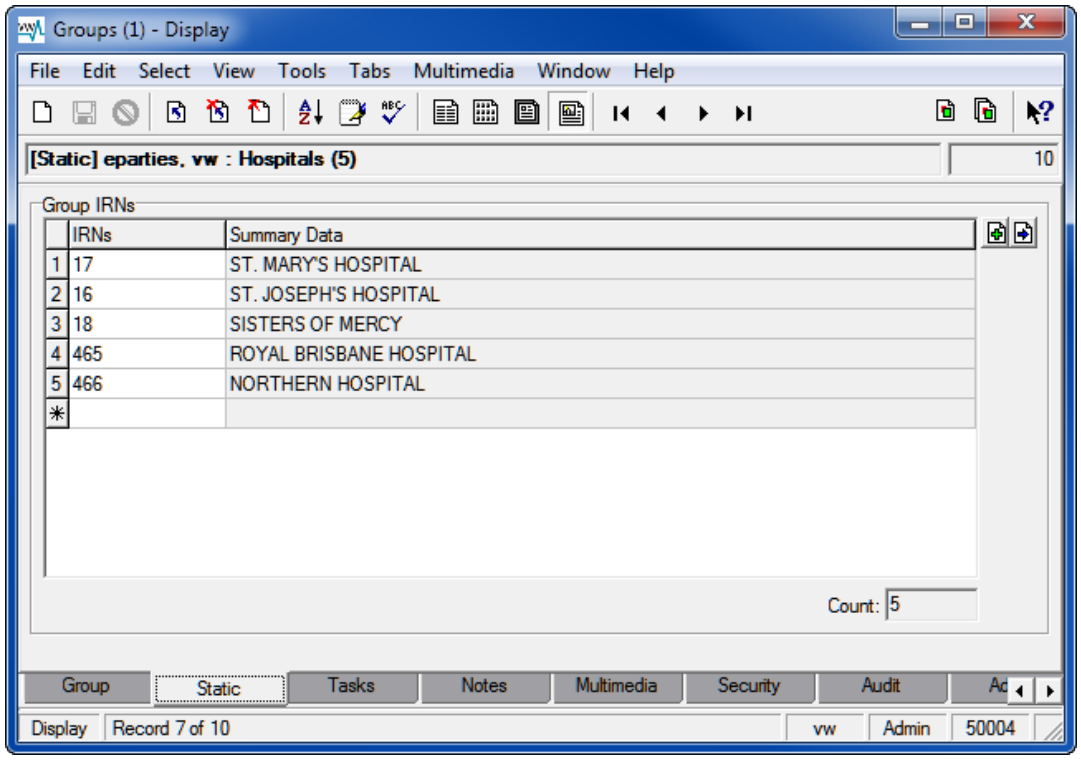
Support of pop-up calendars

Date fields may now have a pop-up calendar associated with them. The calendar may be used to select a date or view the day on which a particular date falls. Users can step backward and forward a month at a time as well as select a specific month or year:



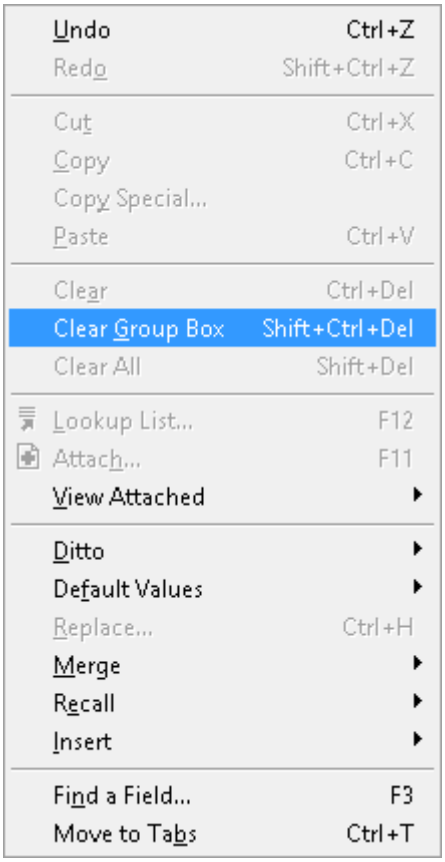
Static groups display Summary Data

The grid containing the list of IRNs making up a Static group has been extended to include the *Summary Data* for each IRN. The grid now provides an easy way to see what records are part of the Static group:



Clear group command

A new command has been added to the `Edit` pull down menu allowing all the controls in a Group box to be cleared. The command is available while entering search terms and inserting or editing records. A keyboard shortcut (`Shift+Ctrl+Del`) has also been made available:



Autofill Lookup Lists

The Vitalware autofill feature that provides a mechanism to add values into a hierarchy where a unique combination is specified has been extended to function on single level Lookup Lists. The change means that when a value is entered into a Lookup List field via the keyboard, the value is replaced with the corresponding Lookup List value if a matching entry exists in the Lookup List module. For example, if a Lookup List contains the value `IBM` and a user enters `ibm`, the user's text (`ibm`) will be replaced with the Lookup List value (`IBM`). Institutions may use this feature to enforce consistent entry of values.

Scan multi-page documents

Vitalware now supports the scanning of multi-page documents where the connected scanner supports such functionality. Once the pages are scanned, a multi-page image format should be used to save the document (e.g. TIFF). If the selected image format does not support multi-pages, then only the first page is saved.

Scan from all modules

The ability to scan documents has been added to all Vitalware modules that allow multimedia to be displayed. The `Multimedia` pull down menu has two new commands:

- `Setup Scanner`
- `Scan Image`

Alter POS payment type

Vitalware now allows you to alter a payment type when it is determined that an incorrect payment type has been recorded as part of balancing end of day. This may be done by using the new epos Change Payment Type admin task.

Error tracking with web orders

It was possible for erroneous web orders not be loaded so that they appeared to be missing in the day's sequence of orders. The processing of web orders now terminates so that erroneous web orders can be investigated. Additionally the original data from all web orders is now transferred to the POS order to better facilitate enquiries from customers.

Balancing

For balance checking a new parameter (`-i`) has been added to the `endofday` script which

end of day	allows users to run end of day even if there are still some tills signed on.
Work queues	Work queues may now be defined by setting queue conditions in the Registry. See the Vitalware Help for details.
NRS schema validation	Support has been added to the NRS export mechanism to validate exported records against defined schemas. To invoke this functionality xmllint (http://www.xmlsoft.org) must be installed on the production server and the `NRSXSDVALIDATION` environment variable must be set to a non `0` value.

Issues Resolved

Issue	Resolution
The *Multimedia* and *Supplementary_tab* fields may not appear in the list of fields available for selection when creating a report. In order to add the fields it is necessary to add *Reportable* Registry entries for each of the fields.	The Registry entries are no longer required as the *Multimedia*and *Supplementary_tab* fields are now shown in the field list (permissions permitting).
The auditserver may not complete processing the current record when it is terminated. When it is restarted it may then process the same record again, possibly resulting in two entries in the Audit Trails module.	The auditserver now finishes processing the current record before shutting itself down.
Admin Tasks that have specified input types of either number, date or time do not have values entered validated. Hence it is possible to enter a badly formatted number, date or time value that is not detected until the server-side script checks its arguments.	Numeric, date and time values entered as input values in Admin Tasks are now checked when the field containing the value is exited. If the value is not valid, a suitable message is displayed and the value must be amended.
The vwlutsrebuild server-side command does not load system based Lookup Lists when they are named on the command line. For example, vwlutsrebuild -t 'System Yes' does not add the Lookup List entry for the System Yes Lookup List.	System based Lookup List entries are now loaded correctly when named on the command line.
When selecting a file to import into Vitalware the file selection dialogue box restricts the file extensions that may be chosen. The list of possible files is restricted to those matching *.csv, *.tab, *.txt and *.xml. In some cases an Import file may not match one of the listed extensions, forcing users to rename files.	The file selection dialogue box has been extended to allow *All Files (*.*)* to be selected. The change allows any file to be chosen. Vitalware parses the first part of the file to determine whether it contains XML or CSV based data.
When copying a large number of rows from a list of matching records onto the Windows clipboard, the operation may take a long time. In particular, if the data contains values from other modules the time taken may be substantial.	The time taken to copy records onto the Windows clipboard has been reduced dramatically where a large number of records is involved.

Issue	Resolution
The file permissions set on the Vitalware server allow access to server-side files for users registered in the UNIX group `vwadmin`. In general, user `vw` is the only account that is part of the `vwadmin` group. If a system administrator chooses to add another user to the group, the user may be able to view information that is sensitive.	The Vitalware server-side permissions have been tightened so that only user `vw` may view files containing information that may be sensitive.
The header displayed at the top of each column in List mode may not display the text correctly if the text contains extended characters (that is characters with diacritics).	The header text is now displayed correctly for extended characters.
The data shown in the Shortcut grid for a given record may not display correctly. The issue only arises for users with Unicode based data where the data contains extended (or non-ASCII) characters.	Unicode data is now displayed correctly in the Shortcut grid.
The message box displayed asking for confirmation for a new entry to be added to a Lookup List may not display the new value correctly for Unicode based systems where the value contains extended (or non-ASCII) characters.	The correct text for the new value is now displayed for extended Unicode characters.
When a series of rows in List mode is copied to the Windows clipboard and then pasted into Microsoft Word any diacritic characters will not be displayed correctly. The issue only occurs for non-Unicode based clients. The issue does not occur if the *Paste Special...* command is used in Microsoft Word.	All diacritic text pasted into Microsoft Word now displays correctly.
The error message *Cannot allocate memory* may be displayed when copying a large number of rows from List mode onto the Windows clipboard. The issue only occurs where the number of rows is in the tens of thousands.	The error message no longer appears when copying a large number of rows onto the Windows clipboard.
An incorrect hint string is displayed when hovering over the Discard Record speed button. When a series of records is selected the *Discard Current* hint is displayed, rather than the *Discard Selected Records* hint.	The correct hint is now displayed when hovering over the Discard Record speed button.

Issue	Resolution
The command to insert the current date (*Edit>Date>Current Date)* may insert the date for the previous day. The incorrect date, while consistent, will only occur spasmodically.	The correct date is now inserted.
The `lutserver` server-side load may fail with a `Not enough space` error message even though there is plenty of space available. The error occurs when the load has a large number of updates to process.	The `lutserver` load no longer fails with an `Not enough space` error message when processing a large numbers of updates.
The `Column Access Modifier` Registry entry may not be triggered until a record is saved. The issue only arises if the column being monitored for changes is not one of the columns used for tab switching.	The `Column Access Modifier` Registry entry is always triggered when the column being monitored is exited after being modified.
If text is entered into a grid control in Query mode and the Insert Record speed button is selected before exiting the grid, an *Access Violation* error message may be displayed.	The error message no longer displays when clicking the Insert Record speed button.
If an Admin Task is defined in a multi-lingual system where strings are provided for all languages as part of the Registry entry, then the strings displayed show all languages rather than the current language selected.	The Admin Task now displays strings in the current language selected, rather than displaying all strings.
If a user has defined a different font colour for data when editing or creating a record and a value in a computed field is modified, the record will stay in Edit mode after the save command has been invoked. The issue only arises where a special handler has been installed to handle a computed value (e.g. Currency conversion).	The record no longer remains in Edit mode for computed fields when saved.
The ICS files generated by Vitalware may not contain the correct format. The ICS files contain a list of notification dates suitable for uploading into calendar applications like Microsoft Outlook, etc.	The ICS files generated now contain the correct format.
The `lutserver` server-side load may create a new Lookup List entry for Unicode based entries that already exist where the entry contains extended (or non-ASCII) characters. The entry created may not contain the correct data.	A new entry is no longer created when the Unicode based Lookup List entry exists, even if it contains extended characters.

Issue	Resolution
The `lutserver` server-side load may generate an error when checking whether a Lookup List value still exists. The `Server error: Can't find "/home/vw/`*`client`*`/data/`*`column`*`" directory. Unable to resolve "`*`column`*`"` error message only occurs where the hierarchy contains a virtual column. In this case the Lookup List entry should not check the column.	A Lookup List hierarchy containing a virtual column is no longer checked by the `lutserver` when determining whether an entry is still valid.
An *Access Violation* error message may be shown when clicking on the After Export tab in the Export Properties dialogue box. The issue only occurs if no After Export scripts have been installed.	The error message no longer appears when clicking on the After Export tab in the Export Properties dialogue box.
If a Lookup List field contains a value consisting of punctuation characters only, then when the record is saved the user will be asked to confirm the insertion of a new Lookup List even if the value is already in the Lookup List.	The confirmation message for Lookup List values which consist of punctuation only is displayed once, rather than each time the value is encountered.
An *Access Violation* error message may be shown when copying rows from List mode to the Windows clipboard. The message is only displayed if one of the columns copied is a client-side column (*DataLocal* column).	The error message is no longer displayed when copying rows to the Windows clipboard where one or more of the columns is a client-side column.
If a new value is entered into a Combo Box and the record saved, the value is added to the Lookup List module but it may not be added to the Combo Box drop down list. The next time the list is dropped down, the value may be missing.	New values are now added to the drop down list when the record is saved.
The `vwlutsrebuild` server-side program may change the Hidden flag on a Lookup List entry from `Y` (Yes) to `N` (No) when rebuilding a Lookup List. The issue occurs only where the default Lookup List permissions for the column do not contain `ReadIgnore`, `WriteIgnore` or `AutoWriteIgnore`.	Lookup List entries with the Hidden flag set to `Y` (Yes) are no longer changed by the `vwlutsrebuild` server-side program.
If a report contains a reverse reference field, under certain circumstances the data associated with the field may not be included in the report. The issue only arises if the data for the field is a single value. Multiple values are reported correctly.	The data for single value reverse reference fields is now reported correctly.

Issue	Resolution
Error messages generated by the Vitalware server are not displayed correctly if the text contains extended (or non-ASCII) characters. In particular, French and Arabic errors may display incorrectly.	Vitalware server error message now display the correct text for all languages including text with extended characters.
Under certain circumstances the last character in the Extended Data text may not be displayed as bold. The issue only occurs if the data contains more than one line of text.	All of the Extended Data text is now displayed in bold.
When copying data from List mode to the Windows clipboard and pasting it into Microsoft Excel using the *Paste Special: CSV* command, the pasted values may not appear in Excel.	Data pasted into Microsoft Excel via the *Paste Special: CSV* command now appears correctly. Due to limitations in Excel, Unicode based data may contain some question mark characters where Excel cannot map the character to its internal character set.
A *System Error. Code: 1400. Invalid Window Handle* error message may be displayed after a record is saved. The error only occurs if a default value is set on a Combo Box control that is only visible in Display mode (a very rare combination).	The error message is no longer displayed when saving a record.
If a drag and drop operation is used to add records to a grid control where the record has been changed by another user, then the record is refreshed and the drop operation removes the changes made by the previous user. The issue only occurs if the grid on which the drop operation is performed was modified by another user after the record was retrieved.	If a drop operation occurs on a grid that has been modified by another user, a message is displayed and the drop operation is aborted. The change ensures no data is lost.
Some currency based calculations may result in rounding errors. The issue only arises where the currency value is greater than seven digits of precision.	The rounding error no longer occurs where a currency value consists of more than seven digits of precision.
If data is copied from List mode and then pasted into Microsoft Word while keeping the source format, the data is no longer displayed in a table but rather as unformatted text.	If the *Keep Source Format* option is selected in Microsoft Word, the pasting data will now appear as a table rather than unformatted text.
The `luts` server-side audit handler generates an error when a name is not associated with a hierarchy. The error causes other audit handlers for the current audit record to be skipped. The issue only arises if a hierarchy has not been configured correctly.	An error is no longer generated if a hierarchy is not configured correctly. The hierarchy is skipped and all audit handlers are executed correctly.

Issue	Resolution
An Admin Task may be executed inadvertently by displaying the list of available Admin Tasks and then double clicking anywhere in the Admin Task dialogue box. The issue only arises if a task is selected when the double click is executed.	An Admin Task is only executed if the double click occurs on the task selected, otherwise no action is taken.
If the `vwlutsrebuild` server-side command is executed with a Lookup List name supplied as a command line argument and the lookup name matches an existing Lookup List name but the character case is different, then the Lookup List entries for the existing name are deleted.	The Lookup List entries are no longer deleted if the Lookup List name supplied does not match exactly the character case of an existing Lookup List.
The Mandatory Modifier Registry entry may not be triggered when a dependent value changes in a nested column. The issue only occurs when the value to check is part of a nested column (a column name ending in *_nesttab*).	The Mandatory Modifier Registry entry is now triggered correctly for nested columns.
Under certain circumstances the value displayed in a query grid for an attachment query (e.g. <# 4 #>) may result is a search for the number of records (in this case 4) rather than the attached records. The issue only occurs if the cursor is placed in the query field after the attachment process is completed.	The attached records are always used as part of the query rather than the number of attached records.
Under certain circumstances the resolution height and width set on the Multimedia tab in the Report Properties dialogue box may not be honoured when the report is generated. The issue only arises if the *Multimedia* field is the last field listed in a report group.	The resolution height and width settings are honoured when the *Multimedia* field is the last field listed in a report group.
If a multi-page TIFF image is ingested by the Multimedia Repository, then the last page of the document is used to generate the thumbnail image rather than the first page. Since the last page may be a lower resolution image of the first page a blurry thumbnail may be produced.	The first page of a multi-page TIFF document is used to generate the thumbnail and resolution images.
The image types PDF and DNG are not recognised as image formats when producing a report with the **Use first image only** option enabled. As such, an image is not displayed in the report.	Both PDF and DNG are now recognised as image formats when generating reports.

Issue	Resolution
The Summary tab in the Audit module may not display extended (or non-ASCII) Unicode characters correctly. Data stored as ISO-8859-1 (latin1) will be displayed correctly.	Unicode data is now displayed correctly on the Summary tab in the Audit module.
If a double click occurs on a row in List mode or on a cell in Contact Sheet mode and the mouse cursor is moved away from the row or cell clicked, under certain circumstances the record displayed in Details mode may not be the same as the record on which the click occurred.	The correct record is now displayed in Details mode based on the row or cell on which the double click occurred.
Dupping of data to a remote host fails when one of the dup files is not a database name.	Data is now dupped to the remote hosts irrespective of the dup file name.
It is possible to perform functions (Process Menu) on a record even though the record has a pending maintenance/amendment.	Functions from the Process Menu can no longer be performed if the record has a pending maintenance/amendment.
The POS module does not offer all of the visible fields for selection when defining a new List display.	All fields displayed on a tab are now available for selection for a List display.
For systems using Batch printing it is possible for the POS record to briefly go into Edit mode even when a certificate does not print.	The POS record no longer goes into Edit mode when a certificate does not print during Batch printing.
On rare occasions when using Batch printing for an individual record, certificates can still print even though the print lock could not be obtained.	Batch printing will not proceed until the print lock is obtained.
On rare occasions, missing stock returns an error *Order Check Failed*, leading the user to believe that the report has not run.	All error reporting has been moved into the report output ensuring that the user always sees the report complete.
When applying an amendment/maintenance it is possible for the `PENDING` record to be left in Edit mode if an unexpected error (Unique Number) occurs.	Rollback is correctly called when an unexpected error occurs during the apply amendment/maintenance process ensuring the user is returned to Display mode.
Occasionally when trying to save a POS record a *Grid Index Out Of Bounds* error would be shown.	The *Grid Index Out Of Bounds* error no longer displays when the record is saved.
When dropping a party record into a registration with an incomplete date of death and a date output format set to `yyyy-MM-dd`, a date decode error can occur.	A party record may now be dropped without a date decode error occurring.

Issue	Resolution
An api error is shown when querying on the Table column in eweblog.	An api error is no longer shown when querying on the Table column in eweblog.
Running `reportsupdate` with the `-t` option does not add the paper size to the Report Registry entry.	Running `reportsupdate` with the `-t` option now adds the paper size to the Report Registry entry.
When processing matches through the Match List database it is possible to overwrite an existing link.	An error is now shown if an existing link is already in place for a list match.
When running overnight matching it is possible for the same match pair to be inserted in the Matchlists database multiple times.	The same match pair are only inserted once into the Matchlists database.
The email button does not refresh when changing rows on the Sales tab.	The email button refreshes when changing rows on the Sales tab.
Running `reportsupdate` in test environments may result in incorrect Registry entries being inserted.	Only correct Registry entries are inserted when running `reportsupdate` in a test environment.
The Sales and Payment grids in POS allow the movement of individual cells when only row movements should be allowed.	Individual cells can no longer be moved in the POS Sales and Payments grids.
In the NRS system there is an XML conversion issue with diacritic characters being read as HTML code.	The NRS system correctly handles diacritic characters.
On occasions the Communications module will not list all templates or load a template once selected.	The Communications module correctly lists all templates and loads each selected template.
When dropping a product into POS, the product is accepted even if the Sales row is read-only.	The product is only accepted if the row is not read-only.
A *List Index Out Of Bounds* error can occur in POS when keying a two part registration number.	A *List Index Out Of Bounds* error no longer occurs in POS when keying a two part registration number.
When issuing a credit card refund it is possible for the payment Till reference to be altered to the refund Till reference.	The payment Till is no longer altered when a credit card refund is performed.
When multiple users add notes to a record at the same time it is possible for notes to be lost.	Notes are no longer lost when multiple users add them at the same time.
It is possible for Audit records to be corrupted when dupping to multiple machines.	Audit records are not corrupted when dupping to multiple machines.
Certificate print statistics may not be correct.	Certificate print statistics are generated correctly.

Issue	Resolution
On occasions the `vwmaintenance` script may not complete.	The `vwmaintenance` script now runs to completion.
Users with appropriate permission are able to go into Insert Mode in the Matchlists module even though they are unable to insert any record data.	Users are no longer able to go into Insert mode in the Matchlists module.

Upgrade Notes

The upgrade from Vitalware Version 2.3 to Vitalware 2.4 involves a number of steps. Please follow the instructions below carefully.

Do not skip any steps under any circumstances.

Before proceeding with the update please ensure that a complete backup of the Vitalware server exists and is restorable.

There are four components that require upgrading:

- Texpress (the database engine)
- TexAPI (web services)
- Vitalware Server (the application)
- Vitalware Client (the client)

The notes below detail how to upgrade all systems. Check the Releases table for Client specific notes.

In the notes below, *clientname* refers to the name of the client directory for the current installation. The term `~vw` is used to refer to user `vw`'s home directory. This is normally `/home/vw`.

Stopping Vitalware services

1. Log in as `vw`
2. Enter `client` *`clientname`*
3. Enter `ls -l loads/*/data* local/loads/*/data*`
4. Check that each `data` directory is empty and that no `data.t` files exist.
   If `data.t` files do exist, please wait for the loads to drain before proceeding.
5. Enter `vwload stop`
6. Enter `vwweb stop`
7. Enter `texlicstatus`
   Make sure no one is using the system.
   The upgrade will not complete successfully if users are accessing data.

Record Session

Each step in the upgrade process produces detailed output. In most cases this output will exceed the size of the screen. It is `strongly` recommended that the output of the upgrade session is recorded, so if errors occur, the output can be examined.

1. Enter `script /tmp/output-2-4`

A new shell will start and all output recorded until the shell is terminated.

Installing Texpress

Installing Texpress 8.3 is only required for the first client upgraded to Vitalware 2.4. Once Texpress 8.3 has been installed, this section may be skipped for subsequent upgrades.

1. Enter `cd ~vw`
2. Enter `mkdir -p texpress/8.3.xxx/install`
3. Enter `cd texpress/8.3.xxx/install`
4. Obtain the appropriate Texpress version for your Unix machine.
   Save the release in `~vw/texpress/8.3.xxx/install`, calling it *texpress.sh*.
5. Enter `sh texpress.sh`
   The Texpress release will be extracted.
6. Enter `. ./.profile`
7. Enter `bin/texinstall~vw/texpress/8.3.xxx`
   The Texpress installation script will commence.
8. Enter `cd ~vw/texpress/8.3.xxx`
9. Enter `. ./.profile`
10. Enter `bin/texlicinfo`
    Obtain your Texpress licence code and place it in a file called `.licence`.
11. Enter `bin/texlicset< .licence` to install the licence.
12. Enter `\rm -fr install`
13. Enter `cd ~vw/texpress`
14. Enter `ln -s 8.3.xxx 8.3`

Upgrading KE TexAPI

Installing TexAPI is only required for the first client upgraded to Vitalware 2.4. Once TexAPI has been installed, this section may be skipped for subsequent upgrades.

1. Enter `cd ~vw/texpress`
2. Enter `mkdir 6.0.xxx`
3. Obtain the appropriate TexAPI version for your Unix machine.
   Save the release in `~vw/texpress`, calling it *texapi.sh*.
4. Enter `sh texapi.sh -i~vw/texpress/6.0.xxx` (expand the `~vw`).
5. Enter `\rm -f texapi`
6. Enter `ln -s 6.0.xxx texapi`
7. Enter `\rm -f texapi.sh`

Upgrading Vitalware Server

1. Enter `cd ~vw/clientname`
2. Enter `mkdir install`
3. Enter `cd install`
4. Obtain the appropriate Vitalware server version bundle.
   Save the release bundle file in `~vw/clientname/install` calling it *vw.sh*.
5. Enter `sh vw.sh`
   The Vitalware release will be extracted.
6. Enter `. ./.profile`

7. Enter `bin/vwinstall`*`clientname`*
   The Vitalware installation script will commence.
8. Enter `cd ~vw/`*`clientname`*
9. Enter `cp .profile.parent ../.profile`
10. Enter `. ../.profile`
11. Enter `client `*`clientname`*
12. Enter `vwreindex`
13. Removal of the temporary directory (and its contents) is recommended:
    Enter `\rm -fr install`
14. Enter `upgrade-2-4`
    The client will now be upgraded to Vitalware 2.4. If you are upgrading from a version prior to Vitalware 2.3, you must run the upgrade scripts for all versions after the old version `before` running the Vitalware 2.4 upgrade.

Starting Vitalware services

1. Enter `vwload start`
2. Enter `vwweb start`

Record Session

The recording of the upgrade session may now be terminated.

1. Enter `exit`

The session output is available in `/tmp/output-2-4`.

Upgrading Vitalware Client

Vitalware 2.4 does not require the new Windows client to be installed on every machine for network installations. Updating the network server is sufficient. For standalone installations a new client is required on each machine. To upgrade the Vitalware Client follow the Installing Vitalware Client notes.

Vitalware®
Vital Records Management

The World's Premier Vital Records Software

Vitalware Documentation

Editing in a single language

Document Version 1.1

Vitalware Version 2.3

www.kesoftware.com

Contents

SECTION 1

Overview

It has always been possible in Vitalware to store data in multiple languages in the same field. For instance, a client can specify that their data consists of English text followed by French text with a delimiter marking the end of one language and the start of another. The default delimiter is `;:;` (i.e. semi-colon colon semi-colon). For example:

```
hat;:;chapeau
```

indicates that `hat` is in one language and `chapeau` is in another language.

A number of Vitalware facilities need to understand what languages are used and in what order they are stored. The spell checker, for instance, must be able to determine what languages are available and in what order so that the correct dictionary can be loaded and used against the correct values. Vitalware uses the Supported Registry entry (page 6) to calculate how many languages are supported, what those languages are, and in what order they are stored. For example:

```
System|Setting|Language|Supported|0;1
```

specifies that two languages are supported, English (International) and French (language `0` = English (International) and language `1` = French), and that they are stored in that order.

See the Supported Registry entry (page 6) for details of Vitalware language codes.

Given a data value of `hat;:;chapeau` and the previous Registry entry we see that `hat` is in English (International) and `chapeau` is in French.

It is important to understand that the Supported Registry entry also defines the order in which the language data is stored within Vitalware. If the previous Supported Registry entry was changed to:

```
System|Setting|Language|Supported|1;0
```

the value `hat` would then be treated as French and the value `chapeau` as English. When planning to use Vitalware to store multiple languages it is very important to define the order of the languages before entering data into the system. Once data has been entered, any changes to the language order will require existing data to be exported, manipulated (swapping the order of the data) and imported. While this is possible it is far easier to leave the order in which languages are stored unchanged once set.

Keep in mind that there is an important difference between the storage and the display of language data in Vitalware: once defined using the Supported Registry entry, the order in which language data is stored in Vitalware cannot be changed easily. As we'll see however, it is a simple matter to change the order in which language data is displayed in the Vitalware client (page 17).

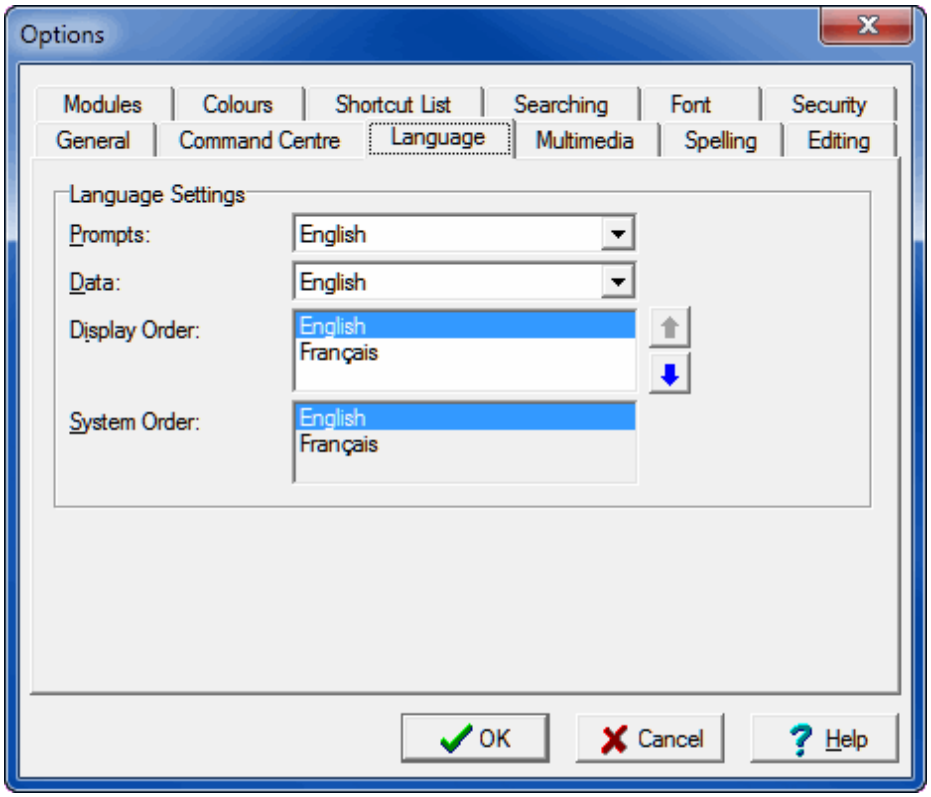
Adding one or more languages is a simple matter however, and only requires adding the relevant language code(s) to the end of the Registry entry. In this case there is no need for any data manipulation. For example, if Arabic support was needed, our earlier

Supported Registry entry would simply be changed to:

`System|Setting|Language|Supported|0;1;12`

where `12` is the language code for Arabic.

When Vitalware displays data on the screen it can show values for all languages or for an individual language. The setting used to control which languages are displayed is found on the Language tab of the Vitalware Options box (page 17):



If a system only supports one language, the Data drop list and Display Order are disabled. The Data alternatives allow users to decide how data should be displayed within modules in Vitalware. In Vitalware versions prior to version 2.3, if a single language is chosen from the Data drop list, it is not possible to make changes to the data and all controls in a module are disabled. It is only possible to edit or insert data while viewing the data in `All Languages`. This ensures that when editing data, users can view the complete contents of a field while altering it. In effect it forces users to consider all supported languages while entering data.

Vitalware 2.3 has seen the introduction of a number of extensions that provide a more flexible mechanism for displaying and altering data within a single language:

- When the `All Languages` option is selected, users may now select the order in which languages are displayed (but not, it is important to keep in mind, the order in which language data is stored in Vitalware, which is defined by the Supported Registry entry). This allows users who are dominant in a particular language to display this language first with other languages following.
- User may now alter data while a single language is displayed. Any data entered into a field will only update the value of the language selected.
- When displaying data in a single language and a value does not exist for a given field, the value of the first filled language (page 13) may be displayed. The text is shown in a user selectable colour (grey by default) to indicate that it is not

associated with the current language. When the field is entered, the text is removed, allowing data to be entered.

- A number of data entry helpers always display all languages in the user defined order to assist data entry. For example, if a Lookup List is displayed (page 31), all languages are always shown so that users may select the correct entry based on context. This is particularly important where one term in a given language has multiple terms in another language.

The remainder of this document describes these new features in detail.

SECTION 2

Registry settings

Seven Registry entries control how language data is stored and displayed within Vitalware.

With the exception of the Supported Registry entry these entries are only applicable in an Vitalware system that supports more than one language: they have no effect in a single language system.

The Registry entries are:

Supported Registry entry

Registry Entry	Purpose
`Supported`	In a single language environment this entry specifies which dictionary is used for spell checking. In a multi-language environment this entry also specifies a list of supported languages and the order in which data in each language is stored within a field of the database (e.g. English, German, then French). This storage order can differ from the order in which data in each language displays (page 17) in Vitalware (e.g. French, German, then English).

The format of this Registry entry is:

```
System|Setting|Language|Supported|#[;#;#...]
```

where:

`#[;#;#...]` is one or more language codes. If more than one, the codes are separated with a semi-colon.

`0`	English
`1`	French
`2`	English (US)
`3`	Spanish
`4`	German
`5`	Italian
`6`	Dutch
`7`	Danish
`8`	Polish
`9`	Norwegian
`10`	Swedish
`11`	Greek
`12`	Arabic
`13`	Hebrew
`14`	French (CA)
`15`	Finnish

This entry would be suitable for a system that supports Arabic, French and English:

```
System|Setting|Language|Supported|12;1;0
```

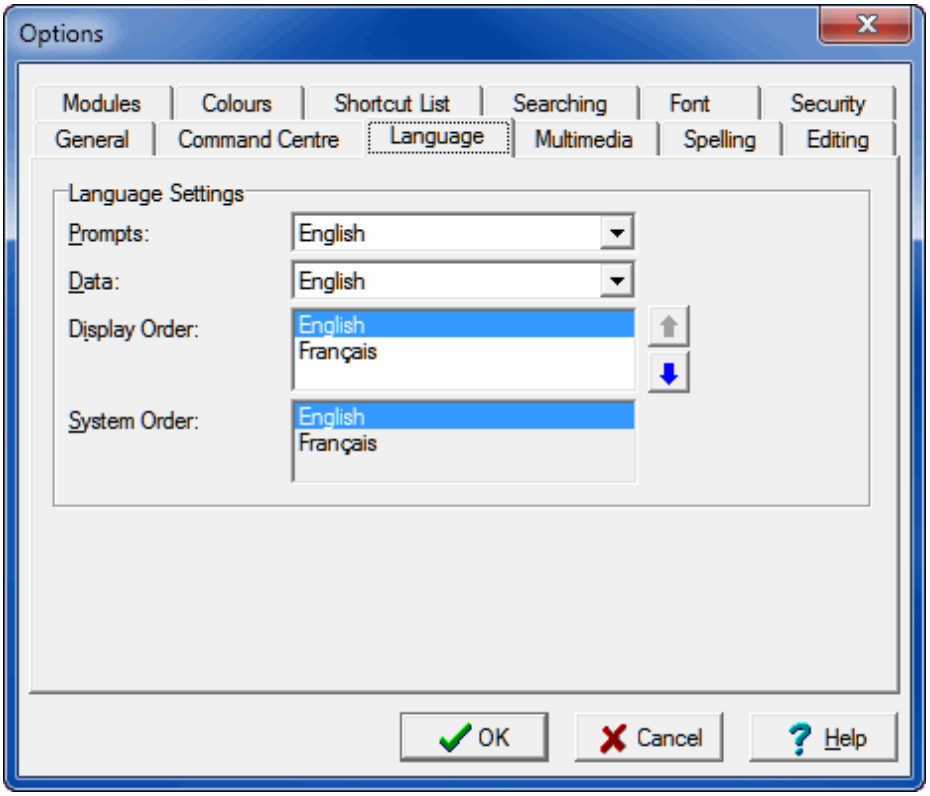
The order of the languages specified in the entry defines the order in which data is stored within Vitalware. In this example, data will be stored in a field in Arabic, French and then English.

Once data has been entered with a particular order set, it is not recommended that the order is altered. If the order is to be changed, the existing data must be exported, manipulated and reloaded.

The language order defined by this entry is used by a number of Vitalware facilities that are language dependent (e.g. spell checker).

There is no limit to the number of languages that can be specified.

This system order (or data storage order) of languages specified using the Supported Registry entry is displayed on the Language tab of the Options box:



If a system is to operate with only one language, only a single language number is specified. If the Supported Registry entry does not exist, English (International) is assumed by default (language code `0`). The following Registry entry would be suitable for an Vitalware system that supports English with USA based variations:

```
System|Setting|Language|Supported|2
```

The Supported Registry entry should never be set on a user or group basis.

Delimiter Registry entry

Registry Entry	Purpose
`Delimiter`	In a multi-language environment this entry specifies the marker used to separate each language in a field.

The Delimiter Registry entry specifies the character sequence that appears between languages. The default sequence is `;::;` (semi-colon colon semi-colon). There is no restriction on the length of the delimiter. It is not possible to have the delimiter sequence treated as text, so a sequence should be selected that will never occur in data values.

The delimiter should be set system wide and never on a per user or group basis.

This entry sets the character sequence `###` as the delimiter between languages:

```
System|Setting|Language|Delimiter|###
```

Edit Any Language Registry entry

Registry Entry	Purpose
`Edit Any Language`	In a multi-language environment this entry specifies whether users can alter data when a single language is displayed.

By default, when a single language is displayed in a multi-language environment all display controls are disabled (greyed out) and users are not able to alter data. Setting the Edit Any Language Registry entry to `True` allows users to enter values when a single language is displayed. All values updated by the user via the data entry fields only update values for the language currently displaying.

The Edit Any Language Registry entry may be set on a system-wide, per user or group basis.

This entry will enable all users to edit data in a single language:

```
System|Setting|Language|Edit Any Language|True
```

Together, the following entries would restrict the ability to edit in a single language to users in group Admin:

```
System|Setting|Language|Edit Any Language|False
Group|Admin|Setting|Language|Edit Any Language|True
```

Show Trailing Delimiters Registry entry

Registry Entry	Purpose
`Show Trailing Delimiters`	In a multi-language environment this entry specifies whether the language delimiter sequence should be displayed for trailing languages when there are no values in the trailing languages. This entry can be used in conjunction with Show Empty Delimiters (page 12) to display every delimiter in every field even if there is no data in the field.

By default, delimiters are not shown for trailing languages when there are no values in the trailing languages. Setting the Show Trailing Delimiters Registry entry to `True` will display the delimiters.

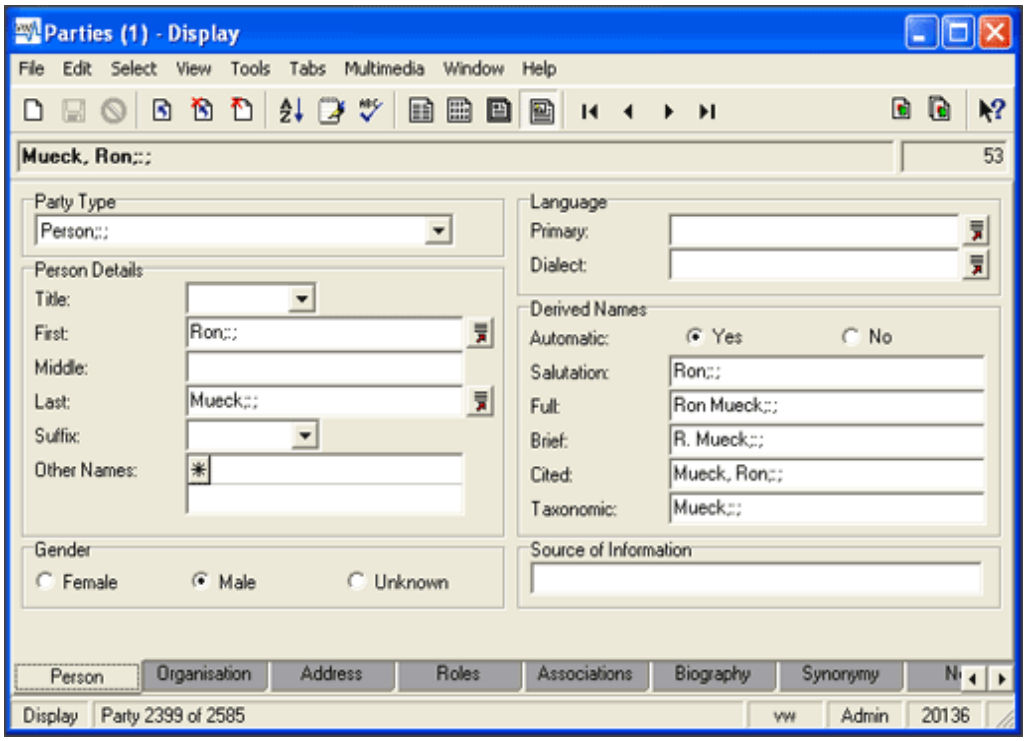
Consider the following data. If we have a system that is configured, via the Supported Registry entry (page 6), to handle two languages and we have a data value of:

```
Some data
```

there are two ways the data could be displayed:

```
Some data
Some data;:;
```

For the first case the value of the Show Trailing Delimiters Registry entry is `False`. In the second case it is `True`. The screen shot below shows data displayed where the Show Trailing Delimiters Registry entry is `True`.



Notice how all fields that contain a value now have a trailing delimiter. The setting does not apply to empty fields. The primary purpose of this entry is to allow users to append the next language to existing values without the need to enter the language delimiter.

The Show Trailing Delimiters Registry entry only works when `All Languages` are showing. This entry may be set on a system-wide, per user or group basis.

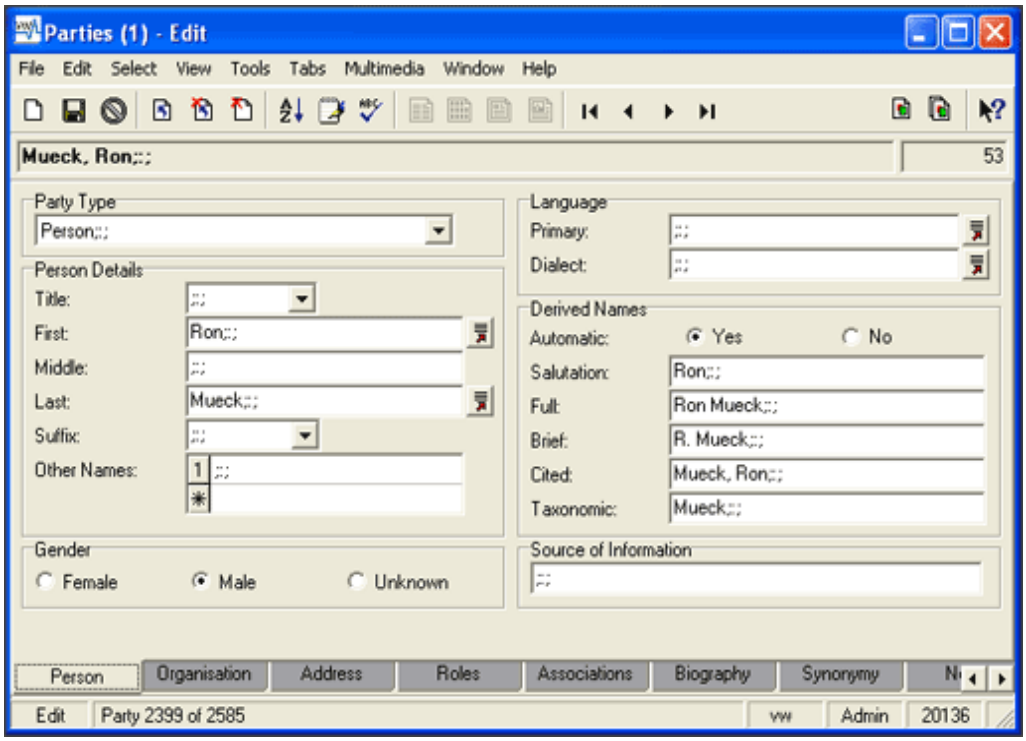
This entry displays trailing delimiters for users in group Document:

```
Group|Document|Setting|Language|Show Trailing Delimiters|True
```

Show Empty Delimiters Registry entry

Registry Entry	Purpose
`Show Empty Delimiters`	In a multi-language environment this entry specifies whether or not the delimiter (page 8) will display in an empty field. This entry is used in conjunction with Show Trailing Delimiters (page 10).

This entry is similar to the Show Trailing Delimiters entry (page 10) except that it applies to empty fields rather than fields with values. The following screen shot shows the Show Empty Delimiters Registry entry enabled (set to `True`), along with the Show Trailing Delimiters Registry entry. Notice how the empty fields now display the language delimiter sequence:



Enabling this Registry entry allows users to enter values into empty fields without the need to enter the language delimiter sequence between values.

> The Show Empty Delimiters Registry entry may be set on a system-wide, per user or group basis.

This entry displays delimiters in empty fields for users in group Document:

```
Group|Document|Setting|Language|Show Empty Delimiters|True
```

Show First Filled Registry entry

Registry Entry	Purpose
`Show First Filled`	In a multi-language environment this entry specifies whether to display data in the first language that has data if there is no data in the display language selected.

This entry only has meaning when a single language is displayed. If `All Languages` is in effect, this entry is ignored.

The entry applies to fields that do not contain any value for the display language, but a value is specified in another language. If the Show First Filled entry has a value of `True`, the first language found to contain a value, as defined by the user's language order (page 17), will be shown. If the entry is set to `False`, a blank field is shown. When the first filled value is displayed, it is shown in a different colour to distinguish it from values that exist for the current language. The default value for this Registry entry is `False`.

This screen shot shows a Parties record that has been entered in English only. The user has chosen to display French data only. If the Show First Filled entry is set to `False`, the tab will appear completely empty except for the *Party Type* value. As the Registry entry has been set to `True`, the English values are displayed where a French value is empty and an English value is available. The English values are displayed in grey (this can be configured to any colour) to indicate that the data is not French:

When a user enters a control containing the English value, by either clicking on the control or using the keyboard to navigate to it, the value is removed. Once the field is exited and a value has not been entered, the English value is displayed again. The image below shows an empty *First: (Person Details)* field as the user has entered the field:

The Show First Filled Registry entry may be set on a system-wide, per user or group basis.

In general, it is recommended that this setting is enabled as it allows users to see values for empty fields where a value has been specified in another language.

This entry enables the display of the first filled language for all users:

```
System|Setting|Language|Show First Filled|True
```

Multiple Languages Registry entry

Registry Entry	Purpose
`Multiple Languages`	Specifies whether a field is multilingual and therefore affected by the other Language Registry entries.

The Multiple Languages Registry entry provides a mechanism for defining whether a field is multilingual. If a field is not multilingual, the field's contents are unaffected by these other Language Registry entries:

- Supported (page 6)
- Delimiter (page 8)
- Edit Any Language (page 9)
- Show Trailing Delimiters (page 10)
- Show Empty Delimiters (page 12)
- Show First Filled (page 13)

Each Vitalware module has a number of fields defined to support multilingual values. System Administrators may adjust the list of multilingual fields to suit their needs.

The format of this Registry entry is:

`Group|`*groupname*`|Table|`*table*`|Multiple Languages|`*column*`|`*value*

`Group|Default|Table|`*table*`|Multiple Languages|`*column*`|`*value*

`Group|Default|Table|Default|Multiple Languages|`*column*`|`*value*

where:

groupname	may be used to restrict the persons to whom the Registry entry applies. In general, group `Default` should be used as a field is either multilingual or not, regardless of the person accessing it.
table	is the name of the table that contains *column*. If the column is common to all modules, `Default` may be used.
column	is the name of the column to which the setting is to be applied. The column name for a given field can be determined by using the **What's this help?** facility. The Column value under the Field Information section contains the required name. See the screen shot below, where *column* is `NamLast`.
value	is `True` (the field should support multiple languages) or `False` (the field should not support multiple languages).

The **What's this help?** information displays not only the column name, but also whether the field supports multiple languages. The Multilingual setting under the *Display Information* section displays `Yes` if the field supports multiple languages.

The following entries turn off support for multiple languages in the *First*, *Middle* and *Last* name fields in the Parties module:

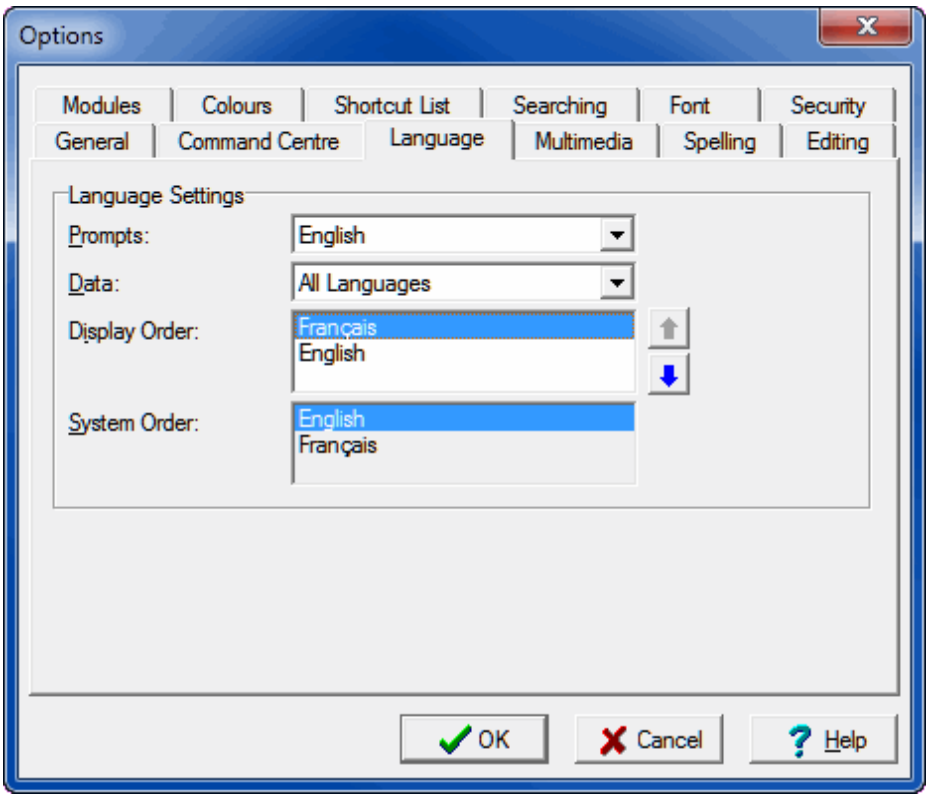
```
Group|Default|Table|eparties|Multiple Languages|NamFirst|False
Group|Default|Table|eparties|Multiple Languages|NamMiddle|False
Group|Default|Table|eparties|Multiple Languages|NamLast|False
```

SECTION 3

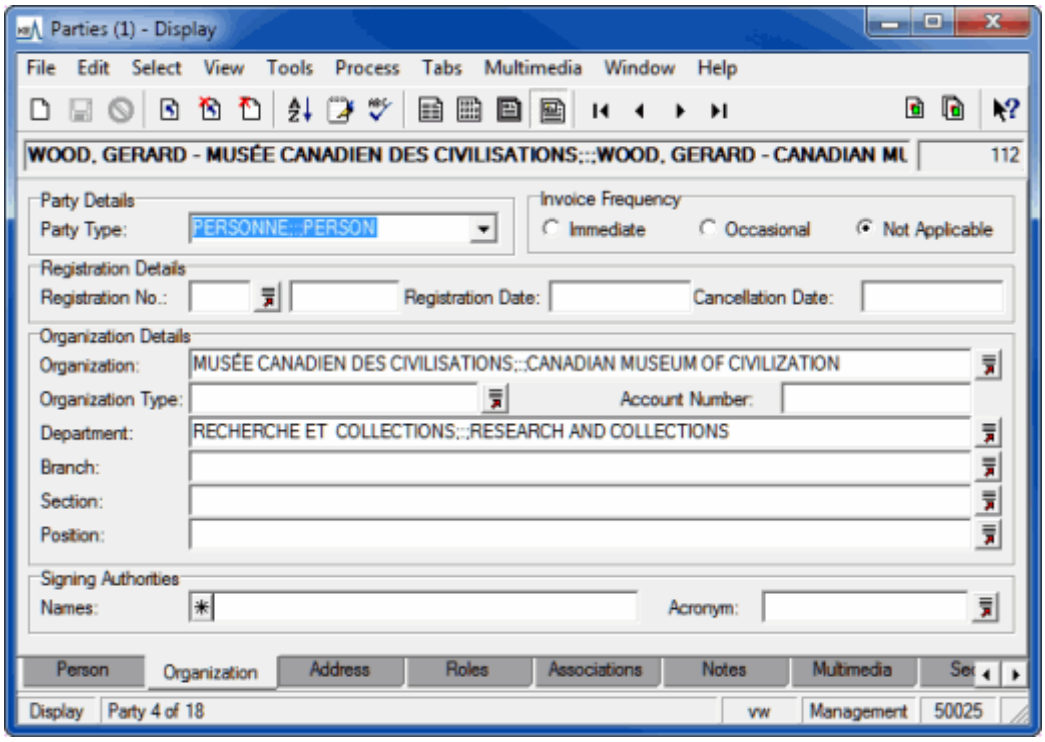
Options

The Language tab in the Options box has been extended to allow the order in which languages are displayed to be specified. The Display Order list box lists each supported language. Select a language in the Display Order list and use the arrow buttons beside the list to change the order in which data in each language displays in the Vitalware client.

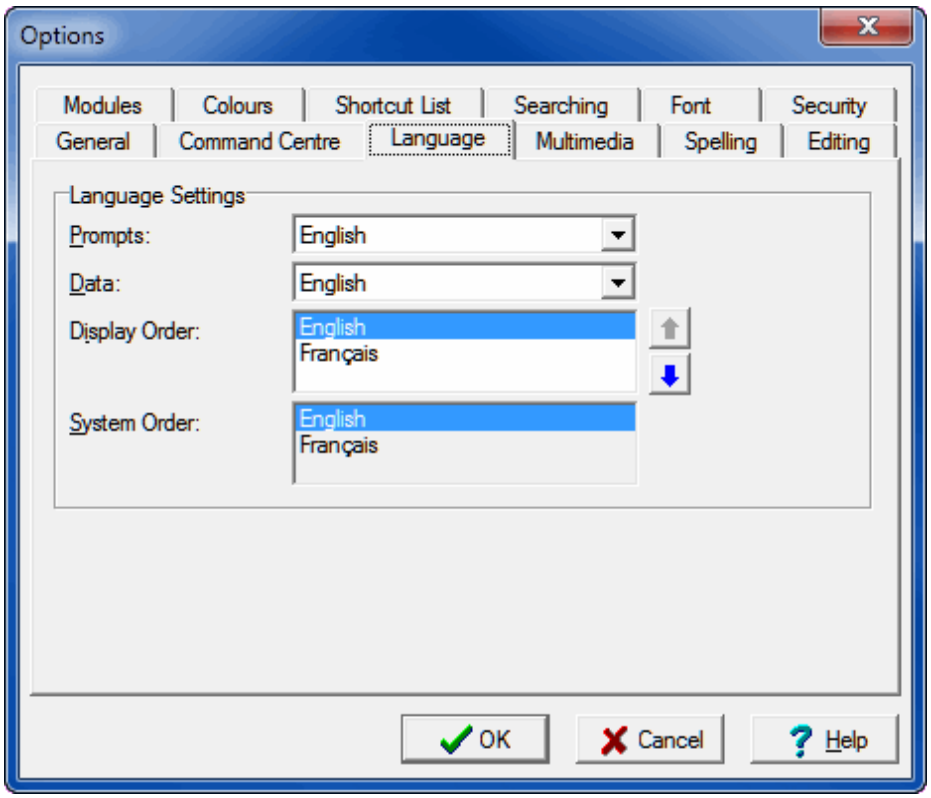
When `All Languages` is selected from the Data drop list, it is possible to specify the order in which languages are displayed in fields in the Vitalware client. For instance, if the supported languages are English and French, the following setting specifies that values in French will display before their translation in English:



This would give us prompts in English, displaying all data with French data displaying before the English translation (e.g. `French;:;English)`:



However, the following setting is also useful:



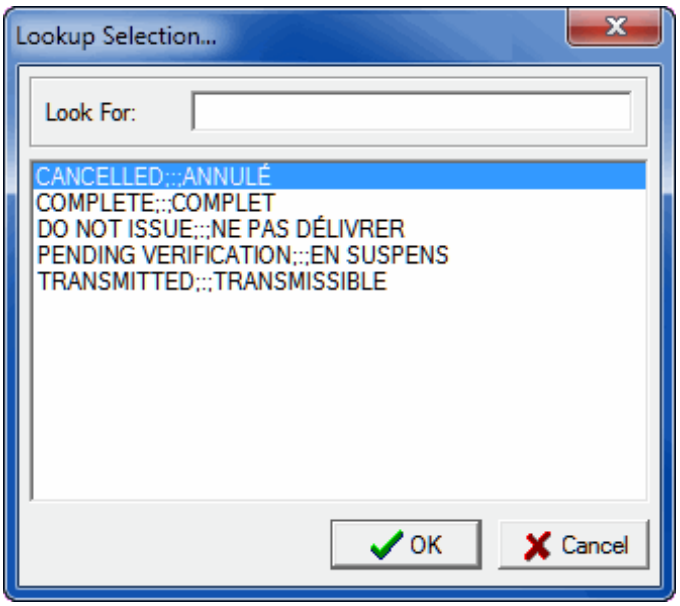
To assist with data entry, various facilities in a multi-language version of Vitalware always show strings with all languages even if a single language is selected in the Data drop list. For example, if a Lookup List box is displayed, all languages are always shown so that users may select the correct entry based on context. This is particularly important where one term in a given language has multiple terms in another language. For example, the French word `voiture` may be translated to English as `car`, but also as `carriage` (as in

a train carriage). As such there are two possible Lookup List entries for `voiture`, namely:

```
voiture;:;car
voiture;:;carriage
```

If the Lookup List only displayed the French entries when French is the data language selected, it would not be clear which variant was required. By showing all languages it is possible to decide which entry is correct.

Language data in these facilities is displayed in the order specified in the Display Order drop list. Based on the Language Settings shown in the last screen shot, prompts and data will display in English, and in a Lookup Selection box English data will display before the French translation:



In general, when a single language is selected for displaying Data, the same language should appear first in the Display Order list.

Each of the Language Settings options is described below:

Prompts	The language selected here determines what language is used to display text. The only text not affected by this setting is data, where data consists of values entered into the system. The setting also affects how dates containing month names are displayed. The Prompts drop down list contains a list of all languages supported by Vitalware. Hence it is possible, but not every useful, to display text in Spanish even though you do not store Spanish data!
Data	The Data setting determines what language should be shown for values that have been entered into the system. A user may select a single language (e.g. English) or `All Languages`. The list of languages selectable is those specified by the Supported Registry entry.
Display Order	The Display Order list contains an entry for each language supported by the system. The list can be adjusted to reflect the order in which each language should be displayed when showing a value in all languages.

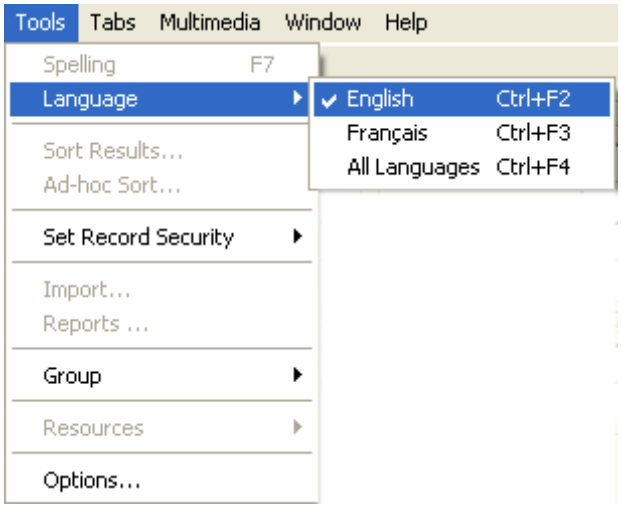
System Order

The System Order list displays the supported languages in the order defined by the Supported Registry entry (page 6). The System Order defines the order in which data is stored in Vitalware. It also defines the order in which data must be submitted for importing (page 48) into Vitalware.

Since the order is defined by a Registry entry it cannot be adjusted via the Options box.

If Vitalware is configured to support only one language, both the *Data* and *Display Order* controls are disabled. The *Prompts* control is always available.

Vitalware also provides a quick mechanism for switching languages without the need to display the Options box. Under the **Tools** menu in any open Vitalware module there is a **Language** sub-menu. The menu is built dynamically and contains an entry for each data language supported by the system, along with the `All Languages` entry (the list is the same as that shown by the Data control in the Options box):

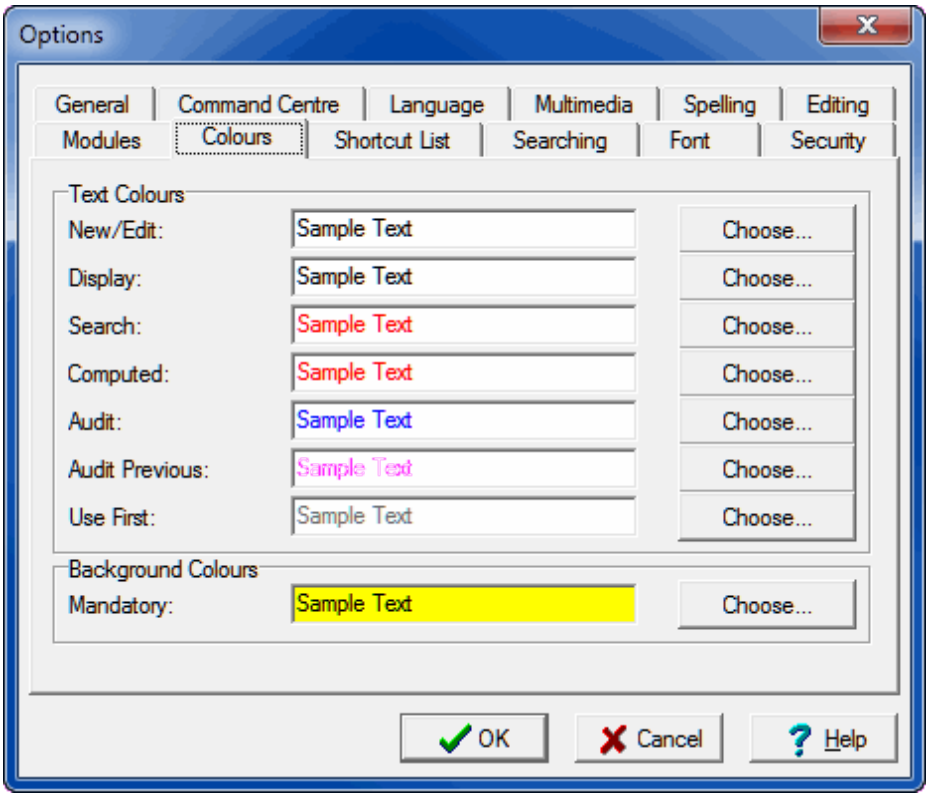


Selecting a language from the sub-menu has the following effects:

- Changes Prompts to the language selected. If the `All Languages` data option is selected, Prompts are not modified.
- Sets Data to the language selected.
- Adjusts the Display Order so that the selected language is the first in the list. If the `All Languages` data option is selected, the display order is not adjusted.

The Language sub-menu does not allow all possible combinations of Prompts, Data and Display Order options to be set, however it does provide the most common alternatives. If further tuning of the settings is required, the Options box should be used.

An option has been added to the Colours tab in the Options box, allowing the colour used to display the text for the Use First filled value to be specified:


Options
General
Command Centre
Language
Multimedia
Spelling
Editing
Modules
Colours
Shortcut List
Searching
Font
Security
Text Colours
New/Edit:
Sample Text
Choose...
Display:
Sample Text
Choose...
Search:
Sample Text
Choose...
Computed:
Sample Text
Choose...
Audit:
Sample Text
Choose...
Audit Previous:
Sample Text
Choose...
Use First:
Sample Text
Choose...
Background Colours
Mandatory:
Sample Text
Choose...
OK
Cancel
Help

SECTION 4

Editing in a single language

In this section we look at how specific functionality in a multi-language Vitalware environment changes when editing in a single language:

Data entry

Processing of data by Vitalware always uses the full data value, that is the complete text as displayed when the `All Languages` option is selected. Data is always stored in the order defined by the Supported Registry entry (page 6). When data is displayed however, Vitalware looks at the current Language Settings (specified on the Language tab of the Options box (page 17)) and adjusts the data to reflect those settings: each control makes the necessary text adjustments when displaying their data.

The following controls allow the display and editing of data in a single language:

- Single value input controls (RichEdit), e.g. text fields
- Single value selection controls (ComboBox)
- Multi-value input controls (LinkGrid)
- List View (ListGrid)
- Shortcuts Panel (Shortcuts)

The following functionality applies for data entry in a single language:

- When editing data in a single language, it is not possible to enter data for other languages.

  In other words, if `All Languages` has not been selected as the data language setting, it is not possible to enter the language delimiter followed by more text: if the language delimiter followed by more text are entered, they are removed. For example, if the data language in an English/French system is set to English and `hat;:;chapeau` is entered into a field, `;:;chapeau` will be removed by the system, leaving `hat` in the field.

  In the case where `All Languages` is selected for the data language, any language delimiters and text exceeding the number of supported languages will be removed. For example, if `hat;:;chapeau;:;hut` is entered into an English/French system, `;:;hut` will be removed as only two languages are supported.
- Text entered into a control only affects the current language. If a field contains the data `hat;:;chapeau` and the data language is set to English, then the field's control will display the text `hat`. Any changes made to the text only affect the English part of the data. For example, if `hat` is changed to `bonnet` and the record is saved, the field's data would now be `bonnet;:;chapeau`. If the field has a Lookup List associated with it, the user making the change will be prompted to accept the new entry `bonnet;:;chapeau`.

  Clearing the contents of the field only clears the text for the data language selected. For example, if a field contains `hat;:;chapeau` with the data language set to English, then the field's control will display `hat`. If the field is cleared and the record saved, the field's data will be `;:;chapeau` as only the English component was removed. If the Show First Filled Registry setting is enabled (page 13), any French values will be displayed in grey (by default) if the English text is removed.
- The **Edit>Clear** command may be used to remove the value for a field regardless of the data language selected. As mentioned in the previous point, removing the text from a control only clears the value for the current data language. In some situations it may be necessary to remove the contents for all languages. The **Edit>Clear** command provides this functionality. A keyboard shortcut of `Ctrl+Del` is also available.

- When the data for a given field is saved, any trailing language delimiters are removed.

  For example, if there is a data language setting of `All Languages` on an English/French system and `hat;:;` is entered, the trailing language delimiter will not be stored in the system. The control may still display the delimiter if the Show Trailing Delimiters Registry entry (page 10) is enabled.

Copy and Paste

Copy and Paste provides a convenient mechanism for moving data between controls.

The following functionality applies when copying and pasting data:

- When copying text, only the text displayed on the screen is placed on the clipboard. In other words, the Data and Display Order settings (specified on the Language tab of the Options box) determine what data is copied.
- The previous point applies for all controls. For example, in an English/French system displaying French text only, selecting a line in a grid and pasting it into another row (or grid) only copies the French text. Any English text will not be copied into the new row.
- When records are copied from List mode into a third party application (e.g. Excel), the data pasted will reflect the current Language Settings. In particular, when copying records into a third party application and then importing them back at a later time, it is vital that `All Languages` is specified and that the Display Order is the same as the System Order, which is easily checked on the Language tab of the Options box (page 17).

Searching

The data for a given field is stored in Vitalware in the order defined by the Supported Registry entry (page 6). The language delimiter, as defined by the Delimiter Registry entry (page 8), is stored between the text for each language, and any trailing language delimiters are removed. Hence the value stored is a single string consisting of all the language values.

The following functionality applies to searching:

- When entering values as search terms, the Data language setting (specified on the Language tab of the Options box) is ignored. Search terms may be entered in any language and Vitalware will locate all records that match the terms entered regardless of the language in which the term appears. For example, in an English/French system, if a field contains the data:

  `cellulose tape;:;a ruban de cellulose`

  entering a search term of `tape` will return the record regardless of the Data language setting. Similarly, entering a search term of `ruban` will also return the record. A search term of `cellulose` will find the record and highlight both the English and French word `cellulose` as matching.
- It is not possible to search for a term in a single language only using the standard query tabs.

  If a user has the `daEditQuery` operational privilege (see Operations Registry entry in the Vitalware Help), it is possible to adjust a query statement to restrict a search to a given language. For example, if searching for all records in a module in an English/French system where `cellulose` is mentioned in English but not in French, the required query statement is:

```
select all
from emodulename
where true and
(
   fieldname contains 'cellulose'
)
and not
(
   fieldname like '*;:;*'
)
```

  In other words, we want to find all records where *fieldname* has the word `cellulose` but does not have the string `;:;` in it. Since the data is stored as English followed by French and trailing language delimiters are removed, the query looks for records that contain `cellulose` without a language delimiter, and as English is the first language, it is only searching English data. It is important to note that the query is based on the data occurring in the order defined by the Supported Registry entry (page 6).

  Assume we want to find all records in a module where `cellulose` is mentioned in French but not in English. In this case we want to find the word `cellulose` after a language delimiter. The required query statement is:

```
select all
from emodulename
where true and
(
   fieldname contains 'cellulose'
)
and
(
   fieldname like '*;:;*cellulose*'
)
and not
(
   fieldname like '*cellulose*;:;*'
)
```

In other words, find the term `cellulose` somewhere after a language delimiter and also where `cellulose` does not appear before a language delimiter. In general, the query `like` operator can be used, in combination with the `not` operator to restrict which parts of a field's text are matched.

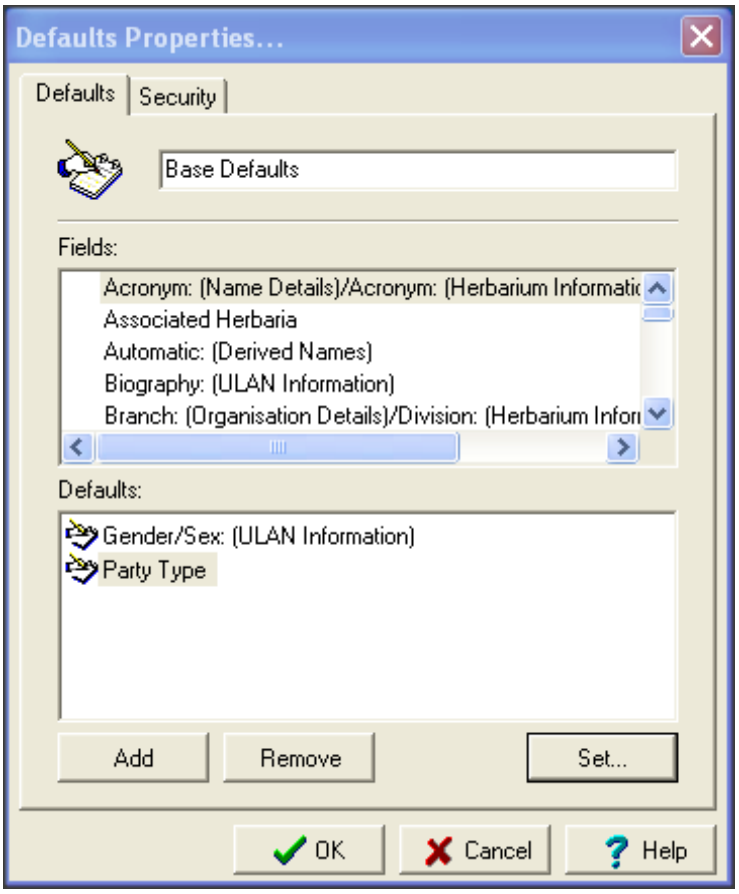
If you require further help with building the correct query, please contact Vitalware support staff.

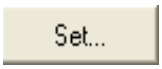
Default values

Vitalware allows users to set default values for searching and also for record creation. In the case of record creation the default values are loaded before the user begins entering information. The default values loaded should contain data for all languages to ensure users can view the data regardless of their data language setting.

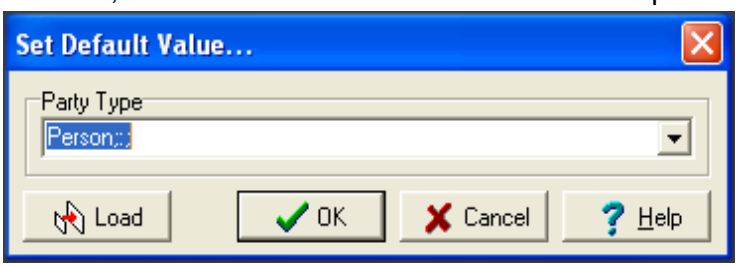
The following functionality applies to setting default values:

- The Default Properties box allows users to select fields and set default values on those fields:



Clicking 

displays the Set Default Value box. The layout of the box depends on the control type selected. For example, the *Party Type* field in the Parties module is a drop down list for data entry. When setting a default value for this field, the Set Default Value box contains a drop down list:



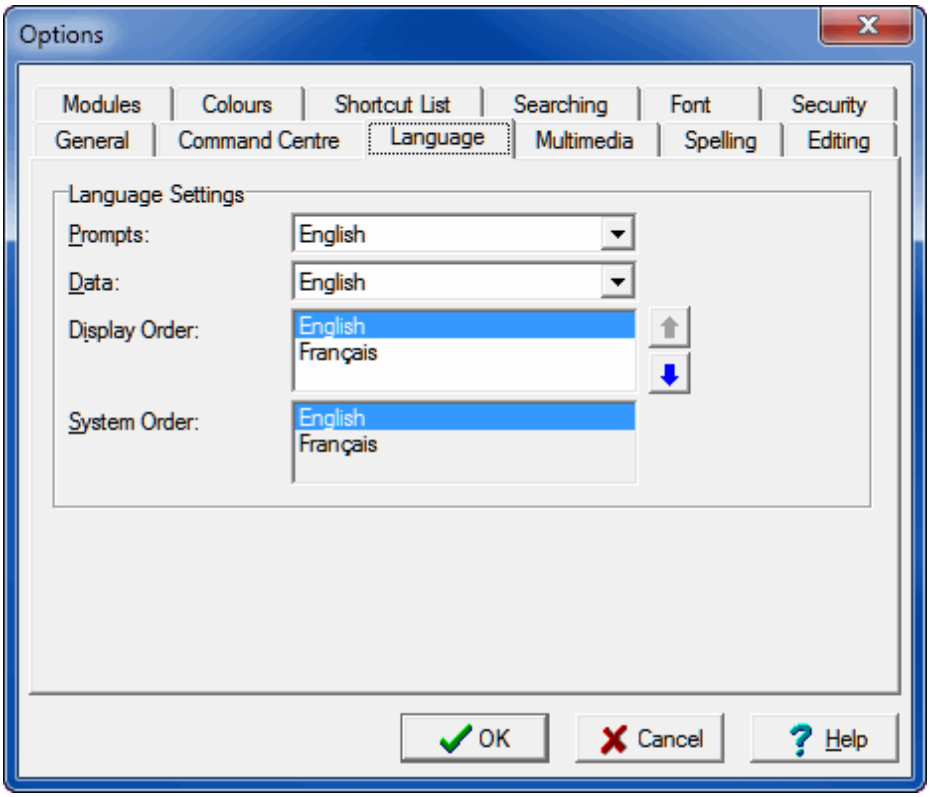
The control displayed in the box operates in the same way as the control within the module, except that the data for all languages is displayed. The data is displayed as defined by the Display Order setting (specified on the Language tab of the Options box). Data entered for default values should be recorded in the same order as the Display Order setting.

Lookup Lists

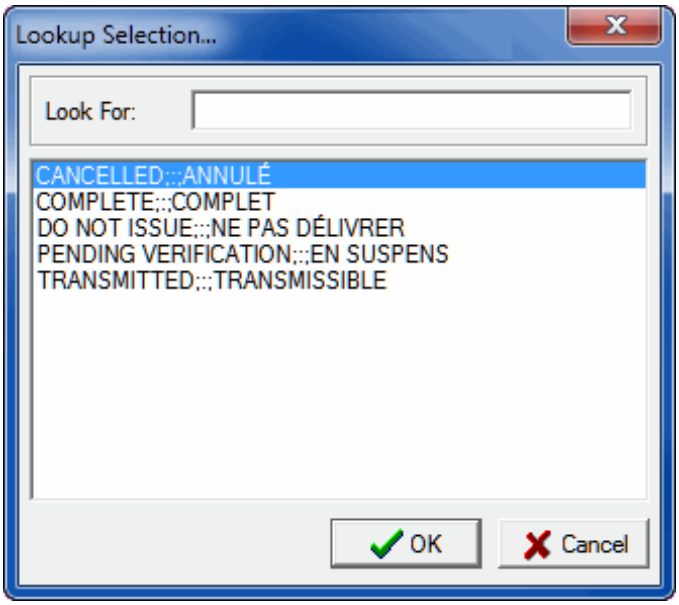
The following functionality applies to Lookup Lists:

- Lookup Lists always display their entries in all languages regardless of the current Data language setting (specified on the Language tab of the Options box). The order in which the languages are displayed is determined by the Display Order language setting (page 17).

  For example, in an English/French system in which the Display Order setting is set to English/French:



  the entries in a Lookup List will be displayed with the English text followed by the French text. The list will be sorted in alphabetic order in English:



The reason for always displaying all languages for Lookup List entries is because a term in one language may have multiple terms in another language. For example, the French word `voiture` may be translated to English as `car` but also as `carriage` (as in a train carriage). As such there are two possible Lookup List entries for `voiture`, namely:

```
voiture;:;car
voiture;:;carriage
```

If the Lookup List only displayed the French entries when French is the data language selected, it would not be clear which variant was required. By showing all languages it is possible to decide which entry is correct.

- Leading letters may be used to restrict the list of lookup entries displayed. Leading letters may be specified for any language regardless of the data language setting.

  For example, in an English/French system in which Display Order is set to French/English, the following applies:

Text Entered	Functionality
`voit`	All Lookup List entries where the French text begins with `voit` are displayed.
`;:;car`	All Lookup List entries where the English text begins with `car` are displayed.
`voit;:;carr`	All Lookup List entries where the French text begins with `voit` and the English text begins with `carr` are displayed.

The order of the terms entered is dependent on the Display Order setting (specified on the Language tab of the Options box) and is not affected by the language order setting specified in the Supported Registry entry.

- When an entry is selected from a Lookup List the value for `All Languages` is set regardless of the Data language setting (specified on the Language tab of the Options box). Hence the Lookup List entry replaces the text for all languages for the given field.
- When a new value is entered into a Lookup List field and the record is saved, the message box that prompts for confirmation of the new entry displays the Lookup List entry text in all languages in Display Order. Hence the complete entry can be checked before confirmation is given.

Auto Fill

The Vitalware Auto Fill facility allows all levels in a hierarchy to be filled automatically as soon as sufficient levels have been completed to identify a Lookup List entry uniquely.

For example, we have a hierarchy consisting of:

- Country
- State
- Town
- Postcode

When a postcode is entered, the corresponding town, state and country are populated provided there is only one combination of town, state and country for the given postcode.

When editing a single language in a multi-language environment, the Auto Fill facility handles not only the completion of hierarchies, but also the completion of Lookup Lists.

The following functionality applies to the Auto Fill facility:

- If an entry is selected from a Lookup List in a hierarchy that results in a single combination of values, then all values are assigned to the fields in the hierarchy. When the fields are assigned the entries, the data for all languages is used regardless of the current Data language setting (specified on the Language tab of the Options box). In essence the behaviour is the same as if the user had gone to all fields in the hierarchy and selected a Lookup List entry. All levels that currently contain a value are updated with the value from the Lookup List entry. This means that the text in the Lookup List entry replaces the entered text, allowing sites to control the layout of the entry (e.g. text is upper case, or leading upper case, etc.).
- If entering a value in the current Data language setting would result in a single Lookup List entry matching, the complete value for all languages replaces the single language when the control is exited. For example, if we have a Lookup List that contains the entry:

  `hat;:;chapeau`

  and the Data language is set to French, then entering the term `chapeau` and exiting the field will result in the value `hat;:;chapeau` being stored in the field. The substitution will only occur if there is no other Lookup List entry for the French word `chapeau`. If another entry exists, or if no entry exists, the text entered is left unchanged. In other words, if the text for a Lookup List entry is entered in the current Data language, the values for all other languages will be added provided a single Lookup List entry matches the text entered. Where a single match exists, the text entered is replaced with the text from the Lookup List entry, allowing the layout of the entry to be determined by the Lookup List entry.

Spell checking

The following functionality applies to spell checking:

- When spell checking is invoked, only the Data language(s) selected are checked. For example, if the Data language has been set to French in an English/French system, only French text will be spell checked. If the data language is set to `All Languages`, then each language is checked in the order defined by the Display Order setting (on the Language tab of the Options box): if text is set to display as French/English, all French text will be checked first, followed by all English text.

Global Replace

The Vitalware Global Replace facility is used to perform changes over a number of records. It provides a mechanism for applying the same update to a selected set of records. The Global Replace facility works by applying a user supplied string or pattern against a particular field, and where matches occur the matching text is replaced with another user supplied value. The important point here is that the string or pattern entered by the user to match against is compared to the data for a given field as it is stored on the server, that is as it is specified in the Language|Supported Registry entry (page 6): any string or pattern supplied for replacement must therefore be in this same order. In order to enforce this restriction, the following functionality applies to the Global Replace facility:

- The Global Replace facility is disabled unless `All Languages` is selected for the language Data setting: that is, the Replace command cannot be invoked when viewing data in a single language.

  The reason for the restriction is that any string or pattern used to find text to replace is searched for in the complete field value, not just the current Data language. For example, to replace the word `total` with `aggregate` in an English/French Vitalware system, the Global Replace facility will locate all instances of `total` and replace them with `aggregate` regardless of where in the data the text appears. Hence, where `total` appears in the French text, it is replaced with the English word `aggregate`. This means that a user could change values in a language other than the one they are viewing currently. In order to avoid this issue, the Global Replace facility is not available when viewing a single language.
- The Global Replace facility is disabled when `All Languages` is selected for the Data language setting and where the Display Order language setting is not the same order as that defined by the Supported Registry entry (which is displayed in the System Order field on the Language tab of the Options box (page 17)).

  The reason for the restriction is that it may not be possible to translate strings or patterns entered with arbitrary display order settings to values suitable for replacing. For example, in an English/French system with a Display Order of French/English, the string/pattern `total;:;` cannot be converted to a single pattern for applying to the English/French data store in Vitalware. The string/pattern means to look for the word `total` where it is the last French word (as it has a trailing language delimiter entered). One possible solution is to translate the string/pattern as `French$` so that French is the last word in the data. However such a string/pattern would match the following data:
  - `French`
  - `English;:;French`

  In other words English data may be modified where only French changes were intended. There are a number of other patterns that are also problematic as they cannot be converted to a pattern suitable for use with English/French data. In order to avoid the accidental updating of values not intended to be updated, the Global Replace facility is only enabled for users who are viewing `All Languages` and the language Display Order is the same as that defined by the Supported Registry entry.

Examples of Global Replaces in a multi-language environment are provided in Global Replace: Pattern matching examples (page 36).

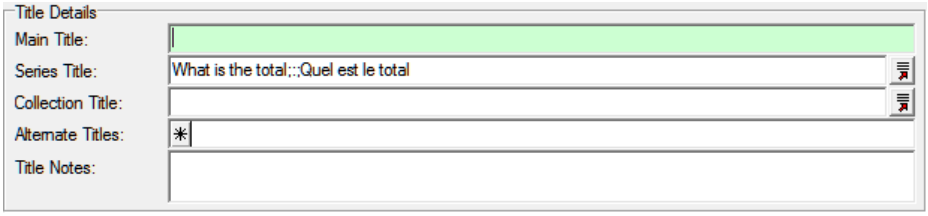
Global Replace: Pattern matching examples

These examples assume a Dual language environment in which data is stored on the server as English and then French.

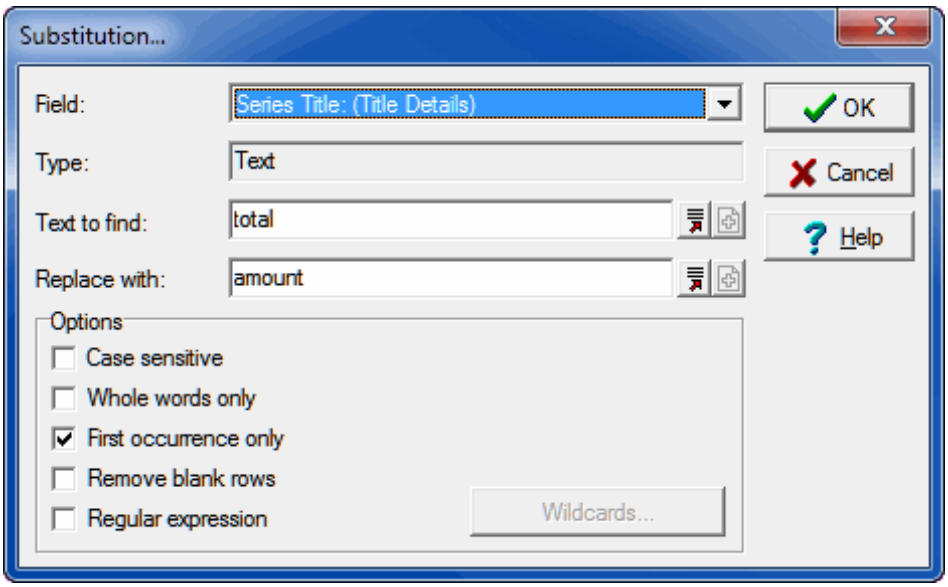
See Global Replace for details about defining a substitution; see Wildcards in a Global Replace for a description of the wildcard symbols used in these examples.

1. **Objective** — Search the *Series Title: (Title Details)* field for the first occurrence of the word `total` and replace it with the word `amount`.

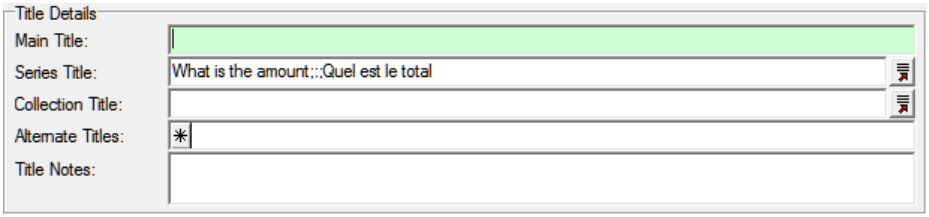
Original data



Substitution



Result



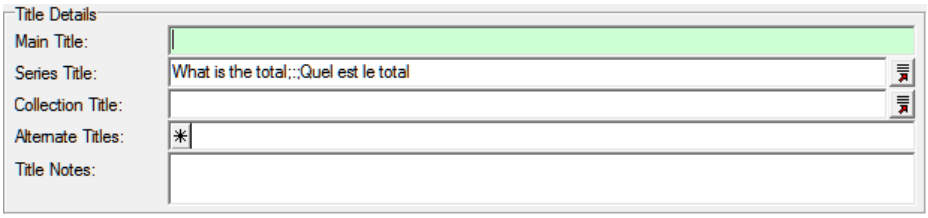
Description — Selecting **First occurrence only** tells Vitalware to search for the first occurrence of the word `total` and replace it with the word `amount` and then stop.

2. **Objective**

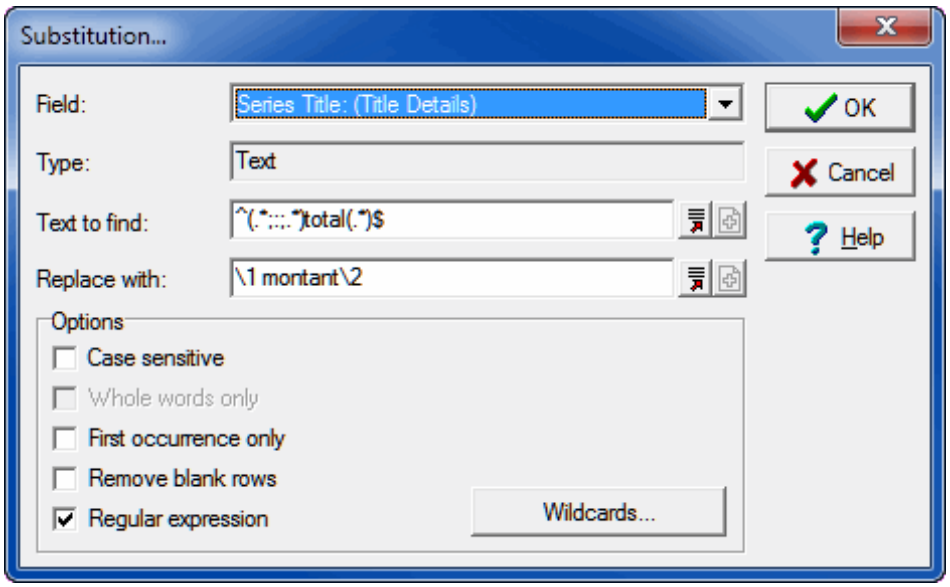
This example demonstrates "greedy" patterns in operation. A "greedy" pattern is one which matches the longest string possible. `(.*)` is an example of a "greedy" pattern, although more generally, use of the `*` wildcard will initiate a "greedy" pattern match. When the *Text to find* includes a "greedy" pattern, pattern matching in the target field will find the longest matching sequence.

Here we search the *Series Title: (Title Details)* field for the French word `total` (that is, `total` appears after the language delimiter) and replace it with `montant`.

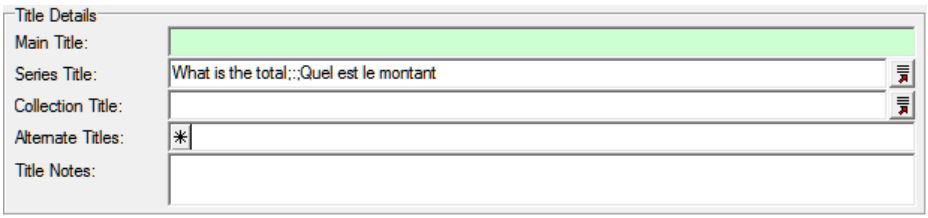
Original data



Substitution



Result



Description

Text to find

```
^(.*;;;.*)total(.*)$
```

Entering the text to find (`total`) and expressions between `^` (which indicates the beginning of the field) and `$` (which marks the end of the field indicates that we're looking for (and replacing) the entire contents of the field.

There are two expressions in this example, marked by the brackets:

- `(.*;;;.*)`
  and
- `(.*)`

Recall that the `.*` pattern is a greedy pattern and that it will therefore match the longest possible sequence:

`^(.*;;;.*)`, which is Expression 1, matches everything from the last occurrence of the word `total` to the beginning of the field, i.e.:

```
What is the total;;;Quel est le
```

Breaking this down further:

- the first `.*` matches everything from the `;;;` to the beginning of the field, i.e.:
  ```
  What is the total
  ```
- the second `.*` matches everything from the last occurrence of `total` up to the `;;;`, i.e.:
  ```
  Quel est le
  ```

Putting these three components together (the data that matches the first `.*`, the `;;;` and the data that matches the second `.*`), Expression 1 is:

```
What is the total;;;Quel est le
```

In theory, `^(.*)` would have the same result as `^(.*;;;.*)`, but with this difference: in the second case, the pattern being matched must include `;;;`

The second expression `(.*)` then matches everything from the end of the field up to the word to be replaced, which is the last occurrence of `total`. As it happens there is nothing after the word `total` but if there was, we would have matched it.

Replace with

```
\1 montant \2
```

In this usage, the `\` does not perform an escape function (as in `\?` which means treat the `?` as a question mark, not as a character with a special function).

`\1` refers to the first expression in the *Text to find* field: `(.*;;;.*)`

`\2` refers to the second expression in the *Text to find* field: `(.*)`

In this case Vitalware is required to locate and replace:

```
^(.*;;;.*)total(.*)$
```

with:

```
(Expression 1) montant (Expression 2)
```

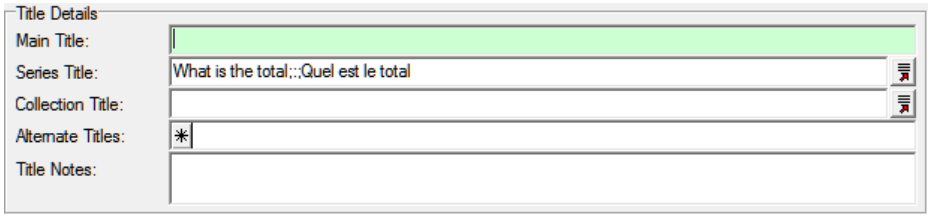
In other words:

```
(What is the total;;;Quel est le) montant( )
```

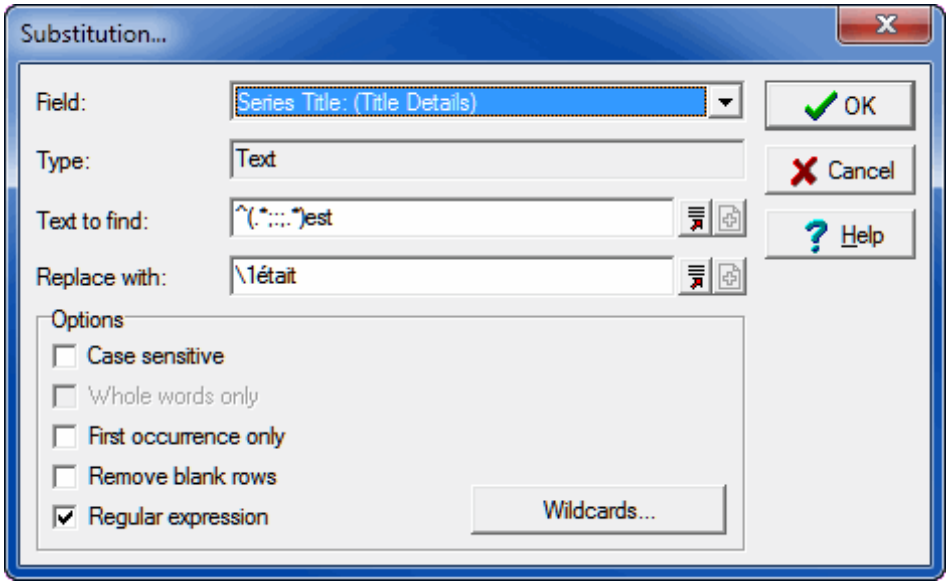
3. **Objective**

Search the *Series Title: (Title Details)* field for the French word `est` and replace it with `était`.

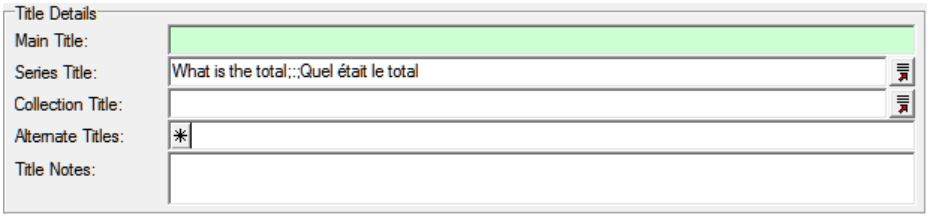
Original data



Substitution



Result



Description

Text to find

```
^(.*;;;.*)est
```

The first thing to notice is that we are not replacing the entire content in the field as our *Text to find* does not sit between `^` and `$`.

There is only one expression and it affects data from the word to be replaced (`est`) to the start of the field (again this involves a "greedy" pattern which matches the longest possible sequence).

Our *Text to find* tells Vitalware to match everything from the last occurrence of `est` to the beginning of the field. Breaking this down further:

- the second `.*` matches everything up to the `;;;`, i.e.:
  `Quel`
- the first `.*` matches everything else up to the beginning of the field, i.e.:
  `What is the total`

Putting these three components together (the data that matches the first `.*`, the `;;;` and the data that matches the second `.*`), Expression 1 is:

```
What is the total;;;Quel
```

In theory, `^(.*)` would have the same result as `^(.*;;;.*)`, but with this difference: in the second case, the pattern being matched must include `;;;`

Replace with

```
\1était
```

`\1` refers to the first expression in the *Text to find* field: `(.*;;;.*)`

In this case Vitalware is required to locate and replace:

```
^(.*;:;.*)est
```

with:

```
(Expression1) était
```

Anything else in the field after `est` is left alone.

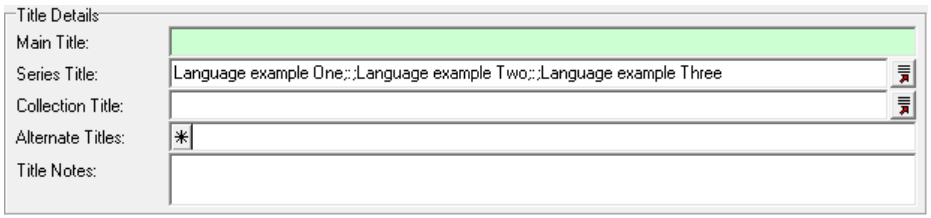
In other words:

```
(What is the total;:;Quel) était le total
```

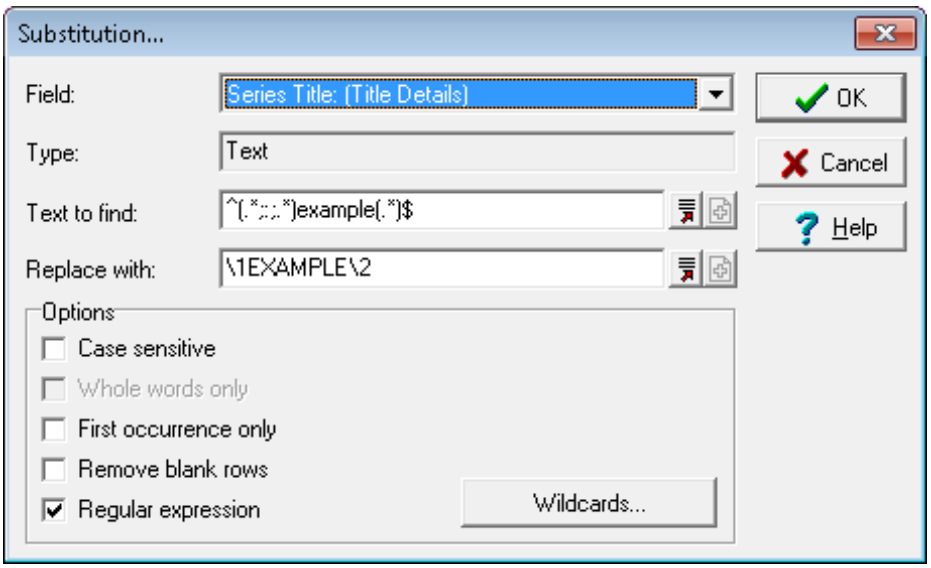
4. **Objective**

This example demonstrates "greedy" patterns in operation again in a more complex setting.

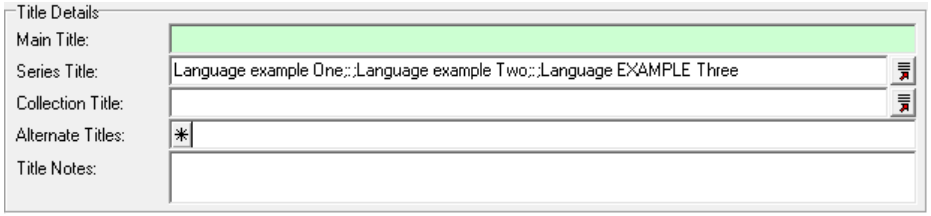
Original data



Substitution



Result



Description

If we weren't aware that use of the "greedy" pattern `(.*)` processes a match from the end of the field forwards, we might expect that this substitution would replace the second `example`, giving us:

```
Language example One;:;Language EXAMPLE Two;:;Language example Three
```

Instead, the following occurs:

Text to find

Starting from the end of the field `(.*)`, which is Expression 2, matches everything up to `example`, i.e.:

```
Three
```

As a "greedy" pattern `(.*)` matches the longest string possible, which in this case is everything from the end of the field up to the last occurrence of `example`, which is the text to be replaced with `EXAMPLE`.

`^(.*;:;.*)`, which is Expression 1, now matches everything from this point to the beginning of the field, i.e.:

```
Language example One;:;Language example Two;:;Language
```

Breaking this down further:

- the second `.*` matches everything up to the last occurrence of example, i.e.:
  ```
  Three
  ```
- the first `.*` matches everything else, i.e.:
  ```
  Language example One;:;Language EXAMPLE Two; Language
  ```

In theory, `^(.*)` would have the same result as `^(.*:;:.*)`, but with this difference: in the second case, the pattern being matched must include `;:;`

Replace with

```
\1EXAMPLE\2
```

`\1` is the first expression, which equates to:

```
Language example One;:;Language example Two;:;Language
```

\2 is the second expression, which equates to:

```
Three
```

Which gives us:

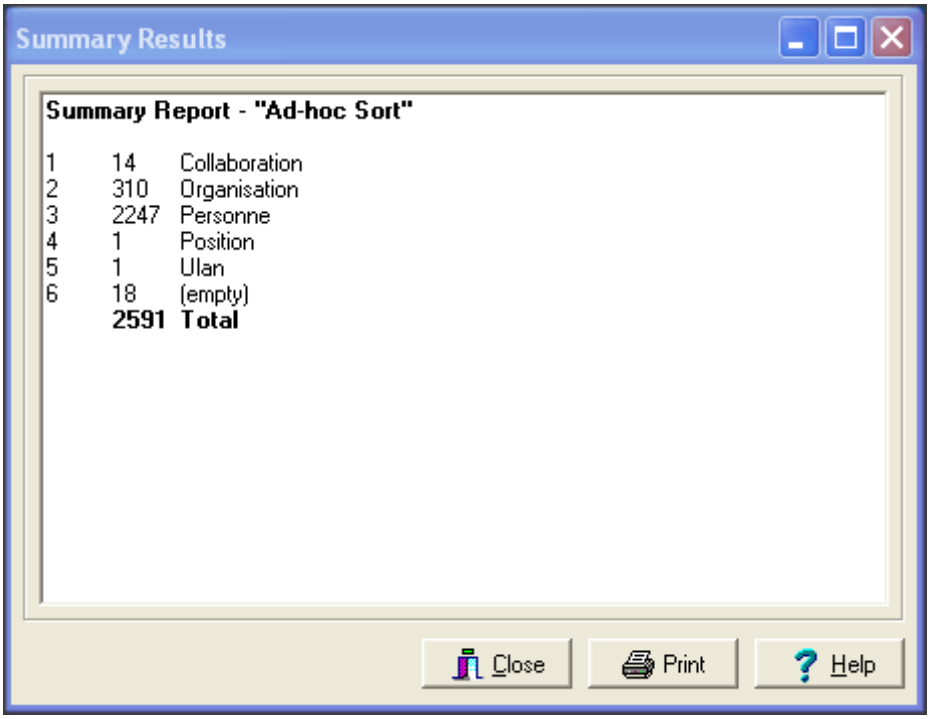
```
Language example One;:;Language example Two;:;Language
EXAMPLE three
```

Sorting

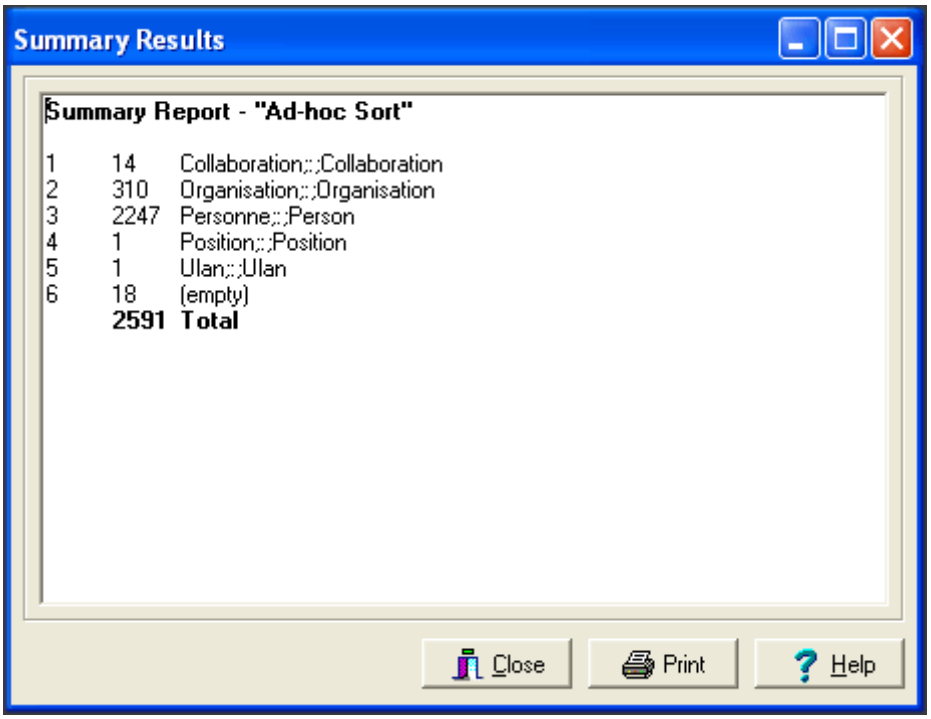
The Vitalware Sorting facility allows a set of matching records to be sorted based on values in the records. After the sort is complete a summary of each term and the number of times it appears may be displayed.

The following functionality applies when sorting records:

- The sorting of multi-lingual fields complies with the setting of the Data language and Display Order options (specified on the Language tab of the Options box). If the Data language option is set to a specific language rather than `All Languages`, the sorting will sort by the language selected and the summary will display terms for the language selected only. For example, in an English/French system with the Data language set to French, the data will be sorted based on the French values only and the summary will list French values only:



If the Data language is set to `All Languages`, the Display Order setting is used to determine in what order the language values should be sorted. For example, if the Display Order option is set to French then English, the sorting will occur with French terms first and where two French terms are the same, the English terms will then be used:



- Where a value in one language has multiple values in another language, the summary count for the term will vary depending on the Data Language option selected. For example, the French term `voiture` could have the following English translations:

  ```
  voiture
  voiture;;;car
  voiture;;;carriage
  ```

  Assume the following number of records contain each value:

  ```
  voiture - 25 records
  voiture;;;car - 12 records
  voiture;;;carriage - 3 records
  ```

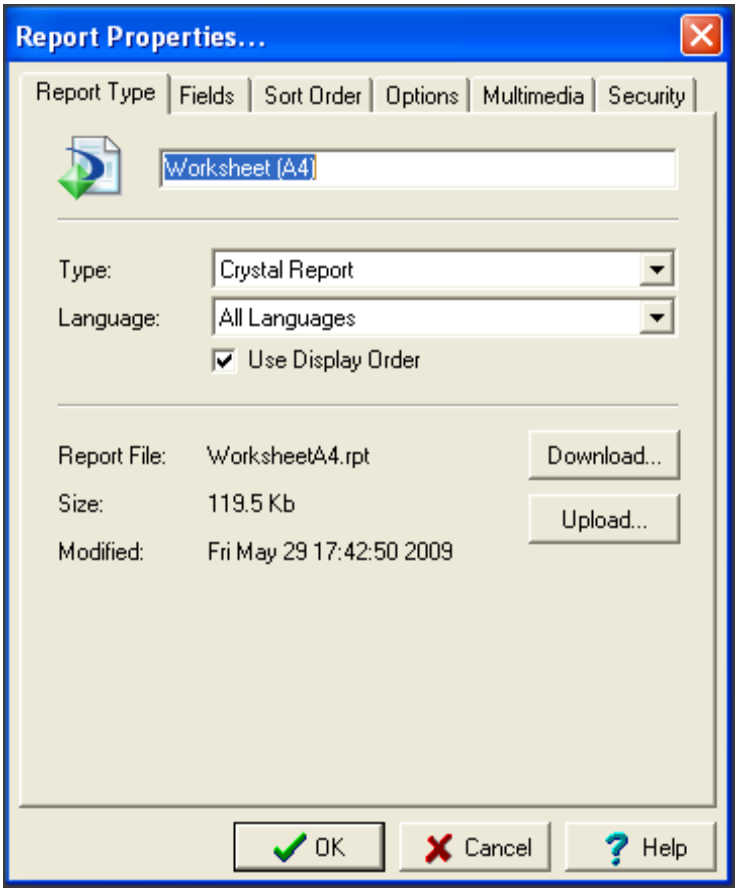
  If the Data language is set to French and a summary is produced, the count for the term `voiture` will be the sum of all records containing `voiture, voiture;;;car` and `voiture;;;carriage`. Based on our record counts the number of records with the term `voiture` would be 40 (25 + 12 + 3). If the Data language is set to `All Languages` and a summary is produced, the summary will display the breakdown of each term along with its associated count. In all cases the total number of records will be the same.

Reports

Vitalware reports are generally designed to display data in a specific language. If a report consists of French prompts, then French data should be displayed. In fact, French data should be displayed regardless of the settings for the Data language and Display Order options (specified on the Language tab of the Options box).

The following functionality applies when producing reports:

- When defining the attributes for a report, the reporting Language may be specified. The language selected determines the data to be embedded in the report, regardless of any user settings:



- If a specific language is selected, the values output for the report will be based on that language. If `All Languages` is selected, then the Use Display Order check box determines the order in which the language data is displayed:
  - If unchecked, the data is output in System Order as defined by the Supported Registry entry.
  - If checked, the data will be output based on the current setting of the Display Order option.

  In most cases the Use Display Order option is not checked since this results in the data being embedded in a predictable order. The main exception is a report designed for use in Page View mode. In this case the Use Display Order option should be turned on so that the data displayed matches the user's current settings.

- If a Sort Order is defined for the report, the sorting uses the same options as those defined for the report. In other words, the report Language selected and the Use Display Order setting define what values are used when sorting the records for the report. It is not possible to define a sort order independently of the report language settings (it is hard to imagine a case where this would be desirable).

Page View mode

Page View mode provides a mechanism for viewing HTML based reports in the Vitalware client.

The following functionality applies when in Page View mode:

- Since Page View mode uses a report registered in Vitalware, the data shown reflects the settings of the report not the settings for the Data and Display Order options (specified on the Language tab of the Options box). This allows reports to be designed to reflect how the data is to be displayed regardless of the current Language Settings.
- The Default Page View report always displays all languages in System Order, which is easily checked on the Language tab of the Options box (page 17). The default report provides a useful way of showing the text for all languages regardless of the current Language Settings.

Scheduled Exports

The Scheduled Export facility allows data to be exported from Vitalware on a regular basis. The data is generally fed into third party systems.

The following functionality applies when exporting data:

- When defining Scheduled Export settings, it is not possible to set the language to be exported. In all cases, all languages will be exported in System Order as defined by the Supported Registry entry, which is easily checked on the Language tab of the Options box (page 17).
- If a data export is required for a particular language, then an export filter is required. The filter can parse the data and rewrite it with the language required. Using an export filter allows the data to be manipulated on a per column basis, enabling the data to be massaged to suit any third party system.
- If a Sort Order is defined for the export, the data is sorted based on the complete contents of the column.

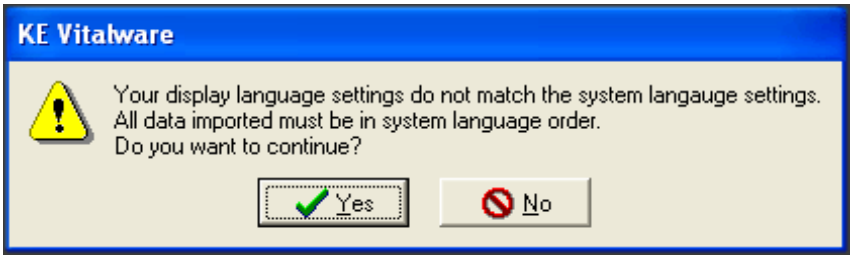
Imports

The Import facility allows data to be taken from an external source in either CSV (Comma Separated Values), Tab (Tab Delimited Values) or XML (eXtensible Markup Language) format and loaded into Vitalware.

The following functionality applies when importing data:

- When importing data the user's Data language and Display Order options (specified on the Language tab of the Options box) are ignored. The data value can contain all languages. The language order is in System Order as defined by the Supported Registry entry, which is easily checked on the Language tab of the Options box (page 17).

  In other words, the way that multi-lingual values are specified for data import is fixed, regardless of any user settings. This allows the same data file to be imported by different users with different settings but have the same records with the same language order created.
- If the Display Order of languages does not match the System Order of languages, or if a single language is displayed, and an import is commenced, a message displays indicating that all data imported must be in the System Order. The user may choose to abort or proceed:



Index