Vitalware Documentation

Release Notes: Vitalware 2.2.01

Document Version 1

Vitalware Version 2.2.01

www.kesoftware.com

Contents

Here you will find collected together the Release Notes for Vitalware 2.2.01, alongside all documents referenced in the notes. These release notes and documents are also available on the KE 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, e.g. Multi-group Support, has its own internal numbering:

Release Notes: Vitalware 2.2.01
Release Date: 15 June 2011

Requirements

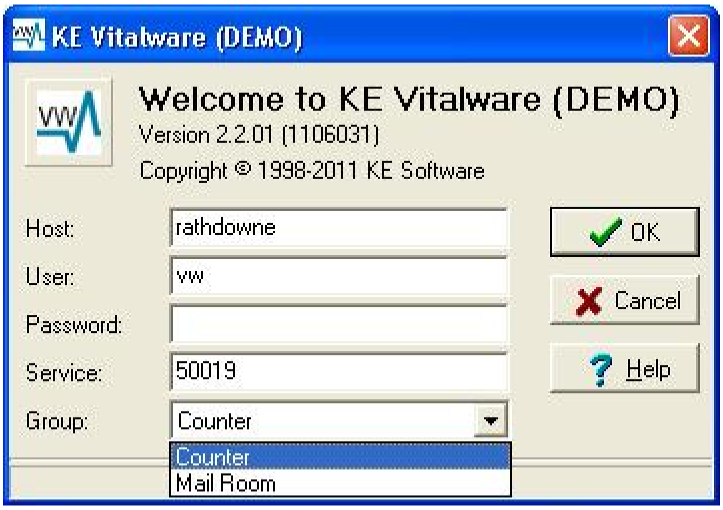
- For Windows 2000, XP, 2003, Microsoft Windows Services for UNIX (version 3.5)
- Texpress 8.3.001 or later
- TexAPI 6.0.003 or later
- Perl 5.8 or later

New Features

- Multiple Group Support: Multiple Group support allows a user to be a member of more than one group. The Vitalware Registry entry that defines the group into which a user is placed has been extended to allow a semicolon separated list of groups to be specified. For example:

  `User|vw|Group|Counter;Mail Room`

  The above entry indicates user `vw` is a member of two groups, `Counter` and `Mail Room`. If a user is a member of multiple groups, the Vitalware Login screen provides a drop-list with the groups that the user is a member of:

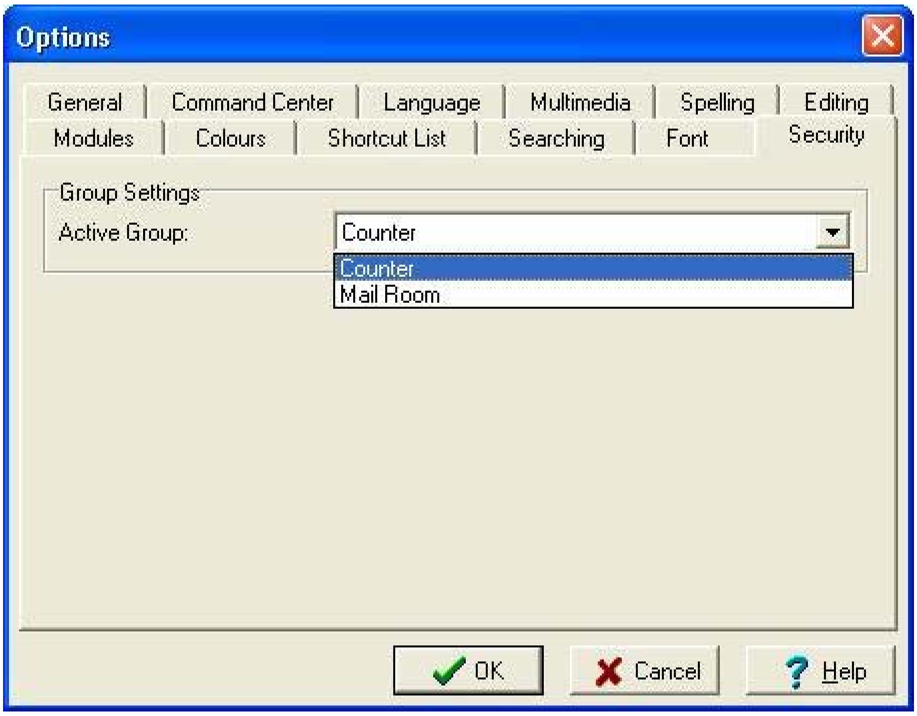
  

  KE Vitalware (DEMO)

  Welcome to KE Vitalware (DEMO)
  Version 2.2.01 (1106031)
  Copyright © 1998-2011 KE Software

  Host: rathdowne
  User: vw
  Password:
  Service: 50019
  Group: Counter

  The group selected becomes the *active* group and is used to determine permissions until another group is selected. The Security tab in the Options dialogue allows another group to be chosen:

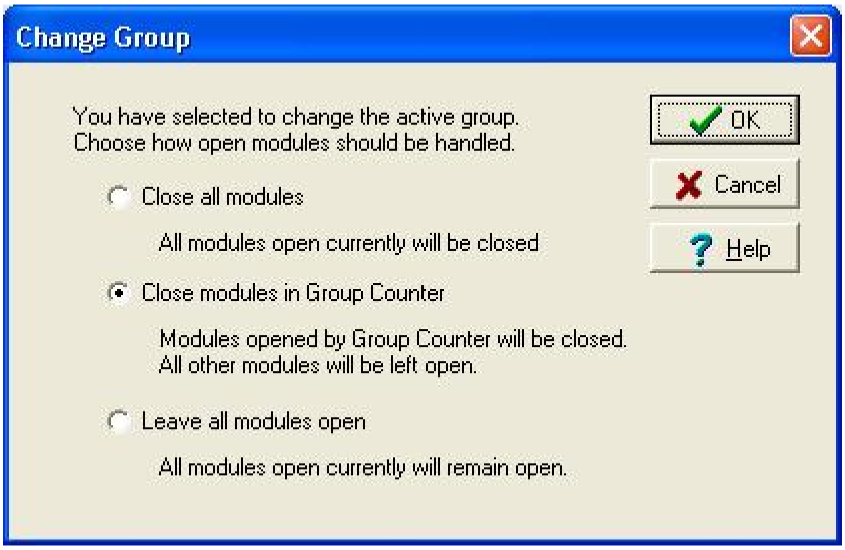


Group Settings

Active Group: Counter

Counter
Mail Room

When the active group is changed, a dialogue displays allowing the user to indicate how opened modules should be handled:



You have selected to change the active group.
Choose how open modules should be handled.

○ Close all modules

All modules open currently will be closed

● Close modules in Group Counter

Modules opened by Group Counter will be closed.
All other modules will be left open.

○ Leave all modules open

All modules open currently will remain open.

Any module left open will remain in its existing group. Hence it is possible to have two (or more) modules open, each in a different group. The permissions used for a given module are dictated by the group in which the module exists. The module's group is displayed in the status bar at the bottom of the module window.

A complete description of the support for multiple groups can be found in the Multi-group Support documentation.

- **Encrypted connections**: Encrypted connections support allows all data transmissions between the Vitalware client and server to be encrypted. As part of the login process, X509 based certificates are exchanged allowing Vitalware clients to ensure they are connecting to the correct Vitalware server. TLS v1.0 (Transport Layer Security) is used to provide the encryption. System Administrators may choose the encryption cipher used, allowing different levels of compression and security to be configured. The Vitalware server may be configured only to accept encrypted connections, otherwise encryption is optional.

  Encrypted connections are also supported for:

  - Java connections (TexJDBC)
  - C/C++ connections (TexAPI)
  - perl connections (texapi.pm)

  A complete description of the support for encrypted connections can be found in the Encrypted Connections documentation.

- **Read-only Modes**: A new Registry entry allows System Administrators to make part or all of Vitalware read-only. The format of the new entry at a system wide level is:

  `System|Setting|ReadOnly|`*value*
  `Group|Default|Setting|ReadOnly|`*value*
  `Group|`*group*`|Setting|ReadOnly|`*value*
  `User|`*user*`|Setting|ReadOnly|`*value*

  At a table level the format is:

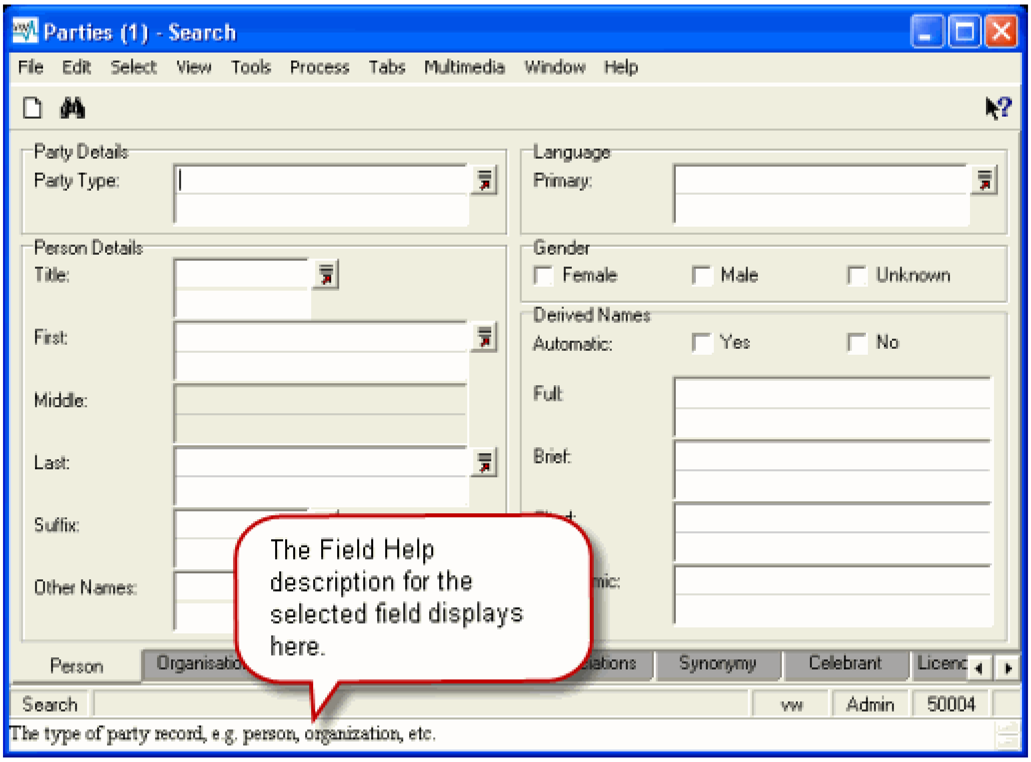
  `Group|Default|Table|Default|ReadOnly|`*value*
  `Group|Default|Table|`*table*`|ReadOnly|`*value*
  `Group|`*group*`|Table|Default|ReadOnly|`*value*
  `Group|`*group*`|Table|`*table*`|ReadOnly|`*value*
  `User|`*user*`|Table|Default|ReadOnly|`*value*
  `User|`*user*`|Table|`*table*`|ReadOnly|`*value*

  where *value* is either:

  - `true` - read-only functionality is enabled.
  - `false` - read-only functionality is disabled.

  A complete description of the support for read-only modes can be found in the Read-only Modes documentation.

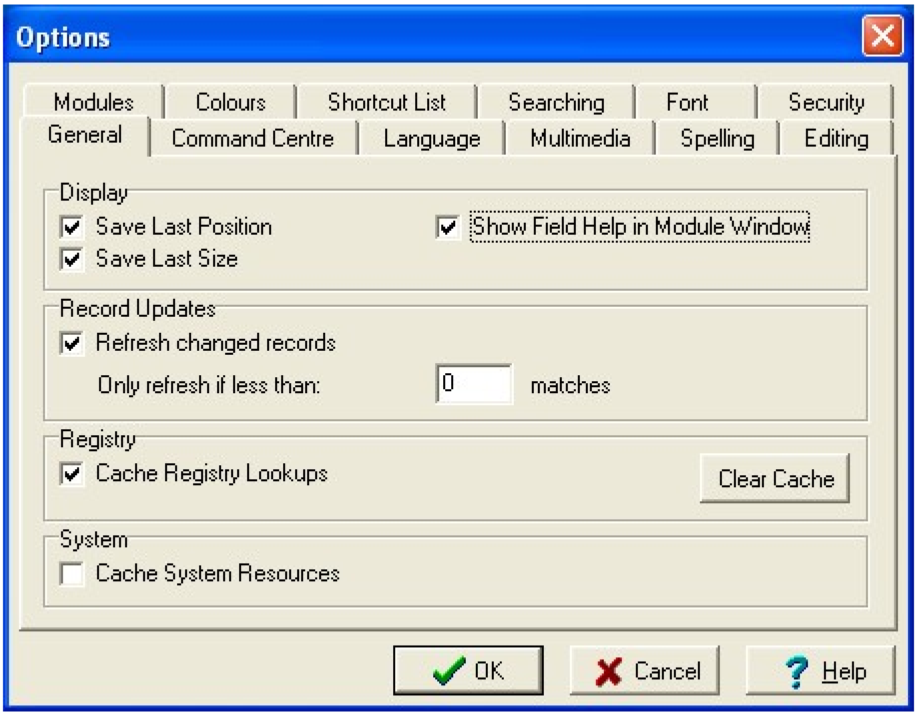
- **Field Help Display**: A new option allows the field level help for a given field to be displayed below a module's Status bar. The help display area may be re-sized to suit:



The Field Help description for the selected field displays here.

The type of party record, e.g. person, organization, etc.

The *Show Field Help in Module Window* option is located on the General tab in the Options dialogue:



Display

☑ Save Last Position ☑ Show Field Help in Module Window

☑ Save Last Size

Record Updates

☑ Refresh changed records

Only refresh if less than: 0 matches

Registry

☑ Cache Registry Lookups

System

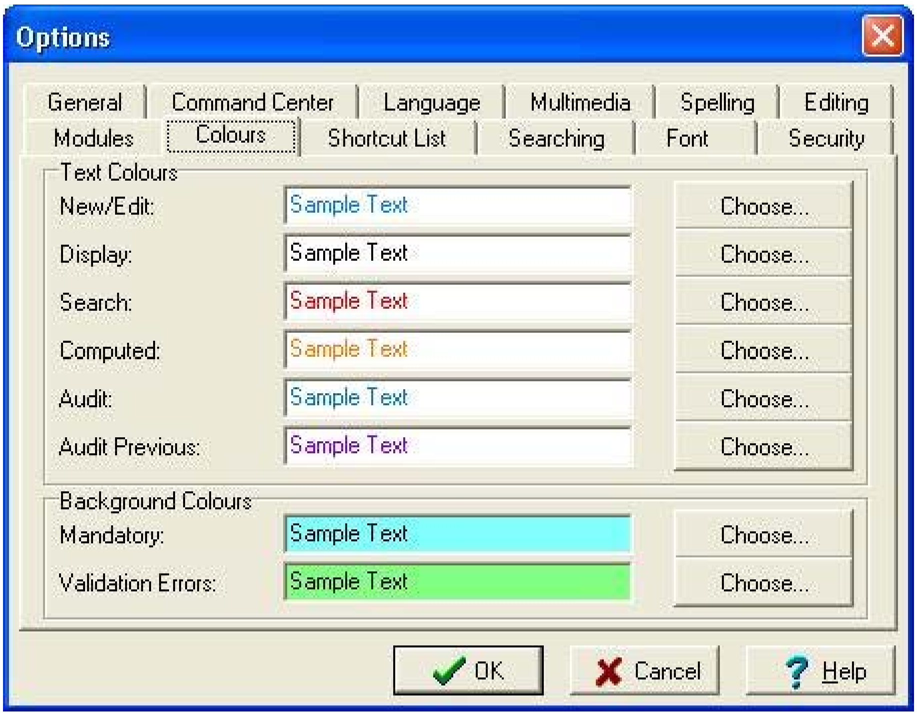
□ Cache System Resources

- Fields which fail validation may now have their background highlighted in a particular colour

  New functionality has been added which uses the following Registry setting:

  `System|Setting|Show Validation Errors|true`

  When this entry is set to `true` and the column *ValValidationFields* is present in a module, any column listed in this field will be highlighted with the background colour selected for Validation Errors on the Colours tab of the Vitalware options:



Text Colours

New/Edit: Sample Text

Display: Sample Text

Search: Sample Text

Computed: Sample Text

Audit: Sample Text

Audit Previous: Sample Text

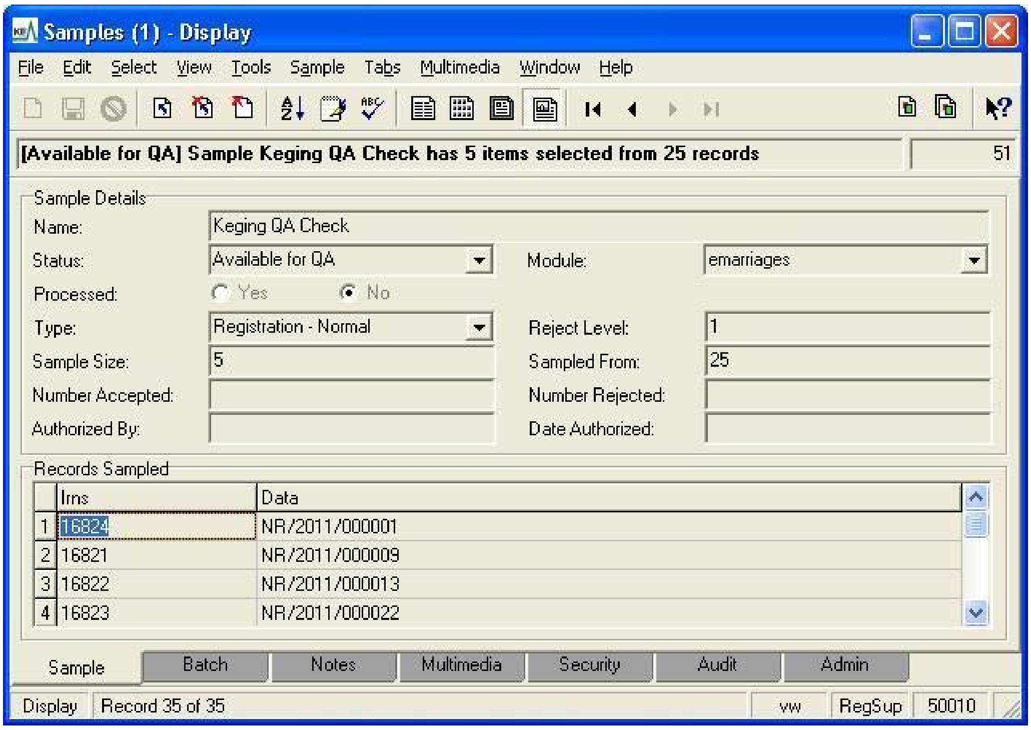
Background Colours

Mandatory: Sample Text

Validation Errors: Sample Text

- Ability to generate a sample of records from a set of records

A new Samples module has been added to record information about a sample of records. Users with sufficient privilege can monitor the progress and authorise the completion of each batch:


Samples (1) - Display
File Edit Select View Tools Sample Tabs Multimedia Window Help
Sample Batch Notes Multimedia Security Audit Admin

[Available for QA] Sample Keging QA Check has 5 items selected from 25 records 51

Sample Details

Name:	Keging QA Check		
Status:	Available for QA	Module:	emarriages
Processed:	○ Yes ● No		
Type:	Registration - Normal	Reject Level:	1
Sample Size:	5	Sampled From:	25
Number Accepted:		Number Rejected:	
Authorized By:		Date Authorized:	

Records Sampled

	Irns	Data
1	16824	NR/2011/000001
2	16821	NR/2011/000009
3	16822	NR/2011/000013
4	16823	NR/2011/000022

Display Record 35 of 35 vw RegSup 50010

The Lot Sizes Registry entry is used to configure sample options. This entry is in the format:

```
System|Setting|Lot Sizes|Record Type|Sample Type|sample config; sample
config;...
```

The sample config part of the entry defines lot sizes and conditions. Using the first sample config in the example below, if the number of records to base the sample on is 25 or less, then the generated sample for these records will contain five records. In order for this sample to be successfully processed, the sample may only contain a maximum of one error:

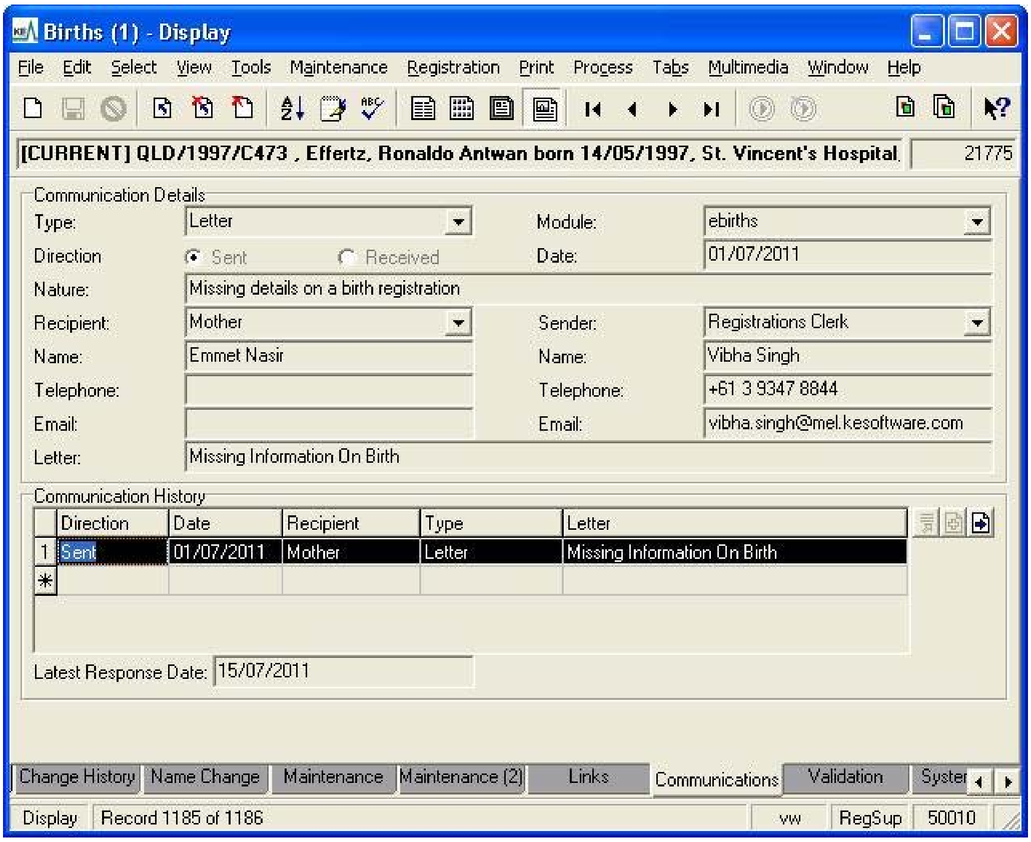
```
System|Setting|Lot Sizes|Registrations|Normal|25-5-1;90-20-2;150-32-3;280-
50-4;500-80-5;1200-125-7;3200-200-10;10000-315-13;35000-500-19
```

Functionality has been added to the POS and Registration modules to allow for generating and processing samples. A sample may be generated by selecting the **Generate Sample** menu option found under the Tools menu and processed by selecting the **Sample** menu option under the new Process menu.

See the latest version of the Vitalware Help for full details.

- Ability to attach communications directly to Registration or POS module records

A new Communications module has been created to record individual communications between the registry and its clients. An additional tab has been added to the POS and Registration module to track the communications associated with a record. New communications may be added to a record by selecting the **Communications** menu option under the new Process menu:


Births (1) - Display
File Edit Select View Tools Maintenance Registration Print Process Tabs Multimedia Window Help

[CURRENT] QLD/1997/C473 , Effertz, Ronaldo Antwan born 14/05/1997, St. Vincent's Hospital 21775

Communication Details

Type:	Letter	Module:	ebirths
Direction	● Sent ○ Received	Date:	01/07/2011
Nature:	Missing details on a birth registration		
Recipient:	Mother	Sender:	Registrations Clerk
Name:	Emmet Nasir	Name:	Vibha Singh
Telephone:		Telephone:	+61 3 9347 8844
Email:		Email:	vibha.singh@mel.kesoftware.com
Letter:	Missing Information On Birth		

Communication History

	Direction	Date	Recipient	Type	Letter
1	Sent	01/07/2011	Mother	Letter	Missing Information On Birth
*					

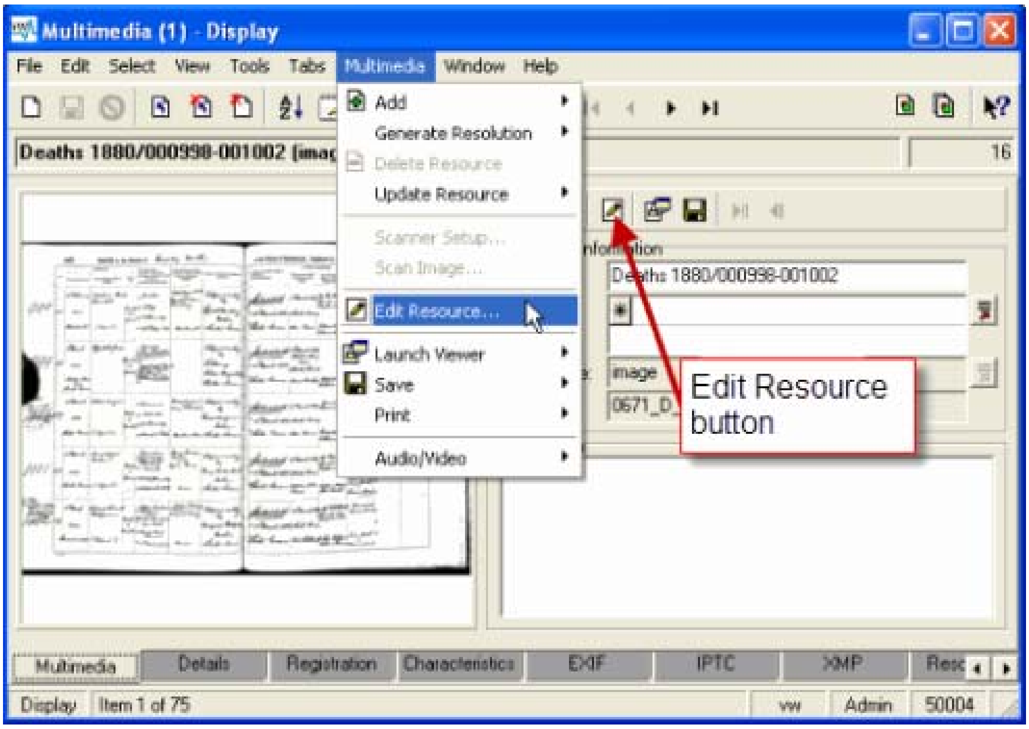
Latest Response Date: 15/07/2011

- **New Call Centre Module**

  A new Call Centre module has been added to the client, containing summary information from Registration and POS module records. This can be used to provide staff in a call centre with reduced record access to respond to first level enquiries.

Improvements

- **Edit Resource command**: A new command has been added to the Multimedia Repository allowing a user to invoke an external editor to modify the multimedia resource (image, document, etc.). The Edit Resource command downloads the resource from the Vitalware server, invokes an editor and monitors the file for any changes. It also looks for any other files created with the same name as the resource file but with a different extension (file type). When Vitalware receives focus the user is asked whether the modified file should be re-imported into the Multimedia Repository. An affirmative response saves the resource on the Vitalware server and generates any resolutions, etc. The new command streamlines the process of modifying multimedia resources:



Deaths 1880/000998-001002 (imag

16

Deaths 1880/000998-001002

image

0671_D

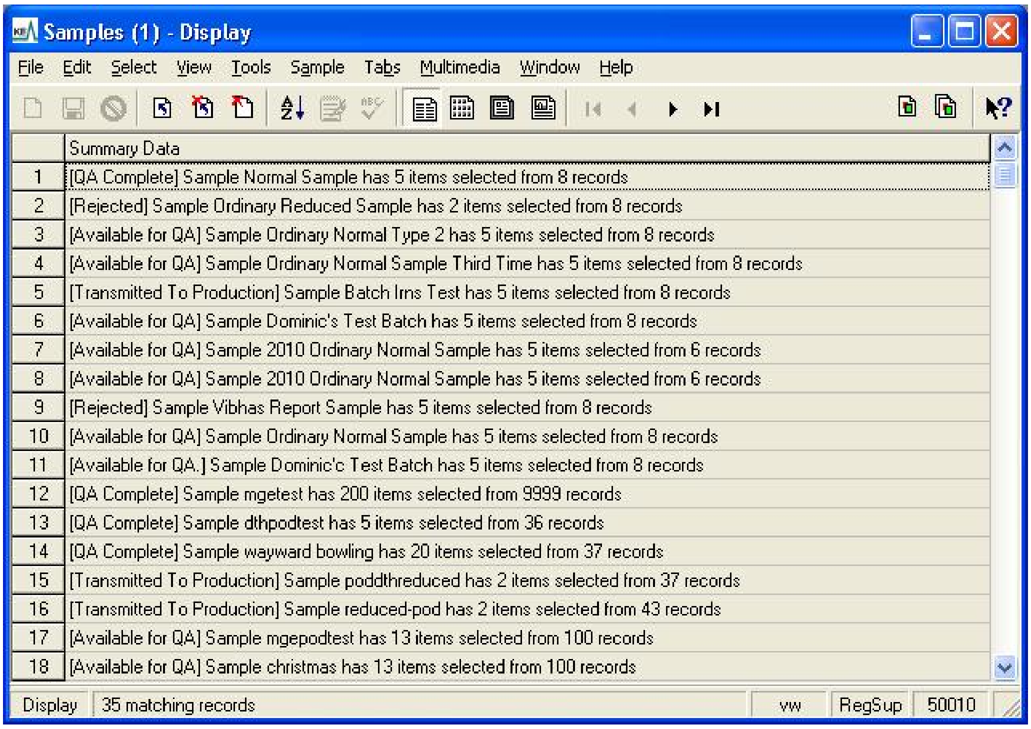
Edit Resource button

- **XMP and IPTC upgraded:** The Multimedia Repository has been upgraded to support the latest versions of the IPTC and XMP metadata standards. The standards implemented are:
  - IPTC Core Version 1.1
  - IPTC Extension Version 1.1
  - XMP July, 2010
- **Login details on a per client basis:** The Host/User/Service/Group login values are remembered on a per client basis. If an institution runs multiple clients, the details displayed in the login dialogue are the values from the last time that particular client was run, rather than the values for the last time any client was invoked.
- **Change password utility uses SSH:** The administration task allowing a user to change their password now uses SSH (secure shell) to effect the change. Previously TELNET was used, requiring the TELNET service to be enabled on the server for the utility to work. By using SSH, the TELNET service no longer needs to be enabled, rather the SSH service is required. Most sites enable SSH to allow secure access to the Vitalware server.
- **ICS support for notifications:** The Vitalware notification system, used to generate email notifications and HTML pages for upcoming activities (registrations, tasks, etc.), now builds an iCalendar (.ics) file containing upcoming dates. The iCalendar file may be imported into a large number of products, including:
  - Google Calendar
  - Apple iCal
  - IBM Lotus Notes
  - Microsoft Outlook

  Once imported, the upcoming dates, along with a detailed description, are incorporated into the user's calendar.
- **Reverse scheduling of tasks:** The Task tab functionality has been extended to allow the completion and start dates of each task to be computed by specifying the overall completion

date. The new functionality allows a user to set the date on which all tasks should be completed and have Vitalware calculate the start and completion dates for each task.

- **Modifications to vwperiodic:** The `vwperiodic` server-side command is used to run tasks on a regular basis. In particular, it is used to generate the raw data for the Statistics module. The command has been extended to allow:
  - Individual scripts to be specified, restricting the scripts executed.
  - Execution of local scripts only, via the `-l` option.
- **Read-only records in List View:** The display of read-only records in List View has been changed. Read-only records are now displayed with a grey background and black text, rather than the harder to read, grey text on a white background:


Samples (1) - Display
File Edit Select View Tools Sample Tabs Multimedia Window Help

	Summary Data
1	[QA Complete] Sample Normal Sample has 5 items selected from 8 records
2	[Rejected] Sample Ordinary Reduced Sample has 2 items selected from 8 records
3	[Available for QA] Sample Ordinary Normal Type 2 has 5 items selected from 8 records
4	[Available for QA] Sample Ordinary Normal Sample Third Time has 5 items selected from 8 records
5	[Transmitted To Production] Sample Batch Irns Test has 5 items selected from 8 records
6	[Available for QA] Sample Dominic's Test Batch has 5 items selected from 8 records
7	[Available for QA] Sample 2010 Ordinary Normal Sample has 5 items selected from 6 records
8	[Available for QA] Sample 2010 Ordinary Normal Sample has 5 items selected from 6 records
9	[Rejected] Sample Vibhas Report Sample has 5 items selected from 8 records
10	[Available for QA] Sample Ordinary Normal Sample has 5 items selected from 8 records
11	[Available for QA.] Sample Dominic'c Test Batch has 5 items selected from 8 records
12	[QA Complete] Sample mgetest has 200 items selected from 9999 records
13	[QA Complete] Sample dthpodtest has 5 items selected from 36 records
14	[QA Complete] Sample wayward bowling has 20 items selected from 37 records
15	[Transmitted To Production] Sample poddthreduced has 2 items selected from 37 records
16	[Transmitted To Production] Sample reduced-pod has 2 items selected from 43 records
17	[Available for QA] Sample mgepodtest has 13 items selected from 100 records
18	[Available for QA] Sample christmas has 13 items selected from 100 records

Display | 35 matching records | vw | RegSup | 50010

- **vwsecurity run automatically when required:** A new Audit Trail handler has been added that monitors the Vitalware Registry for changes where a user's settings have been modified. Where settings have been changed, `vwsecurity` is executed to update the security settings for all Vitalware databases. The following events are handled:
  - Addition of a new user.
  - Deletion of an existing user.
  - Changing the group(s) of an existing user.
  - Altering database security via the Security Registry entry.

  There should be no need to run `vwsecurity` manually.
- **Image Quality available for JPEG 2000:** When generating JPEG 2000 based images, the Image Quality may now be defined. The Image Quality determines the level of sampling used when creating the image. The value is between 1 and 100, where 100 implies full sampling, that is no lose of image quality, and 1 represents the least amount of sampling, resulting in a much smaller file size, but with image degradation.
- **Assign Till number for invoice and zero total amount records:** When records are inserted the Till is assigned to the POS, Ledger and Orders records. Two exception to this were when an invoice was created and when no payment was required for an order. In these cases the Till

was not assigned until a payment was made for the record. This presented a problem for batch processing because each batch was processed by querying for all POS records on the current Till. The Till number is now assigned to all POS records at their time of creation.

- **The ability to continue with batch printing when an error occurs:** A new Registry entry has been added to allow batch printing to continue after an error has occurred. The entry is:

  ```
  Group|Default|Table|epos|Continue Batch Print After Error|true
  ```

  When this is set, each error is shown and then the printing continues from the next record.
- **The invoice module updated to handle multiple payments entered at any one time:** At times it is possible for a customer who has received an invoice to make the payment using two payments and on occasions both payments may come in at the same time. The invoice payment processing only expected one payment and so only processed one. A change was made so that each added payment would be processed in turn thus allowing either one or multiple payments onto an invoice at the same time.
- **POS module updated to recalculate the invoice amount as new products are added** rather than waiting until the record is saved: When additional products were added to a invoice transaction that had not yet been invoiced, the total amount was updated but the invoice payment amount was not. This was confusing for the operator who would try to adjust the total. The calculation for invoice amount has been adjusted so that it is updated when products are added.
- **Resending of failed NRS messages:** If for some reason an NRS message that should have been sent was not sent (e.g. as a result of a logic error), there was no trigger to enable the message to be sent. The `vwauditdump` script was modified to allow audit records to be scanned to find the appropriate audits so these could be passed back through the NRS mechanism. An additional `scriptvwnrsreprocess` was created which takes the found audit records and generates the expected XML files.
- **Event type correction when the barcode scanning process inserts a certificate audit record:** An invalid event type error could be shown when an event was not linked to the order that generated it. A change was made to check the Products module based on the product ordered to determine the correct event type.
- **New Process Menu:** A new Process menu has been added to the Registration and POS modules. The new Communications and Sampling functionality (see above) can be found under this menu. It was decided to move some other functionality under the Process menu: the Adoption (Births) and Image menus (Registration tables and POS) have been moved under the Process menu.
- **Improve efficiency of date range queries used in generation of statistics.** The following statistics generation scripts were also updated to utilise the new stats method:

  ```
  etc/periodic/monthly/AuditOperationTableUser.pl
  etc/periodic/monthly/CertificatesPrintedByType.pl
  etc/periodic/monthly/CertificatesPrintedByUser.pl
  etc/periodic/monthly/CertificatesReprinted.pl|
  etc/periodic/monthly/CertificatesVoidedCancelled.pl
  etc/periodic/monthly/LoginsUser.pl
  etc/periodic/monthly/OrdersByProduct.pl
  etc/periodic/monthly/RegistrationByType.pl
  etc/periodic/weekly/AuditOperationTableUser.pl
  etc/periodic/weekly/CertificatesPrintedByType.pl
  etc/periodic/weekly/CertificatesPrintedByUser.pl
  etc/periodic/weekly/CertificatesReprinted.pl
  etc/periodic/weekly/CertificatesVoidedCancelled.pl
  etc/periodic/weekly/OrdersByProduct.pl
  ```

```
etc/periodic/weekly/RegistrationByType.pl
local/qld/etc/periodic/monthly/DeathsByReceiptMethod.pl
local/qld/etc/periodic/monthly/RegistrationByType.pl
local/qld/etc/periodic/weekly/CertificateTurnaround.pl
local/qld/etc/periodic/weekly/CONApplications.pl
local/qld/etc/periodic/weekly/CONCompleted.pl
local/qld/etc/periodic/weekly/DeathsByReceiptMethod.pl
local/qld/etc/periodic/weekly/RegistrationByType.pl
```

Issues Resolved

Issue	Resolution
If a multi-term query is performed on an attachment field where the field searched in the attached module has `Also Search` Registry entries associated with it, the query generated may not be the most efficient one possible.	The query generated for multi-term attachment queries where the attached column has `Also Search` Registry entries is now efficient.
For systems running in UTF-8 mode, the Audit Trail mechanism will drop audit records containing invalid UTF-8 characters. As the audit records are not posted to the Audit Trail table a "gap" may appear in the audit trail. The issue only occurs where invalid data is stored in a record (normally as a result of bad data when importing into Vitalware).	The Audit Trail mechanism now replaces invalid UTF-8 characters with the invalid UTF-8 character (0xFFFD). The substitution allows the audit record to be processed without error.
The **View>Attachments** command, used to show all records attached to the current record, displays modules the user does not have permission to use. While it is useful to see whether any records in the module attach to the current record, the module cannot be invoked without displaying a permission denied error.	The **View>Attachments** command has been modified to show only those modules the user has permission to access. A check box has been added to *Show all modules*, allowing the old behaviour to be selected.
The `vwlutsupdate` server-side script does not update Lookup List files in`local/luts/default` if it exists. Files in the `luts/default` directory are always updated. The issue means that localised versions of Lookup Lists may not be updated correctly by Vitalware upgrade scripts.	Local versions of Lookup List file are now updated by the `vwlutsupdate` server-side command.
Data in reference fields, that is fields displaying data from another module, may not contain the correct values after certain operations are performed. The affected operations are: • Discard Record • Replace (Global Edit) • Export Records • Merge Record	Data in reference fields now displays the correct values after the listed operations are performed.

Issue	Resolution
• Sort • Delete Record	
The value in a read-only combo-box may be cleared using the `backspace`key the first time data is viewed. The `backspace` key will not clear the value when viewing subsequent records.	The `backspace` key is disabled when a combo-box is read-only, regardless of when the data is viewed.
Random Access Violation messages may display indicating an error has occurred in mshtml.dll. The error message may even appear when a user is not at their machine.	The error message is no longer displayed.
When performing a spell check on a RichEdit control that has multiple lines of text, the highlighting of the words spelled incorrectly may be out by a few characters.	Words spelled incorrectly are now highlighted correctly.
The View Attachments button next to a read-only grid may not be enabled when a single row is displayed in the grid. The button is only disabled for grids displaying data from another module (a reverse reference column).	The View Attachment button is now enabled when a single value is in a read-only grid.
The default value for the *Gender* field in the Parties module is set to`Female`. Ideally the default values should be `Unknown`.	The default value for the *Gender* field is now `Unknown`.
When changing the colours for various attributes (e.g. mandatory fields, etc.) via the Options dialogue box, the new colours selected will not take effect in modules already open. New instances will display the correct colours.	Existing modules now update their colours correctly when the Options dialogue box is closed.
When switching to the Details view of a record on a tab that contains an HTML control, the input focus is set to the HTML control, rather than the first control on tab.	The input focus is set to the first control on a tab when switching from Details view, even if the tab contains an HTML control.
A *List of Values failure: fail to get values* message may be displayed when	The error message is no longer displayed for Crystal Reports

Issue	Resolution
generating a Crystal Report that contains dynamic parameters.	containing dynamic parameters.
The Audit Trail facility does not flush logging information when the processing of a record has been completed. The lack of flushing means the log file contents may lag behind the processing of records.	The Audit Trail facility now flushes logging information after each records processed.
The auto filling of values for the upper levels of hierarchies may be slow where the controls that form the hierarchy appear on a number of tabs.	The speed of auto filling of values in hierarchies has been improved dramatically.
Not all of the certificates from different versions of a registration were showing in the historic tack grid.	All certificate records from the different record versions now show in the historic grid.
Trying to reprint a certificate while logged on a batched Till could result in the reprint not being sent to the printer.	The reprint is now correctly sent to the printer.
Running the special + and !+ reference queries in POS did not return the correct results.	Using the + or !+ reference queries returns the correct results.
Nightly jobs did not complete successfully because record updates are failing due to the fifo server not running.	The nightly jobs now ensure the fifo load is running so that all jobs are successfully completed.
When running logging through `vwlogger, the logs` disappear when the directory specified to log to does not exist.	The logging now creates the directory so that no logs are lost.
When trying to add an invoice payment on a day that was not the day a transaction was inserted, the payment type was disallowed.	Invoice payments can now be made to under transactions on days other than when the POS record was inserted.
When using undo maintenance the status of the record could sometimes be reset incorrectly.	The status of all records is now correctly set when using undo maintenance.
When a duplicate Registration number error was thrown, a duplicate Registration could be created when the record was saved.	A duplicate record is no longer created after a duplication Registration number error is shown.

Issue	Resolution
The nightly matching process could generate an invalid query when running over historic records.	The nightly matching process no longer generates invalid queries.
Background loads fail if the fifo load is not running.	The fifo load is always started before any other load starts so that all other loads now correctly start.
When selecting **Cancel** from the Reprint Quantity box, certificates still print.	Certificates no longer print if **Cancel** is selected from the Reprint Quantity box.
The entire invoice payment could be assigned to the final POS transaction if the initial save of invoice payment failed.	The correct amount is now assigned to each POS transaction.
The end of month report appears to be missing data from newly added Till locations.	The cashbook now correctly shows the data from the new Till locations.
When attempting to delete a record the user may not be allowed to if a backend database does not exist (e.g. in a web environment).	The record now may be deleted provided no reference links exist.
Previous field on the Maintenance tab did not correctly show previous record version.	The previous field now correctly shows the previous record version.
Users can access Registration tables via POS searching when they do not have access to the Registration tables.	Users can no longer access the Registration tables unless they have permission to do so.
The `MoneyChanged` flag is not cleared on a cancelled edit, resulting in the possibility of receipts printing when they should not.	The flag is correctly cleared and so receipts only print when they should.
Work Hours is not correctly calculated when insertion and completion time is outside of business hour.	Work hours are now correctly calculated for times outside of business hours.

Upgrade Notes

The upgrade from Vitalware Version 2.1.02 to Vitalware 2.2.01 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. Upgrading comprises the following steps:

- Stopping Vitalware services
- Installing Texpress
- Upgrading TexAPI
- Upgrading Vitalware Server
- Starting Vitalware services
- Upgrading Vitalware Client

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 each **data** file is empty and no **data.t** files exist.
5. If this is not the case, please wait for the loads to drain before proceeding.
6. Enter **`vwload stop`**
7. Enter **`vwweb stop`**
8. 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-2-2-01**

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 2.2.01. Once Texpress 8.3 has been installed, this section may be skipped for subsequent upgrades.

1. Enter **cd ~*vw***
2. Enter **mkdir -p texpress/8.3.001/install**
3. Enter **cd texpress/8.3.001/install**
4. Obtain the appropriate Texpress version for your Unix machine via the *Texpress* hyperlink at the top of the page.

   Save the release in **~*vw*/texpress/8.3.001/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/8.3.001**

   The Texpress installation script will commence.
8. Enter **cd ~*vw*/texpress/8.3.001**
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 8.3.001 8.3**

Upgrading KE TexAPI

Installing TexAPI is only required for the first client upgraded to Vitalware 2.2.01. Once TexAPI has been installed, this section may be skipped for subsequent upgrades.

1. Enter **cd ~*vw*/texpress**
2. Enter **mkdir 6.0.003**
3. Obtain the appropriate TexAPI version for your Unix machine via the *TexAPI* hyperlink at the top of the page.

   Save the release in **~*vw*/texpress**, calling it **texapi.sh**.
4. Enter **sh texapi.sh -i ~*vw*/texpress/6.0.003** (expand the **~*vw***).
5. Enter **\rm -f texapi**
6. Enter **ln -s 6.0.003 texapi**
7. Enter **\rm -f texapi.sh**

Upgrading Vitalware Server

1. Enter **cd ~*vw/clientname***
2. For Unicode based clients only.

   Enter **vi .profile-local**

   Change **langcode=utf8** to **langcode=utf-8** and save the change.
3. Enter **mkdir install**
4. Enter **cd install**
5. Obtain the appropriate Vitalware server version bundle via the Vitalware *Server* hyperlink at the top of the page.

   Save the release bundle file in **~*vw/clientname*/install** calling it **vw.sh**.
6. Enter **sh vw.sh**

   The Vitalware release will be extracted.
7. Enter **. ./.profile**
8. Enter **bin/vwinstall -u *clientname***

   The Vitalware installation script will commence.
9. Enter `cd` **~*vw/clientname***
10. Enter **cp .profile.parent ../.profile**
11. Enter **. ../.profile**
12. Enter **client *clientname***
13. Enter **vwreindex**
14. Enter **vwbldinstall**
15. Enter **vwbldlinks**

    Removal of the temporary directory (and its contents) is recommended:
16. Enter **rm -fr install**
17. Enter **upgrade-2-2-01**

The client will now be upgraded to Vitalware 2.2.01. If you are upgrading from a version prior to Vitalware 2.1.02, you must run the upgrade scripts for all versions after the old version **before** running the Vitalware 2.2.01 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-2-2-01**.

Upgrading Vitalware Client

When upgrading to Vitalware 2.2.01 the Windows client should be upgraded on each individual machine. The client upgrade installs new DLLs on each individual machine in order for all reports with dynamic parameters to function correctly. To upgrade the Vitalware Client follow the Installing Vitalware Client notes.

Perl Packages

If the administration utility to change a user's password is used, the perl `Net::SSH::Expect` package must be installed. If the password utility is not used, this step may be skipped. To install the package:

1. Log in as **root**
2. Enter **`perl -MCPAN -e shell`**
3. Enter **`install Net::SSH::Expect`**
4. Enter **`quit`**

Vitalware®
Vital Records Management

The World's Premier
Vital Records
Software

Vitalware Documentation

Multi-group Support

Document Version 1

Vitalware Version 2.1.03

KE Software

www.kesoftware.com

Contents

SECTION 1

Overview

Users are given access to Vitalware by assigning them to a group. Each group has a set of permissions associated with it, and the user inherits the permissions of the group to which they have been assigned. Individual user based permissions may then be defined to override group based settings as required. The use of group based permissions means that it's not necessary to specify all the permissions on a per user basis, and only the difference between the group permissions and any user specific permissions need to be defined for any one user.

Groups are generally based around real life roles, with each group reflecting the permissions required by all users who undertake the role. For example:

- A `Registrations` group might allow users to create new Registration records, but not to create new orders.
- A `Counter` group would have permission to create POS records and to print certificates, but not to create new orders.

Assigning a user to a group, casts them in that role.

In general, the group system works well in Vitalware, except when a user performs more than one role within the institution. For example, a mail room clerk (group `Mail Room`) may relieve at the counter (group `Counter`) during lunch times or periods of heavy demand. Until now, Vitalware provided two solutions to this dilemma:

1. Create two usernames, assigning one to group `Mail Room` and the other to group `Counter`. The user must then use the correct username when logging into Vitalware to perform the required role.

   Moving from one role to the other (e.g. while the user is logged in as a mail room clerk), requires the user to log out of Vitalware and back in using the `Counter` username. This could become tedious, and it requires that the user remembers two usernames and two passwords.
2. The second solution requires a new group to be created which is a merge of the permissions of the `Mail Room` and `Counter` groups.

   The problem with this solution is that the combined privileges present a view of the world that is neither mail room nor counter specific.

In most cases the two original roles are sufficient and all that is required is a mechanism to switch between the two groups without having to log out and back in as another user.

Vitalware 2.1.03 introduces multi-group support, which allows a single user to be registered in more than one group:

- A user who is a member of more than one group can select the group name to use from the Login dialogue box when logging into Vitalware.
- At any time it is possible to switch to another group without logging out and back in again.
- When switching groups the user decides whether opened modules should remain open or whether they should be closed.
- Any new modules opened will use the group permissions assigned to the group that the user switched to.

By not closing open modules it is possible to have modules in different groups open at the same time. Vitalware ensures the correct group permissions are observed based on the group associated with the module.

With multi-group support, a user is able to log in using one group, switch to another group while leaving existing modules open, perform operations in the new group and then switch back to the previous group, all without having to close any modules. A key feature of multi-group support is that it allows users to change roles without losing their current work position and then to return to that position at a later time.

> Texpress 8.2.009 or later must be installed to provide the back-end security facilities for multi-group support.

SECTION 2

Specifying a user's groups

The Vitalware Group Registry entry defines which group a user is assigned to. It is consulted whenever a user logs in to Vitalware. The format of the Registry entry is:

`User|`*username*`|Group|`*groupname*

where *groupname* specifies the group to use when determining permissions.

For example:

```
User|badenov|Group|Mail Room
```

specifies that user `badenov` is a member of the `Mail Room` group and has all the permissions assigned to that group.

In order to provide multi-group support, the Group Registry entry has been extended to allow a semi-colon separated list of groups to be specified. The order of the groups is not important.

For example:

```
User|badenov|Group|Mail Room;Counter
```

specifies that user `badenov` is a member of both the `Mail Room` and `Counter` groups.

Changes to the list of groups that a user has access to only come into effect when the user logs into Vitalware following any change to their Group Registry entry. If they are logged in when the change is made, they will need to log out and back in again in order to have access to the updated list of groups.

How to set an existing user's groups

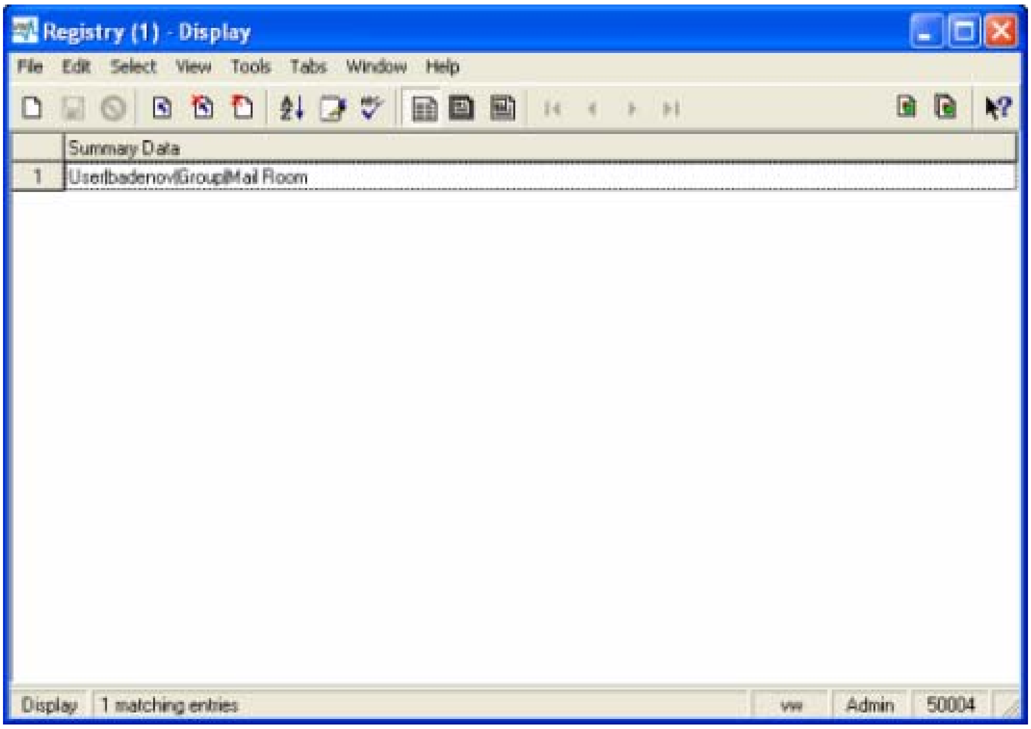
1. Log in to Vitalware as a System Administrator.
2. In the Registry module, search for the user's Group Registry entry:

   i. Enter `User` into the *Key 1* field.

   ii. Enter the user's username into *Key 2*.

   iii. Enter `Group` into *Key 3*.

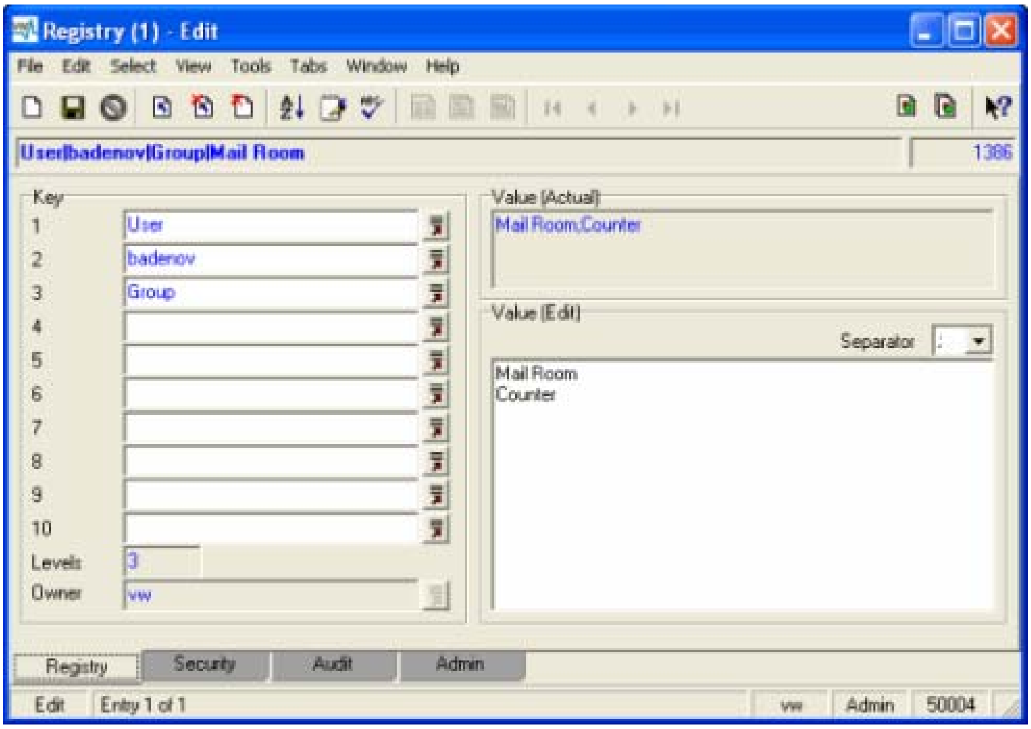
   The matching Registry entry displays:



	Summary Data
1	User|badenov|Group|Mail Room

Display | 1 matching entries | vw | Admin | 50004

3. In the *Value (Edit)* field on the Registry tab, enter the name of all groups that the user is a member of. Each group should be placed on a separate line:


Registry (1) - Edit
File Edit Select View Tools Tabs Window Help
Registry Security Audit Admin

User|badenov|Group|Mail Room 1386

Key
1 User
2 badenov
3 Group
4
5
6
7
8
9
10
Levels 3
Owner vw

Value (Actual)
Mail Room;Counter

Value (Edit)
Separator ;
Mail Room
Counter

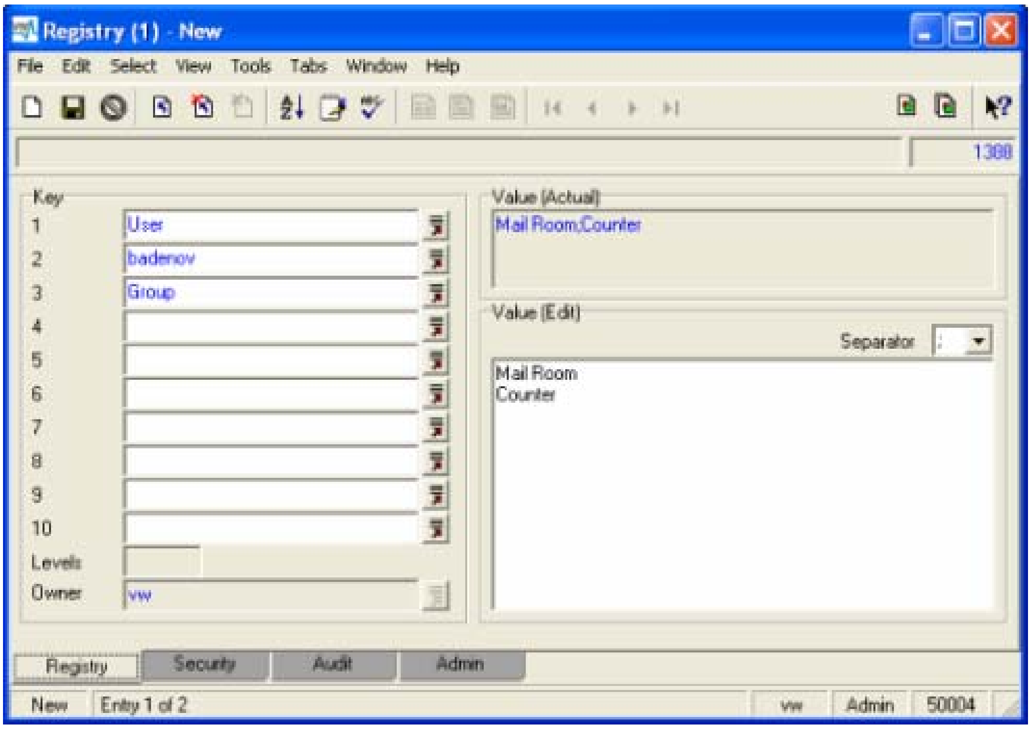
Edit Entry 1 of 1 vw Admin 50004

4. Save the record.

> As of Vitalware 2.1.03, security profiles are generated automatically after saving any user registration based Registry entries and it is no longer necessary to run **Tools>Generate Record Security** after changing user settings.

How to set a new user's groups

1. Log in as a System Administrator.
2. Add a new record in the Registry module:
   i. Enter `User` into the *Key 1* field.
   ii. Enter the user's username into *Key 2*.
   iii. Enter `Group` into *Key 3*.
3. In the *Value (Edit)* field on the Registry tab, enter the name of all groups that the user is a member of. Each group should be placed on a separate line:


Registry (1) - New
File Edit Select View Tools Tabs Window Help
1388
Registry Security Audit Admin

Key
1 User
2 badenov
3 Group
4
5
6
7
8
9
10
Levels
Owner vw

Value (Actual)
Mail Room;Counter

Value (Edit)
Separator ;
Mail Room
Counter

New | Entry 1 of 2 | vw | Admin | 50004

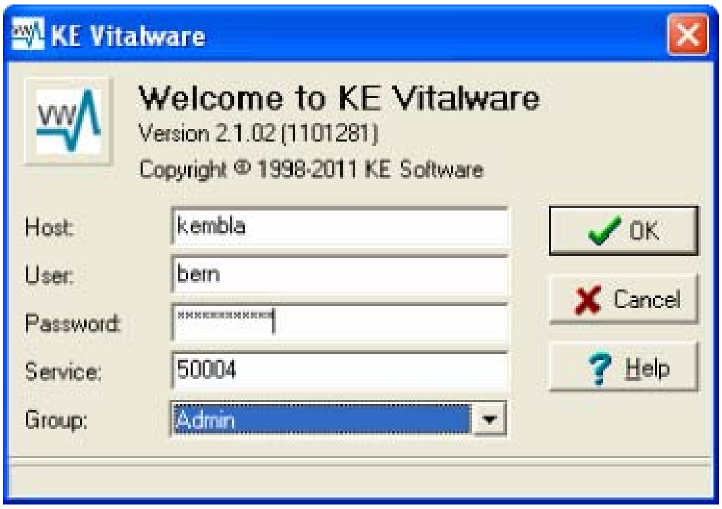
4. Save the record.

As of Vitalware 2.1.03, security profiles are generated automatically after saving any user registration based Registry entries and it is no longer necessary to run **Tools>Generate Record Security** after changing user settings.

SECTION 3

How to work with multiple groups in Vitalware

When a user has been registered to use multiple groups, the Vitalware Login dialogue box includes a drop list with the names of all available groups for the user:



Welcome to KE Vitalware

Version 2.1.02 (1101281)

Copyright © 1998-2011 KE Software

Host: kembla

User: bern

Password: ************

Service: 50004

Group: Admin

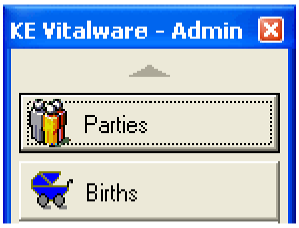
The list of groups that displays is determined by the combination of *username* + *service*. A user can have the same *username* on multiple services, and in each service may be a member of different groups.

The list is displayed only if the *username* supports more than one group.

A user must log in to Vitalware after changes have been made to their group membership before the (updated) list of groups will appear at login. Once a user has successfully logged in to Vitalware, the system Options may be used to switch to another group. See Switching groups via a module (page 10) for details.

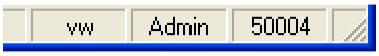
Active group and module group

Vitalware keeps track of the group selected when a user logs in to the system. This group is known as the active group and it is displayed in the title bar of the Command Centre. In this example, the active group is group Admin:



Modules started by clicking a button in the Command Centre belong to the active group. The active group is also highlighted in the Login dialogue box the next time the user logs in to Vitalware.

Modules are also associated with a group. When a module is invoked from the Command Centre its group is that of the Command Centre, namely the active group. If a module is created from within another module, by selecting **Window>New>***module*, the module is associated with the same group as the module from which it was created. A module's group is displayed in the Status bar at the bottom right of the window. In this example the module's group is group Admin:



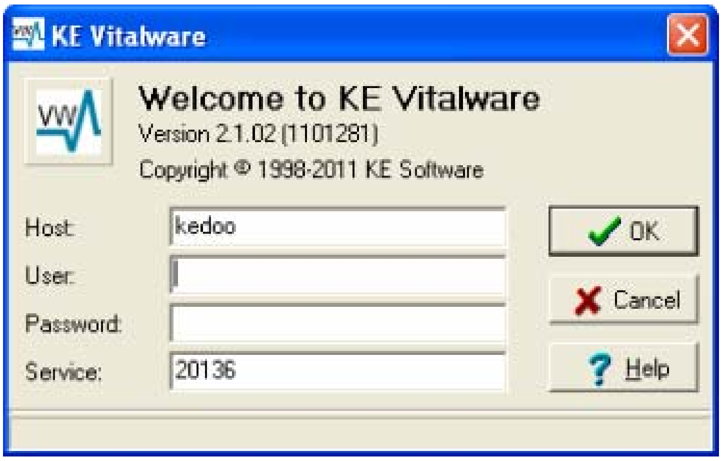
A module cannot change group. Once it is created and associated with a group, it will remain in that group.

It is possible to change the active group however. By doing so it is possible to create modules in different groups, thus allowing users to have multiple roles within the one Vitalware session.

Logging in to a group

1. Start the Vitalware application.

   The Login dialogue box displays:



KE Vitalware

Welcome to KE Vitalware

Version 2.1.02 (1101281)

Copyright © 1998-2011 KE Software

Host: kedoo

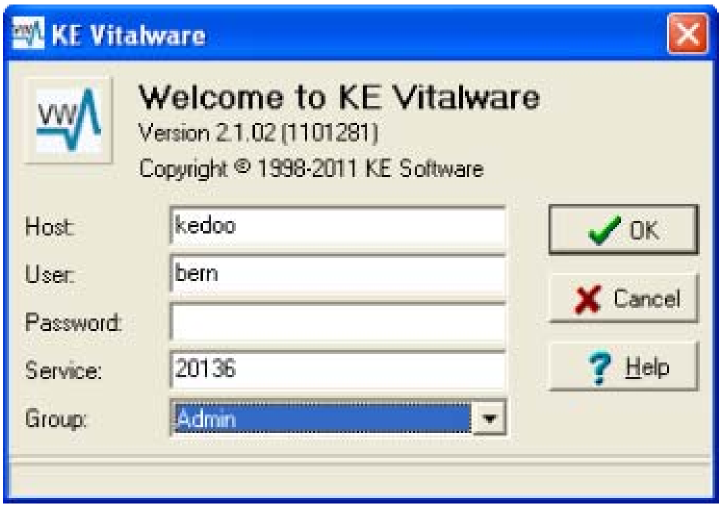
User:

Password:

Service: 20136

2. Enter your username into the *User* field.

   If you are registered for more than one group with the current Service and you have logged in successfully previously, a Group drop list will be added to the Login dialogue box*:*



KE Vitalware

Welcome to KE Vitalware

Version 2.1.02 (1101281)

Copyright © 1998-2011 KE Software

Host: kedoo

User: bern

Password:

Service: 20136

Group: Admin

3. Select a group from the Group drop list.

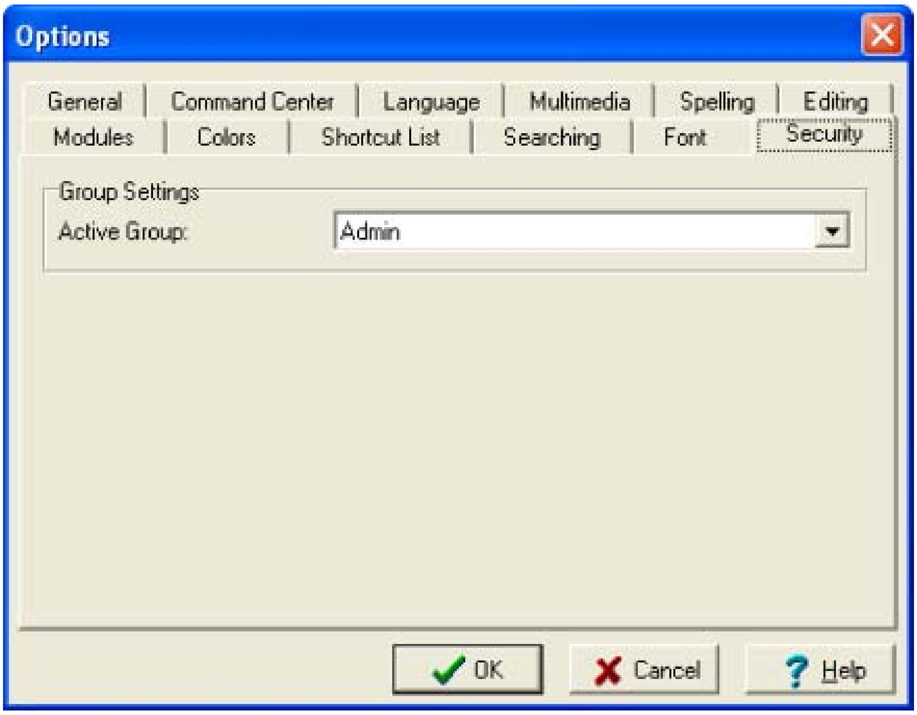
4. Complete the rest of the log in details and select ✓ OK.

Switching groups via a module

1. Select **Tools>Options** from the Menu bar of any open module.

   The Options dialogue box displays.

2. Select the **Security** tab:



Group Settings

Active Group: Admin

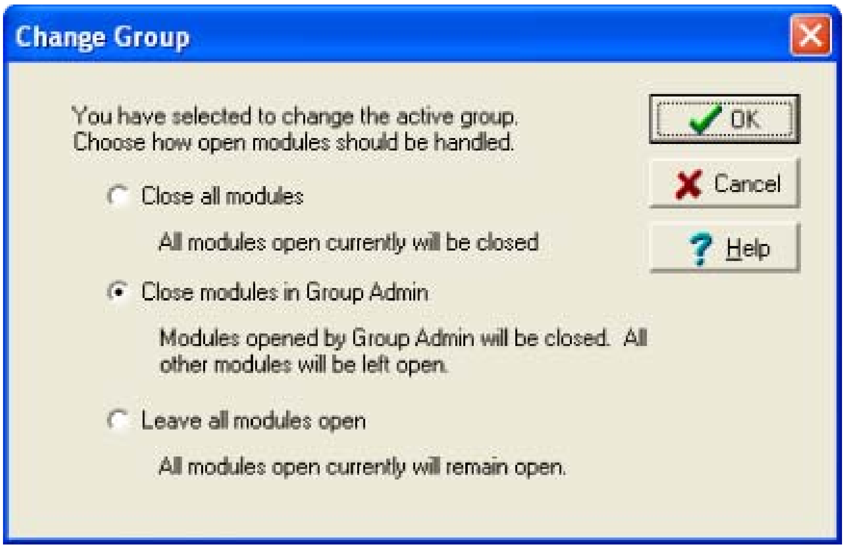
   The current active group is displayed in the *Active Group* drop list.

3. Select a group from the *Active Group* drop list.

   The list only contains those groups the user is registered to use.

4. Select ✓ OK.

   The Change Group dialogue box displays.



You have selected to change the active group.
Choose how open modules should be handled.

○ Close all modules

All modules open currently will be closed

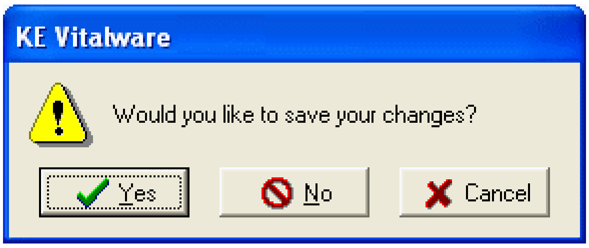
● Close modules in Group Admin

Modules opened by Group Admin will be closed. All other modules will be left open.

○ Leave all modules open

All modules open currently will remain open.

5. Select how any open modules should be handled when the active group is changed:
   - **Close all modules**
     All open modules will be closed. Only the Command Centre will remain open.
   - **Close all modules in Group** *group*
     All modules open in the active group (*group*) will be closed.
     This option is useful for closing all modules created since the last time the group was switched. For example, if you were asked a question that required switching groups to respond, this option will close the module(s) created to answer the question. All other open modules are left untouched.
   - **Leave all modules open**
     All modules currently open will remain open. Each module will remain in its current state and group. Once the switch is complete the modules may be used as required. Each open module will continue to provide functionality consistent with the module's group.
6. Select ✓ OK.
   The active group will be switched. If a module scheduled for closure contains unsaved data, a message will display prompting the user to save the data before the module is terminated:

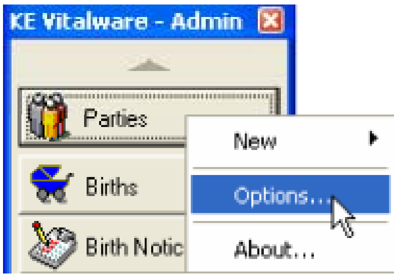


KE Vitalware

Would you like to save your changes?

Selecting ✗ Cancel will abort the switching process.

Switching groups via the Command Centre

1. Right click the Command Centre.
   A context menu displays:
   

2. Select **Options**.
   The Options dialogue displays
3. Continue from Step 2 above (page 10).

SECTION 4

Registry cache

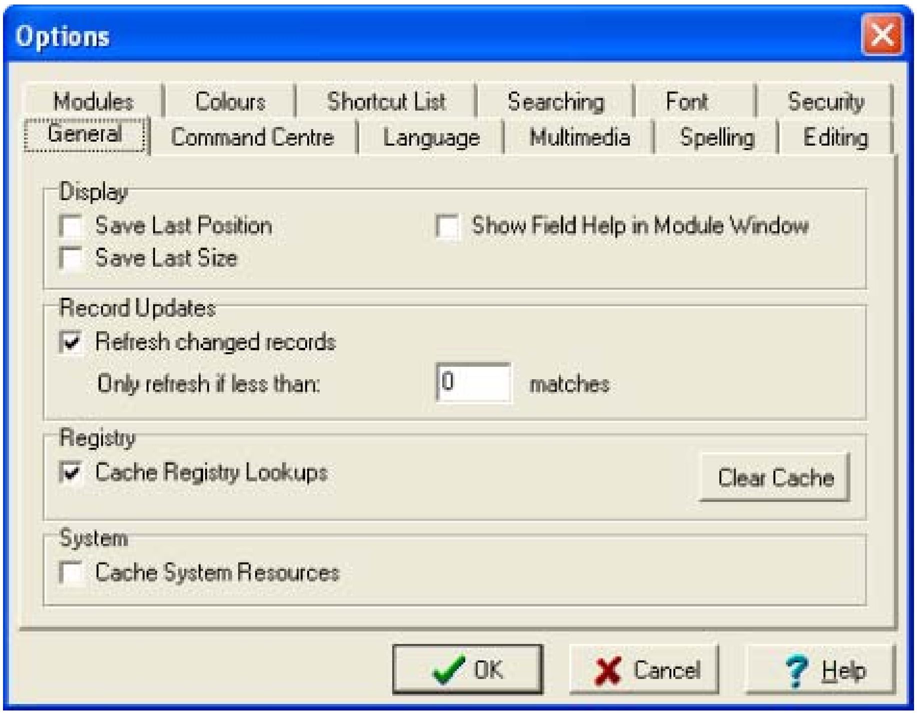
The permissions of all groups and users in the system are specified in the Vitalware Registry and this information is stored on the Vitalware server. The Registry must be consulted each time the Vitalware client is required to determine whether a user is permitted to perform a given operation. As Vitalware provides a sophisticated security module the Registry may be accessed heavily for certain operations (starting a new module for example). In order to reduce the time required to query the Registry for permissions, the Vitalware client now contains a Registry cache. The cache is a mechanism that remembers what permissions have been asked for and the associated response. If the same question is asked again, the answer can be given without the need to access the Vitalware server. The Registry cache reduces traffic to the Vitalware server and provides a speed improvement in the Vitalware client.

The one disadvantage of using a cache is that changes made to the Vitalware Registry by other users will not be picked up until either the cache is cleared (flushed) or you log out and log back in again. For example, if you are using Vitalware and you have been producing reports from the Parties module, and another user adds a new report to the Parties module, you will not see the report listed until you log out and log back in again, or until you flush the cache.

In practice this tends not to be a serious issue and the speed gains more than outweigh any inconvenience. However, it may occasionally be necessary to clear the Registry cache:

Flushing the Registry cache

1. Select **Tools>Options** from the Menu bar of any open module.
   The Options dialogue box displays.
2. Select the **General** tab.



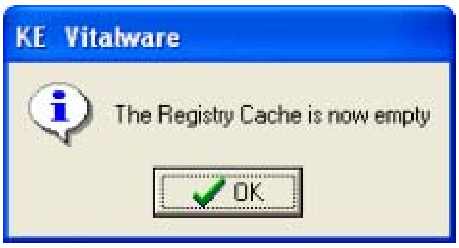
Display
□ Save Last Position
□ Save Last Size
□ Show Field Help in Module Window

Record Updates
☑ Refresh changed records
Only refresh if less than: 0 matches

Registry
☑ Cache Registry Lookups

System
□ Cache System Resources

3. Select Clear Cache.
4. Select ✓ OK.
   A message displays once the cache is empty.

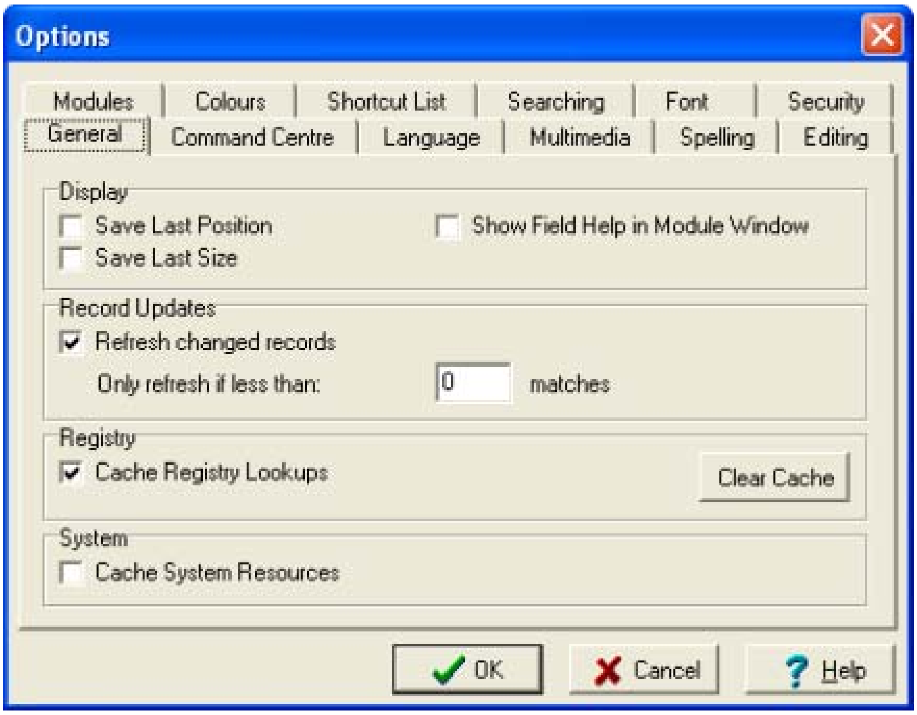


The Registry Cache is now empty

Flushing the cache removes all Registry responses stored in the Vitalware client. Next time a Registry question is raised, the Vitalware client will contact the server for a response and the response will be added to the cache again. Until the cache becomes populated again a decrease in client performance may be experienced.

Enable / disable the Registry cache

1. Select **Tools>Options** from the Menu bar in any open module.
   The Options dialogue box displays.
2. Select the **General** tab:



Display
□ Save Last Position
□ Save Last Size
□ Show Field Help in Module Window

Record Updates
☑ Refresh changed records
Only refresh if less than: 0 matches

Registry
☑ Cache Registry Lookups

System
□ Cache System Resources

3. Tick / un-tick the **Cache Registry Lookup**s checkbox to enable / disable the Registry cache.
4. Select ✓ OK.
   If the Registry cache is disabled, it is flushed silently as its entries can no longer be used.

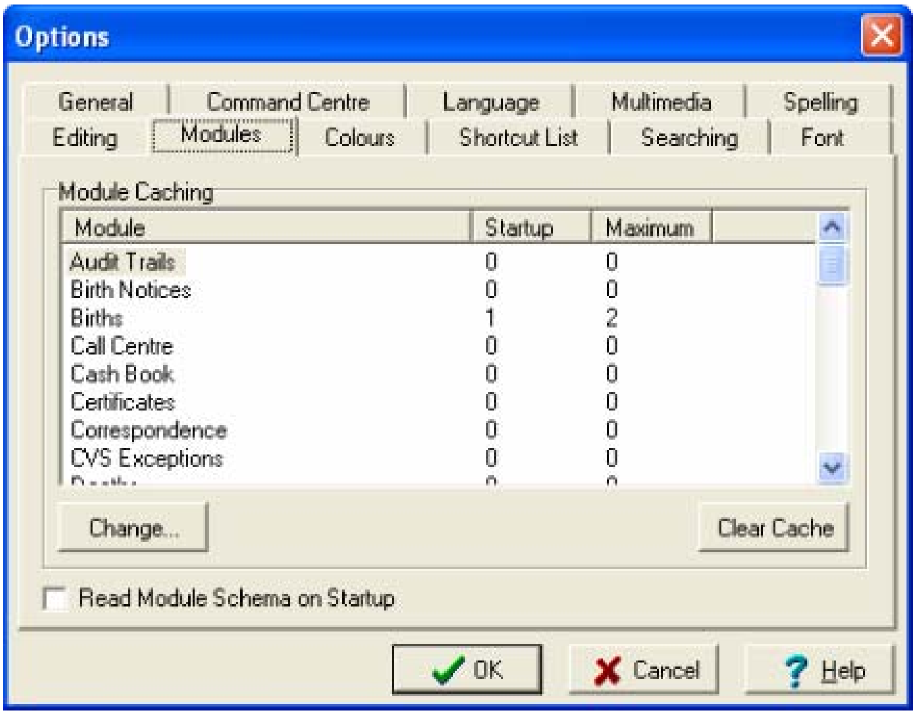
Disabling the Registry cache will cause the Vitalware client to run significantly slower as all Registry queries must be answered by the server. The only useful purpose achieved by disabling the cache is to test the effect changing Registry entries has on a module, without the need to flush the Registry cache. In day to day use, the cache should be enabled.

SECTION 5

Module caching

Allowing users to switch between groups has implications for how Vitalware performs module caching.

Module caching is a mechanism that allows modules to be hidden, rather than destroyed, when they are closed. If a new instance of the module is required, the hidden module may be used. Module caching provides significant speed improvements in situations in which modules are opened and closed on a regular basis. Module caching is configured via the Modules tab in the Options dialogue box:



Module	Startup	Maximum
Audit Trails	0	0
Birth Notices	0	0
Births	1	2
Call Centre	0	0
Cash Book	0	0
Certificates	0	0
Correspondence	0	0
CVS Exceptions	0	0

□ Read Module Schema on Startup

In the example above, one instance of the Births module is created when Vitalware is invoked. When a Births module is closed, a maximum of two instances are maintained in the module cache.

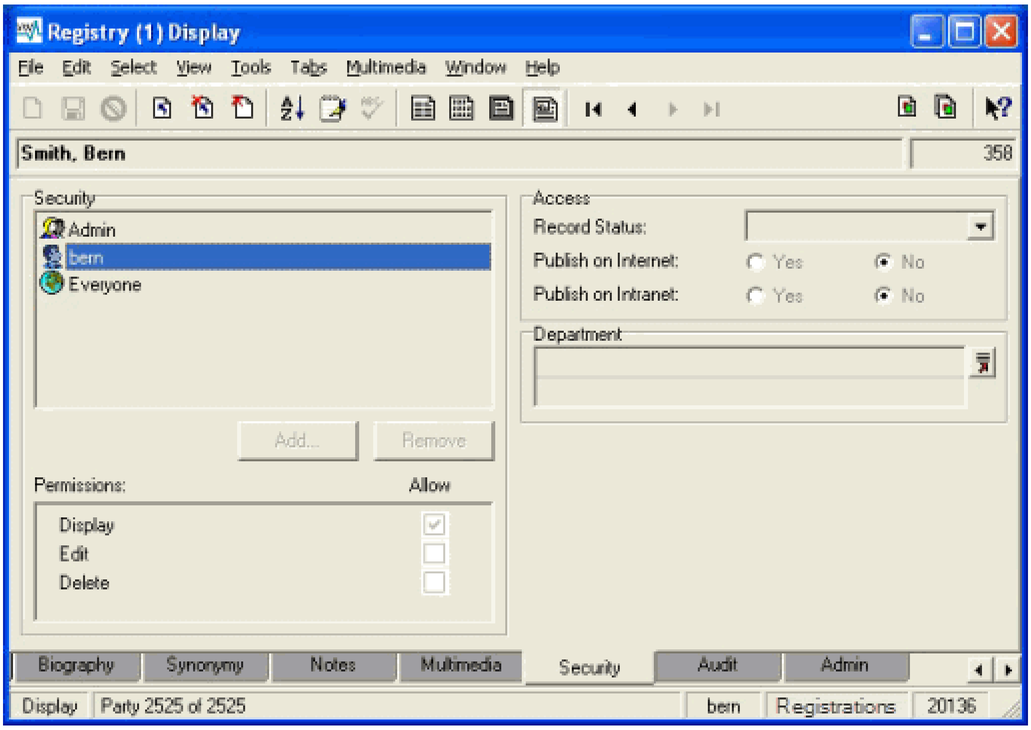
With the introduction of support for multiple groups, the number of modules to cache now applies on a per group basis. Thus for the previous example, up to two instances of the Births module will be cached per group.

When switching groups, the Change Groups dialogue box allows the user to indicate not only how open modules should be handled, but also how cached modules are dealt with. For example, if you choose to *Close all modules*, then all open modules are closed and all cached modules are flushed.

SECTION 6

Record Level Security

Enabling users to be a member of multiple groups has implications for how record level security permissions are displayed in the Vitalware client.



Smith, Bern 358

Security

- Admin
- bern
- Everyone

Permissions:	Allow
Display	☑
Edit	□
Delete	□

Access

Record Status:

Publish on Internet: ○ Yes ● No

Publish on Intranet: ○ Yes ● No

Department

Display | Party 2525 of 2525 | bern | Registrations | 20136

Name selected in the *Security* field:	**Permissions displayed**
The person currently logged in to Vitalware	The permissions displayed are the group permissions that apply to the module. A module joins a group in one of two ways: • If the module was created by clicking a button in the Command Centre, the module's group is the one that the user was logged in as when the module was opened. • If the module was created by selecting **Window**>**New**>*module* from a module Menu bar, the module's group is the same as the module from which it was created. The module's group is shown in the module's Status bar at the bottom right of the window (`Registrations` in the example above). For example, if a user is in two groups, `Mail Room` and `Counter`, and the module was opened from the Command Centre when the user was logged in as a member of the `Mail Room` group, then the permissions displayed are those of the `Mail Room` group. This is as expected as the permissions reflect what operations (edit, delete) can be performed on the displayed record. In the example above, user *bern* is logged in to Vitalware and the Parties module is in group `Registrations` (indicated in the Status bar). The record level permissions indicate that *bern* does not have permission to edit or delete the record, even though *bern* is also in group *Admin*, which does have edit permission.
Anyone else	The permissions displayed are a merging of all the user's group permissions. For example, if a user is in two groups, `Mail Room` and `Counter`, where group `Mail Room` has permission to delete the record, but group `Counter` does not, then the permissions displayed will indicate that the user has delete permission. This is as expected as the user does have permission to delete the record, provided they log in or switch to group `Mail Room`.

SECTION 7

Security profile extensions

The Vitalware server security profiles have been extended to provide support for multiple groups per user. The profiles are maintained in XML format in a file named `security` in the database directory. Two new attributes have been added to the `<user>` tag to provide support for multiple groups:

- `level`

  The `level` attribute defines a label for the user profile. By changing the value of the `level` label for a given user, a different set of security settings is enabled. The group name is used as the label value for Vitalware databases. To switch between groups the Vitalware client changes the `level` value to match the group of the module with focus, that is, the module with which the user is interacting.
- `default`

  A `"yes"` value indicates this set of security settings should be used if the client has not set a `level` value. When the Vitalware client first connects, a `level` has not been set as the Vitalware Registry has not yet been consulted (a chicken and egg problem). Once the Registry can verify the log in group, the `level` is set to the supplied value.

A user security profile is created for each group that a user is registered to use (via the `User|`*username*`|Group` Registry entry). If user `badenov` has the following Registry entry:

`User|badenov|Group|Mail Room;Counter`

the following XML security segments are generated:

```
<user name="badenov" level="Mail Room" default="yes">
 ...
</user>
<user name="badenov" level="Counter">
 ...
</user>
```

The security profiles are built by the `vwsecurity` command. This server side command consults the Vitalware Registry and builds suitable security profiles for all modules, for all users, for each group a user is in. The command is invoked automatically whenever a `User|`*username*`|Group` Registry entry is created, modified or deleted. `vwsecurity` sets the `default` attribute to `"yes"` for the first group listed for each user.

The security level is set via the `seclevel` database option. The value of the option is the security level to use. If the option has not been set or the value is empty, the security profile with the `default="yes"` attribute specified is used. For example, to load data into the parties module using group `Counter` the following commands could be used:

```
epartiesopts=seclevel=Counter
export epartiesopts
texload ....
```

When using TexAPI, the `seclevel` is set via the `TexOptionSet()` call. For example, to change the security level to use group `Counter` for all Vitalware tables, the following call could be used:

```
TexOptionSet(session, NULL, "seclevel", "Counter");
```

For perl based scripts, the `OptionSet()` call is used to alter the security level. For example, to change the security level to use group `Counter` for all Vitalware tables, the following call could be used:

```
$session->OptionSet("", "seclevel", "Counter");
```

The `seclevel` option may be set on a per database basis or a system wide basis.

Index

Vitalware Documentation

Encrypted Connections

Document Version 1

Vitalware Version 2.2.01

www.kesoftware.com

Contents

SECTION 1

Encrypted Connections

When using Vitalware over public networks it may be desirable to encrypt all data transferred between the Vitalware client and server. Vitalware 2.2.01 has been extended to support an encrypted connection between the client and server programs. The encrypted connection uses TLS v1.0 (Transport Layer Security) for the transmission of data, ensuring data integrity and security. The use of data encryption is optional and is not required for internal networks where the risk of unauthorised access to data is minimal. The Vitalware server may also be configured to accept connections only from clients who request data encryption. This provides system administrators with the ability to enforce data encryption, or not, as required.

How it works

The TLS v1.0 protocol uses Public Key Infrastructure (PKI). A key is a sequence of bytes, normally 40, 56, 64, 128 or 265 bits in length, which is used by a cipher (an encryption algorithm) to encrypt data. Using the same cipher with different keys will produce different output. Hence, the key is used to "lock" the encrypted data. In order to unlock the data, that is decrypt it, the right key is required. With public/private key infrastructure two keys are generated, a public key and a private key. Data encrypted with the public key requires the private key in order to be decrypted and data encrypted with the private key requires the public key in order to be decrypted. In other words the keys are symmetrical.

The private key must be kept safe to ensure data privacy. If someone has both the private and public keys, they can decrypt the data and so compromise data security. The public key may be made available to anyone without compromising security as the private key is required to decrypt data encrypted using the public key. The public key is wrapped in a digital certificate, which consists of:

- **Public Key**
- **Subject** - details about the owner of the certificate.
- **Serial Number** - unique number used to identify the certificate.
- **Issuer** - details about who verified and issued the certificate.
- **Valid Dates** - start and end dates for which the certificate is valid.
- **Key Usage** - purpose(s) for which the public key may be used.
- **Thumbprint** - a check-sum to ensure the certificate has not been modified.

In order for a digital certificate to be valid it is necessary to verify the details of the Issuer. A small number of companies are allowed to issue valid and verifiable certificates. When you want a public digital certificate you approach one of these companies and they verify your details before issuing your public digital certificate. They *sign* your certificate with their own private key, making them the Issuer. In order to read your digital certificate you need the Issuer's public key. The key is embedded in their digital certificate which is available freely. These Issuer public digital certificates are known as Certificate Authorities (CA). In order to verify your certificate it is necessary to determine who the Issuer is and locate their CA certificate. Using the public key in their certificate your certificate can now be decrypted and the check-sum verified to ensure it has not been altered.

The private key is stored on the Vitalware server. Login access to the Vitalware server is generally restricted to user vw, hence the key is not available for general access. The public digital certificate is also stored on the Vitalware server. All CA certificates are stored with the Vitalware client.

When a connection is initiated the Vitalware server sends its public digital certificate to the Vitalware client. The client uses the CA certificates stored locally to verify that the certificate is valid. The server's public digital certificate contains the full host name of the Vitalware server machine. The Vitalware client checks the host name against the machine to ensure it has not connected to a rogue server.

Once the Vitalware client has verified the server's public digital certificate it sends a random number to the server encrypted using the public key in the server's digital certificate. As the server is the only machine with the private key it can decrypt the random number. The number is used as a key to a cipher (an encryption algorithm). The cipher uses the key to encrypt all data between the client and server. As the client and server are the only two machines which know the encryption key, data security and integrity is guaranteed.

The complete steps required to establish an encrypted connection are:

- The Vitalware client connects to the Vitalware server requesting a secure connection. The client provides a list of ciphers it supports.
- The Vitalware server selects the strongest cipher it supports from the client's list and notifies the client.
- The Vitalware server sends its public digital certificate to the client. The certificate contains the server's host name, the Issuer used to create the certificate and the server's public encryption key.
- The Vitalware client looks up the CAs on its machine and verifies that the server's certificate is valid.
- The Vitalware client generates a random number and encrypts it with the server's public encryption key. The random number is sent to the server.
- The Vitalware server decrypts the random number using its private encryption key (known only by the server).
- The random number is used as a key for the selected cipher. All data transferred is now encrypted using the agreed cipher.

Vitalware allows public digital certificates to be:

- **Self signed**

  A certificate that is verified by itself, that is the Issuer certificate is the same as the certificate itself. Self signed certificates allow institutions to generate their own digital certificates without the need to have them authenticated by an outside authority. In order for the certificate to be verified the self signed certificate must exist on both the client and server machines, the client version being the CA certificate.
- **Root signed**

  A certificate verified by one of a select set of "root" certificates. A root certificate is distributed as part of the public key infrastructure and can be installed on client machines to provide certificate verification.
- **Chain signed**

  A certificate is verified by its Issuer certificate. The issuer certificate itself is verified by its issuer certificate and so on until either a root or self signed certificate is found.

Vitalware allows both the client and server machines to define a list of ciphers they will support. When a connection is created the strongest (that is hardest to break) cipher supported by both the client and server is selected. System administrators may restrict the ciphers available on the server, forcing the client to use very strong encryption only (e.g. 256 bit ciphers).

Requirements

Support for encrypted connections between the Vitalware client and server requires the following software versions:

- Texpress 8.2.009 or later
- Vitalware 2.2.01 or later
- TexAPI 6.0.02 or later
- TexJDBC 0.9.6 or later

If any software is earlier than the version listed above, Vitalware will drop back to using unencrypted connections. In order to use encrypted connections your System Administrator must create the required keys (public/private) and public digital certificate and install them on the Vitalware server. The CA certificates may also need to be installed on the Vitalware client.

Vitalware server setup

The default installation of the Vitalware server does not have encrypted connections enabled. The following files need to exist to enable encrypted connections:

`server.key`	Vitalware server's private key
`server.crt`	Vitalware server's public digital certificate
`ciphers`	list of ciphers the server will support

These files must be placed in a directory called `certs` under the Texpress `etc` directory. For a standard Vitalware installation this corresponds to:

`$EMUHOME/texpress/8.2/etc/certs`.

If the files are not found, Vitalware will not attempt to use encrypted connections.

See *Generating certificates* (page 9) for details on how to create `server.key` and `server.crt`.

See *Configuring ciphers* (page 19) for details about configuring `ciphers`.

Important!

The permissions on the `server.key` file should restrict access to read-only by user `root`. All other permissions should be disabled, that is the file owner should be `root` and the permissions should be `r--------`. If these permissions are not set, it is possible that someone may access the file and so compromise the integrity of the system!

As described in *Requirements* (page 4), the Vitalware server will drop back to an unencrypted connection if versions earlier than 2.2.01 of the Vitalware client are used. If you want to enforce encrypted connections the `-s` option should be added to the `texserver` command configured in `inetd/xinetd/svcs`.

For example, the following `inetd` entry will accept encrypted connections only:

```
vwclient stream tcp nowait root /home/vw/client/bin/vwrun vwrun
texserver -avw -i -L -t60 -s
```

Vitalware client setup

The Vitalware client needs to verify the Vitalware server's public digital certificate for an encrypted connection to be established. In order to verify the certificate the client must be able to locate a valid CA (Certificate Authority) certificate for the Issuer of the server's certificate. In the case of a certificate chain this must be the Issuer of the first or "root" certificate. There are two locations used to hold CA certificates:

1. The first is on the Vitalware server and is used by programs using either TexAPI or `texapi.pm` (a perl based interface to TexAPI). These programs include web services.
2. The second is on the Vitalware client's machine and is used by the Windows client.

CA certificates stored on the Vitalware server must be placed in the `$EMUPATH/etc/certs` directory. The certificates should be stored in files with a `.crt` extension. CA bundles are also supported. A bundle is simply the concatenation of a number of certificates into one file. An optional `ciphers` file may also exist in the certificates directory. If it exists, it should list the ciphers the Vitalware client is willing to support.

See *Configuring ciphers* (page 19) for details about configuring `ciphers`.

CA certificates required by the Vitalware Windows client should be stored in the `certs` directory under the location where the Vitalware executable (`vw.exe`) is installed. As with CA certificates stored on the Vitalware server, the files must have a `.crt` extension and CA bundles are supported. A `ciphers` file may also be supplied defining the ciphers the client is willing to use.

TexJDBC setup

As with the Vitalware client, TexJDBC needs to be able to verify the Vitalware server's public digital certificate. The required CA certificates must be stored in an accessible Java Key Store (JKS). The system key store is located at `$JAVA_HOME/jre/lib/security/cacerts`. A key store may have a password associated with it. The password allows the integrity of the stored certificates to be checked when they are accessed. The password is not required to access the key store. The location of the key store used may be altered by setting the following system properties:

`javax.net.ssl.trustStore`

The location of the Java Key Store file containing the CA certificates to use for verifying the server's certificate.

`javax.net.ssl.trustStorePassword`

The password to use to check the integrity of the Java Key Store.

For example, if you want to use a key store located at `/home/vw/etc/certs/cacerts` with a password of `vwstore`, you could invoke java using:

```
java -Djavax.net.ssl.trustStore=/home/vw/etc/certs/cacerts -
Djavax.net.ssl.trustStorePassword=vwstore -jar application.jar
```

The location and password of the key store may also be specified using the `trustStore` and `trustStorePassword` connection properties:

```
Properties props = new Properties();
props.setProperty("trustStore", "/home/vw/etc/certs/cacerts");
props.setProperty("trustStorePassword", "vwstore");
...
Connection conn =
DriverManager.getConnection("jdbc:texpress:socket", props);
```

The `keytool` command should be used to import a CA certificate into a java key store:

```
keytool -importcert -alias alias -file certfile -storetype JKS -
keystore keystore
```

where:

alias is an arbitrary unique name used to define the certificate within the key store

certfile is the file containing the CA certificate

keystore is the location of the key store file into which the certificate is imported

If *keystore* does not exist, a new key store is created. You will be prompted for the password if the key store already exists, otherwise you will be asked to set the password for the key store created.

To list the certificates in a key store use:

```
keytool -list -v -keystore keystore
```

where *keystore* is the location of the key store file.

SECTION 2

Generating certificates

In this section we will look at how to generate a private/public key pair and how the public digital signature is created for the public key. As mentioned earlier Vitalware provides support for the following public certificates:

- **Self signed**
- **Root signed**
- **Chain signed**

Each of these types will be examined and appropriate commands provided for OpenSSL (via the `openssl` command).

Self signed

A self signed certificate is one where the certificate Issuer is the same as the certificate Subject. In other words the certificate is used to verify itself. In order for the certificate to be trusted the certificate must be included with the client CA certificates. Self signed certificates are used when you only need one certificate (for example if you only have one Vitalware server and all clients connect to that server). As the certificate is self signed it has not been verified by an external agency and so should be used for internal use only. The steps required to generate the required files are:

Step 1: Generate a private key

Create your private key. The key is a 1024 bit RSA key stored in PEM (Privacy Enhanced Mail, a Base64 encoding of the key) format. It is readable as ASCII text:

```
openssl genrsa -out server.key 1024
Generating RSA private key, 1024 bit long modulus
..++++++
...++++++
e is 65537 (0x10001)
```

The file `server.key` now contains your private key. Remember to keep it safe!

Step 2: Generate public digital certificate

Using the private key generated in the first step the public digital certificate is generated. A number of questions will be asked as part of the creation process. It is important that the Common Name (CN) is set to the full host name of your Vitalware server machine (e.g. `vw.institution.org`). Support for wild card host names is provided by replacing any leading component of the name with an asterisk (e.g `*.institution.org`, or `*.org`):

```
openssl req -new -x509 -key server.key -out server.crt -days 1095
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:Victoria
Locality Name (eg, city) []:Melbourne
Organization Name (eg, company) [Internet Widgits Pty Ltd]:KE
Software Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:*.mel.kesoftware.com
Email Address []:info@mel.kesoftware.com
```

The resulting public digital certificate will be stored in PEM format in `server.crt`.

You can view the contents of the public certificate using:

```
openssl x509 -text -in server.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f5:02:b4:7d:c3:5b:ad:a7
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=Victoria, L=Melbourne, O=KE Software Pty
Ltd, CN=*.mel.kesoftware.com/emailAddress=info@mel.kesoftware.com
        Validity
            Not Before: Nov 19 11:47:46 2010 GMT
            Not After : Nov 18 11:47:46 2013 GMT
        Subject: C=AU, ST=Victoria, L=Melbourne, O=KE Software Pty
Ltd, CN=*.mel.kesoftware.com/emailAddress=info@mel.kesoftware.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c4:c9:0f:04:8f:cd:98:5f:d9:c6:3b:00:54:b2:
                    88:07:9b:06:4c:ea:f2:41:74:a3:68:7d:16:2a:de:
                    cf:bb:cf:73:d5:97:f2:d8:4e:38:b1:7d:a8:94:48:
                    5b:4a:fd:92:3b:45:8c:1b:ce:85:e5:18:2e:c1:60:
                    db:4d:09:32:46:72:b4:a3:f1:f8:ab:96:4a:db:a5:
                    4c:32:6d:83:ee:f9:02:4e:8f:f1:8b:ba:b4:62:b6:
                    29:00:97:fb:3b:06:73:a2:56:5f:04:2c:79:3e:2e:
                    f8:1b:eb:f5:8b:a6:cf:6b:56:bd:74:16:cb:53:a6:
                    91:dd:ec:af:7a:77:40:b0:e5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:

F6:72:9C:A4:91:C2:E2:51:70:26:05:FE:06:C3:E4:E9:4F:AF:A0:D5
            X509v3 Authority Key Identifier:

keyid:F6:72:9C:A4:91:C2:E2:51:70:26:05:FE:06:C3:E4:E9:4F:AF:A0:D5
                DirName:/C=AU/ST=Victoria/L=Melbourne/O=KE
Software Pty
Ltd/CN=*.mel.kesoftware.com/emailAddress=info@mel.kesoftware.com
                serial:F5:02:B4:7D:C3:5B:AD:A7

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
        4c:89:a2:57:d2:3b:3a:11:70:63:41:56:4e:b6:36:8e:28:c5:
        29:d7:7d:22:86:c4:43:90:4f:74:d1:31:32:7f:39:d8:f3:20:
        80:05:53:99:cd:17:28:b8:16:3b:a3:9a:84:ae:2c:08:f5:b0:
        11:6a:d5:ba:42:81:9d:e7:36:8f:39:9d:b4:15:13:52:23:fc:
        37:f6:5c:88:39:f9:9b:d1:e0:06:82:3f:e2:56:a3:f3:83:55:
        4d:8b:7c:69:a3:bc:fb:3a:66:18:f2:07:67:bc:39:54:28:c3:
        eb:3e:5c:d9:89:d8:ea:c7:d2:c4:fe:87:ee:24:e0:ce:c0:4f:
        d1:e7
-----BEGIN CERTIFICATE-----
MIIDtTCCAx6gAwIBAgIJAPUCtH3DW62nMA0GCSqGSIb3DQEBBQUAMIGZMQswCQYD
VQQGEwJBVTERMA8GA1UECBMIVmljdG9yaWExEjAQBgNVBAcTCU1lbGJvdXJuZTEc
```

```
MBoGA1UEChMTS0UgU29mdHdhcmUgUHR5IEx0ZDEdMBsGA1UEAxQUKi5tZWwua2Vz
b2Z0d2FyZS5jb20xJjAkBgkqhkiG9w0BCQEWF2luZm9AbWVsLmtlc29mdHdhcmUu
Y29tMB4XDTEwMTExOTExNDc0NloXDTEzMTExODExNDc0NlowgZkxCzAJBgNVBAYT
AkFVMREwDwYDVQQIEwhWaWN0b3JpYTESMBAGA1UEBxMJTWVsYm91cm5lMRwwGgYD
VQQKExNLRSBTb2Z0d2FyZSBQdHkgTHRkMR0wGwYDVQQDFBQqLm1lbC5rZXNvZnR3
YXJlLmNvbTEmMCQGCSqGSIb3DQEJARYXaW5mb0BtZWwua2Vzb2Z0d2FyZS5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMTJDwSPzZhf2cY7AFSyiAebBkzq
8kF0o2h9Firez7vPc9WX8thOOLF9qJRIW0r9kjtFjBvOheUYLsFg200JMkZytKPx
+KuWStulTDJtg+75Ak6P8Yu6tGK2KQCX+zsGc6JWXwQseT4u+Bvr9Yumz2tWvXQW
y1Omkd3sr3p3QLDlAgMBAAGjggEBMIH+MB0GA1UdDgQWBBT2cpykkcLiUXAmBf4G
w+TpT6+g1TCBzgYDVR0jBIHGMIHDgBT2cpykkcLiUXAmBf4Gw+TpT6+g1aGBn6SB
nDCBmTELMAkGA1UEBhMCQVUxETAPBgNVBAgTCFZpY3RvcmlhMRIwEAYDVQQHEwlN
ZWxib3VybmUxHDAaBgNVBAoTE0tFIFNvZnR3YXJlIFB0eSBMdGQxHTAbBgNVBAMU
FCoubWVsLmtlc29mdHdhcmUuY29tMSYwJAYJKoZIhvcNAQkBFhdpbmZvQG1lbC5r
ZXNvZnR3YXJlLmNvbYIJAPUCtH3DW62nMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcN
AQEFBQADgYEATImiV9I7OhFwY0FWTrY2jijFKdd9IobEQ5BPdNExMn852PMggAVT
mc0XKLgWO6OahK4sCPWwEWrVukKBnec2jzmdtBUTUiP8N/ZciDn5m9HgBoI/4laj
84NVTYt8aaO8+zpmGPIHZ7w5VCjD6z5c2YnY6sfSxP6H7iTgzsBP0ec=
-----END CERTIFICATE-----
```

Step 3: Installing the files

We now have the two files we require:

- `server.key` - private key (must be kept safe)
- `server.crt` - self signed public digital certificate

On the Vitalware server these two files should be placed in the directory `$TEXHOME/etc/certs` where `$TEXHOME` contains the location where Texpress is installed. Suitable permissions should be set on the private key file:

```
mv server.key server.crt $TEXHOME/etc/certs
chmod 644 $TEXHOME/etc/certs/server.crt
su root
Password:
chown root $TEXHOME/etc/certs/server.key
chmod 400 $TEXHOME/etc/certs/server.key
exit
```

Next, the public certificate should be stored on the Vitalware server for use by API based programs (TexAPI and texql.pm):

```
cp $TEXHOME/etc/certs/server.crt $EMUPATH/etc/certs
chmod 644 $EMUPATH/etc/certs/server.crt
```

Finally, on Vitalware Windows client machines the `server.crt` file must be placed in a directory called `certs` in the same location as the Vitalware executable (`vw.exe`).

Now that all the required files are in the right place it is possible to connect using encrypted connections. As mentioned earlier, the `-s` option for `texserver` (page 5) may be used to enforce secure connections.

Root signed

A root signed certificate is a public digital certificate created and verified by an external entity. You forward a certificate request to the external entity and they return the signed public digital certificate. Root entities distribute their CA certificates (really just a special form of self signed certificate) for all to use, allowing any certificate signed by them to be verified. Root signed certificates are used when you need a verifiable certificate for external use.

The steps required to generate the required files are:

Step 1: Generate a private key

Create your private key. The key is a 1024 bit RSA key stored in PEM (Privacy Enhanced Mail, a Base64 encoding of the key) format. It is readable as ASCII text:

```
openssl genrsa -out server.key 1024
Generating RSA private key, 1024 bit long modulus
..++++++
...++++++
e is 65537 (0x10001)
```

The file `server.key` now contains your private key.

Step 2: Generate a certificate signing request

We use the private key generated in the first step to create a certificate signing request (CSR). The file generated will contain the Subject information without an Issuer being assigned, that is a certificate that has not yet been signed. The resulting file, `server.csr` is then sent to an external entity for signing (e.g. Verisign).

```
openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:Victoria
Locality Name (eg, city) []:Melbourne
Organization Name (eg, company) [Internet Widgits Pty Ltd]:KE
Software Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:*.mel.kesoftware.com
Email Address []:info@mel.kesoftware.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

```
An optional company name []:
```

Once the external entity has verified the Subject information in the request they will generate a public digital certificate and return it to you. You should save the certificate in a file called `server.crt`.

You can view the contents of the certificate signing request using:

```
openssl req -in server.csr -noout -text
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=Victoria, L=Melbourne, O=KE Software Pty
Ltd, CN=*.mel.kesoftware.com/emailAddress=info@mel.kesoftware.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c4:c9:0f:04:8f:cd:98:5f:d9:c6:3b:00:54:b2:
                    88:07:9b:06:4c:ea:f2:41:74:a3:68:7d:16:2a:de:
                    cf:bb:cf:73:d5:97:f2:d8:4e:38:b1:7d:a8:94:48:
                    5b:4a:fd:92:3b:45:8c:1b:ce:85:e5:18:2e:c1:60:
                    db:4d:09:32:46:72:b4:a3:f1:f8:ab:96:4a:db:a5:
                    4c:32:6d:83:ee:f9:02:4e:8f:f1:8b:ba:b4:62:b6:
                    29:00:97:fb:3b:06:73:a2:56:5f:04:2c:79:3e:2e:
                    f8:1b:eb:f5:8b:a6:cf:6b:56:bd:74:16:cb:53:a6:
                    91:dd:ec:af:7a:77:40:b0:e5
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: sha1WithRSAEncryption
        ae:e0:68:b8:fe:56:53:5e:f4:f4:e0:8d:19:2c:62:ee:ee:83:
        01:d2:8d:55:d0:2d:18:b8:18:0a:f2:5b:c4:a5:da:75:fd:ca:
        87:69:cd:3f:2e:7c:9e:a2:c2:b7:b1:4a:bd:85:2e:24:84:8d:
        cc:81:64:9d:0c:a4:ad:c4:c5:54:4d:cf:22:dc:08:51:3f:ed:
        6d:45:d6:91:e3:a6:c0:7e:2e:f0:0f:9e:be:70:ef:6a:f8:2c:
        93:59:8d:90:ca:23:c4:07:f9:ae:2c:09:03:fd:cf:43:d6:b7:
        8c:2e:48:96:28:98:5c:c3:e8:66:55:b3:4a:8d:bb:c8:d0:bb:
        c8:41
```

Step 3: Installing the files

The process for installing the two files `server.key` (private key) and `server.crt` (public certificate) is exactly the same as for a self signed certificate (page 12).

Chain signed

A chain signed certificate is a certificate that is not self signed and is not root signed. In order for the certificate to be verified, the Issuer of the certificate is verified, then the Issuer of the Issuer certificate is verified and so on until either a self signed or root signed certificate is encountered. If the top certificate is root signed, then the chain signed certificate has the same level of verification as if the certificate had been root signed directly. If the top certificate is self signed, then the level of verification is the same as for any other self signed certificate. Chain signed certificates are used where you will be generating multiple certificates and you only want to distribute one CA certificate to verify them all. The only CA required is the top level self signed or root signed certificate.

The steps below outline how to produce a self signed CA certificate that can then be used to sign all other certificates generated. If you require a root signed CA certificate, you need to generate a certificate signing request (as per the previous section) and have the external entity generate the CA certificate.

Step 1: Generate a self signed CA certificate

The first step creates a self signed CA certificate. The CA certificate is the "root" certificate used to sign (and hence verify) all other certificates we generate. The public digital certificate of the CA certificate needs to be installed on client machines. We only need to generate the CA certificate once.

```
echo "01" > ca.srl
openssl req -new -x509 -nodes -extensions v3_ca -keyout ca.key -out ca.crt -days 365
Generating a 1024 bit RSA private key
................................++++++
.........++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:Victoria
Locality Name (eg, city) []:Melbourne
Organization Name (eg, company) [Internet Widgits Pty Ltd]:KE
Software Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:KE Software CA certificate
Email Address []:info@mel.kesoftware.com
```

Step 2: Generate a private key

Once we have the CA certificate we can generate a new certificate. The first step is to generate the private key:

```
openssl genrsa -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.++++++
..++++++
e is 65537 (0x10001)
```

Step 3: Generate a certificate signing request

We use the private key generated in the previous step to create a certificate signing request (CSR). The file generated will contain the Subject information without an Issuer being assigned, that is a certificate that has not been signed. Make sure `Common Name` is set to the host name of your Vitalware server.

```
openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:Victoria
Locality Name (eg, city) []:Melbourne
Organization Name (eg, company) [Internet Widgits Pty Ltd]:KE
Software Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:*.mel.kesoftware.com
Email Address []:info@mel.kesoftware.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Step 4: Sign the certificate with your CA certificate

Using our CA certificate we sign the CSR to produce our public digital certificate:

```
openssl x509 -CA ca.crt -CAkey ca.key -CAserial ca.srl -req -in server.csr -out server.crt -days 365
Signature ok
subject=/C=AU/ST=Victoria/L=Melbourne/O=KE Software Pty
Ltd/CN=*.mel.kesoftware.com/emailAddress=info@mel.kesoftware.com
Getting CA Private Key
```

Step 5: Creating the certificate chain

A certificate chain is simply the concatenation of all the signing public certificates from the certificate just generated to the root certificate (there may be any number of intermediate certificates). As we only have one CA in the chain in this example we do not need to concatenate the certificates, however if more than one CA has been used all certificates in the chain must be placed in one file. While not required for this example we will concatenate the certificates anyway (it does not hurt):

```
cat server.crt ca.crt > chain.crt
```

Step 6: Installing the files

First we install the private key and chain certificates on the Vitalware server:

```
mv server.key $TEXHOME/etc/certs/server.key
mv chain.crt $TEXHOME/etc/certs/server.crt
chmod 644 $TEXHOME/etc/certs/server.crt
su root
Password:
chown root $TEXHOME/etc/certs/server.key
chmod 400 $TEXHOME/etc/certs/server.key
exit
```

Next, the public CA certificate should be stored on the Vitalware server for use by API based programs (TexAPI and texql.pm):

```
cp ca.crt $EMUPATH/etc/certs
chmod 644 $EMUPATH/etc/certs/ca.crt
```

Finally, on Vitalware Windows client machines the `ca.crt` file must be placed in a directory called `certs` in the same location as the Vitalware executable (`vw.exe`).

Now that the CA certificate is installed on the Vitalware clients there is no need to add any further files when generating new certificates signed by the same CA certificate.

SECTION 3

Configuring ciphers

When an encrypted connection is formed the client and server negotiate to determine the highest level of encryption on which they can both agree. A list of ciphers may be set for the server and/or client allowing System Administrators to enforce a certain level of encryption. As the strongest cipher is chosen as part of the connection negotiation it is not necessary to restrict the list of ciphers used in most cases. The ciphers to use can be set at three levels:

- Server
- TexAPI/texapi.pm
- TexJDBC

Server

The list of ciphers the server will support is found in a file called `ciphers`. The file is located in the `$TEXHOME/etc/certs` directory. The format of the file is a colon separated list of cipher names. For details on what ciphers are supported and the exact format of the setting see the Ciphers section of the OpenSSL documentation.

For example, to enforce the use of MD5 ciphers the following cipher file could be used:

```
#
#  The allowable ciphers for use between the client and server are
#  defined by the last line in this file. See ciphers(1) in
OpenSSL
#  (http://www.openssl.org/docs/apps/ciphers.html) for the format
of
#  statement detailing the ciphers to use.
#
MD5
```

TexAPI / texapi.pm

To set the ciphers supported by the Windows client and client side programs using `texapi.pm` the `TEXCIPHERS` environment variable should be used. For example, to enforce the use of MD5 ciphers the following setting could be used:

```
TEXCIPHERS="MD5"
export TEXCIPHERS
```

It is also possible to set the ciphers when using TexAPI directly. The `Ciphers` member of the `TEXSESSINFO` structure may be used:

```
TEXSESSINFO info;
TEXSESSION  session;

...
info.Ciphers = "MD5";
...
TexSessConnect(&info, &session);
...

For texapi.pm the Ciphers key is used:

my $session = ke::texapi->new(
{
        ...
        Ciphers => 'MD5',
        ...
});
```

TexJDBC

When using TexJDBC the `ciphers` connection property may be set to restrict the ciphers used for a connection:

```
Properties props = new Properties();

props.setProperty("ciphers", "MD5");
...

Connection conn =
DriverManager.getConnection("jdbc:texpress:socket", props);
```

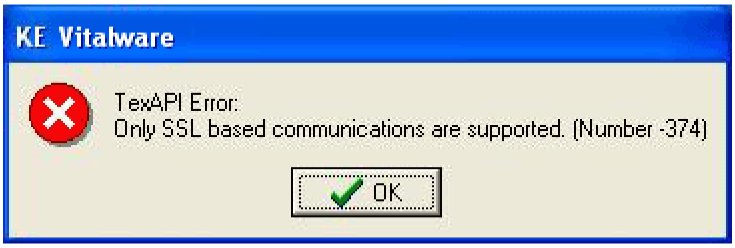
SECTION 4

Common Errors

When configuring the use of encrypted connections a number of common errors may occur. In this section a description of these errors is given, along with possible solutions.

Client does not support SSL

The System Administrator may configure the Vitalware server to accept encrypted connections only. The `-s` option for `texserver` will force the server to only accept connections where encryption is enabled. If the Vitalware client is a version prior to 2.2.01, then encryption is not supported and the following error message is displayed:

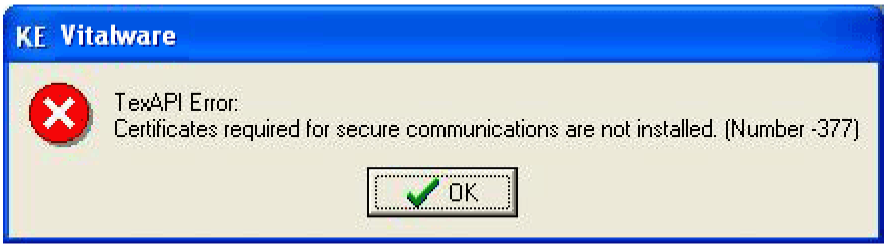


TexAPI Error:
Only SSL based communications are supported. (Number -374)

The solution is to upgrade the Vitalware client to version 2.2.01 or greater.

Server certificates not installed

If the System Administrator has turned on the `-s` option for `texserver`, thus forcing encrypted connections, and the server side certificates (`server.crt` and `server.key`) are not installed, the following error message is displayed:

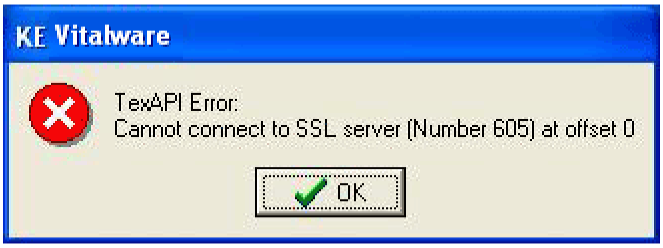


TexAPI Error:
Certificates required for secure communications are not installed. (Number -377)

The solution is to generate the private key (`server.key`) and public digital certificate (`server.crt`) and place them in the correct location (`$TEXHOME/etc/certs`).

Server/Client protocol error

The initiation of an encrypted connection involves a handshake between the client and server programs. If an error occurs as part of the handshake, the following message is displayed:



TexAPI Error:
Cannot connect to SSL server (Number 605) at offset 0

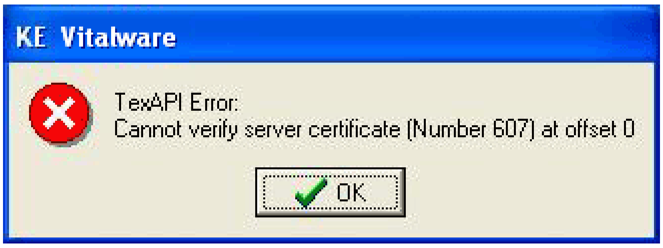
There are many reasons why a protocol error may occur. The most common are:

- The private key file `$TEXHOME/etc/certs/server.key` cannot be read. The error implies the contents of the private key file are corrupted.
- The public digital certificate file `$TEXHOME/etc/certs/server.crt` cannot be read. The error implies the contents of the public certificate file are corrupted.
- A private key/public certificate mismatch. The public digital key `server.crt` was not generated using the private key found in `server.key`. Either the private key or public certificate is incorrect.
- An acceptable cipher cannot be found. The client and server cannot agree on a cipher to use for the connection encryption. The server ciphers file should be altered to match a client cipher or vice versa.

Server side debugging may be required to determine the exact cause of the error. The Texpress debug flags `s15,16` will output the reason for the protocol error. For details on how to set Texpress debug flags please contact KE Software support.

Cannot verify certificate

In order for the client to verify the server's certificate the client must have a copy of the server's top level public CA certificate. If the top level certificate is not installed on the client, the following error is displayed:

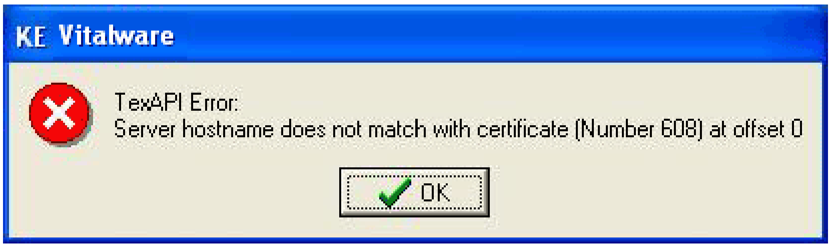


TexAPI Error:
Cannot verify server certificate (Number 607) at offset 0

The solution is to install the top level CA certificate on the client. The certificate should be placed in a directory called `certs` located in the same place as the Vitalware client executable.

Bad host name

The *Common Name* field of the server's public digital certificate must contain the host name of the Vitalware server. Wild cards are supported using the asterisk character. If the *Common Name* field of the server's certificate does not match the host name to which the client is connected, the following error is displayed:



TexAPI Error:
Server hostname does not match with certificate (Number 608) at offset 0

The solution is to fix up your DNS so that the host name of the Vitalware server matches the host name stored in the server's certificate. If this is not possible, a new server certificate should be generated with the correct host name.

Index

Vitalware Documentation

Vitalware Read-Only Modes

Document Version 1

Vitalware Version 2.2.01

www.kesoftware.com

Contents

SECTION 1

Vitalware read-only modes

Overview

KE Vitalware 2.2.01 adds two new Registry entries to make all or part of the system read-only. These entries allow system administrators to "turn off" record insertions and updates while still allowing the system to be operational. Prior to Vitalware 2.2.01 it was possible to make the back-end tables read-only (and it still is), however the Vitalware client did not reflect the read-only nature of the table in the module interface: it was still possible to select the menu entry to insert a new record, but an error would then be displayed as the table was read-only. The `ReadOnly` Registry entries control the Vitalware client while still leaving the back-end tables operational. This means that the menu entry to insert a new record is now disabled if the table or system is marked as read-only.

The system wide read-only entry ensures that data is not modified in any modules, including the Vitalware Registry. Some uses for this entry include:

- Disabling data changes while the system is upgraded.
- Providing read-only access for certain users / groups.
- Allowing data loads to be checked before going live.

The table specific read-only entry allows individual tables, or all tables, to operate in a read-only state. If a table is read-only, data insertions and modifications are not permitted. The only exception is that changes to the Vitalware Registry are permitted even if the eregistry table is set to read-only. Possible uses for this entry include:

- Disabling data changes to a single table (as it may require maintenance).
- Providing read-only access to a table for certain users/groups.
- Allowing Registry based operations (e.g create a new sort) while the data is read-only.

The addition of the two `ReadOnly` Registry entries provides system administrators with the ability to control access to data stored in Vitalware at a system wide and table specific basis.

The `ReadOnly` Registry entries work alongside existing Vitalware Registry entries. For example, if a table is designated as read/write by a `ReadOnly` Registry entry, a user still needs the `daInsert` operational privilege to be able to create records. The `ReadOnly` entries do not override existing Registry entries to provide more privileges than would be the case if the entry was not set.

System ReadOnly Registry setting

The format of the system `ReadOnly` Registry setting is:

`System|Setting|ReadOnly|`*value*

`Group|Default|Setting|ReadOnly|`*value*

`Group|`*group*`|Setting|ReadOnly|`*value*

`User|`*user*`|Setting|ReadOnly|`*value*

where:

value	is either `true` (read-only functionality is enabled) or `false` (read-only functionality is not enabled).

The system `ReadOnly` entry provides the following functionality:

- Data in all modules cannot be updated or created.
- Operations involving updates to tables are disabled, namely:
  - Creating groups (uses egroups table).
  - Manipulating the records in groups (uses egroups table).
  - Creating batch exports (uses eschedule table).
  - Running batch exports (uses eexports table).
  - Create task templates (uses etemplates table).
  - Change help contents (uses efieldhelp table).
- Operations that update the Registry are disabled, namely the creation, modification or deletion of any resources:
  - List Settings
  - Default Values
  - Ditto Record
  - Record Recall
  - Record Template
  - Reports
  - Shortcut Settings
  - Sorts
- Operations that update the Registry when performed, but only to record the entry selected are not disabled, namely:
  - List Settings
  - Page View
  - Query Default Values
  - Reports
  - Shortcut Settings
  - Sorts
  - Ad-hoc Sorts

The system ReadOnly entry may be used to limit access on a group or user basis. For example, if users in group QA may not update any data, the following entry may be used:

```
Group|QA|Setting|ReadOnly|true
```

Entries may also be used in combination to achieve the desired effect. For example, if all users except those in group Admin should have read-only access, the following entries may be used:

```
System|Setting|ReadOnly|true
Group|Admin|Setting|ReadOnly|false
```

Table ReadOnly Registry setting

The format of the table `ReadOnly` Registry setting is:

```
Group|Default|Table|Default|ReadOnly|value
Group|Default|Table|table|ReadOnly|value
Group|group|Table|Default|ReadOnly|value
Group|group|Table|table|ReadOnly|value
User|user|Table|Default|ReadOnly|value
User|user|Table|table|ReadOnly|value
```

where:

value is either `true` (all operations that would result in record creation or updates in the specified table are disabled) or `false` (update operations are allowed, subject to other permissions, e.g. operational permissions).

The one exception to the rule is the Registry table (eregistry). If the Registry is made read-only, only explicit changes, that is changes made from the Registry module, are disabled. All implicit changes (e.g. adding a new sort definition) are still permitted. Hence the creation, modification or deletion of any resources is permitted, namely:

- List Settings
- Default Values
- Ditto Record
- Record Recall
- Record Template
- Reports
- Shortcut Settings
- Sorts

Using the first form of the table `ReadOnly` Registry entry:

```
Group|Default|Table|Default|ReadOnly|true
```

disables data creation and updates in all tables while still permitting most operational commands (e.g. report creation). When compared with the system `ReadOnly` setting, the table setting restricts only data modifications, rather than data and operational functionality.

User/group based variants of the table Registry entry may be used to restrict access to a select number of individuals. For example, if users in group `QA` are not allowed to create or modify records in the registration modules, the following Registry entry may be used:

```
Group|QA|Table|ebirths|ReadOnly|true
```

As with the system ReadOnly entry, table based entries may be combined to produce the desired effect. For example, if users in group QA were only allowed to update the egroups, eschedule and eexports tables (allowing them to create groups and batch exports), then the following Registry entries may be used:

```
Group|QA|Table|Default|ReadOnly|true
Group|QA|Table|egroups|ReadOnly|false
Group|QA|Table|eschedule|ReadOnly|false
Group|QA|Table|eexports|ReadOnly|false
```

Combining system and table ReadOnly Registry entries

In general, it is not wise to mix system based and table based `ReadOnly` Registry entries. Consider the following settings:

```
Group|QA|Setting|ReadOnly|true
Group|QA|Table|eparties|ReadOnly|false
```

Which entry takes precedence? Can users in group `QA` write to the eparties table? The current implementation gives priority to the system entries over table based entries. So for the example above, users in group `QA` would not be able to write to the Parties module.

Examples

Example 1

Your Vitalware installation is to be upgraded to the next release of the software. While the upgrade is proceeding you would like users to be able to view data in the live system, but not make any changes as they will be lost when the upgraded system replaces the live system. The following Registry entry may be used:

```
System|Setting|ReadOnly|true
```

The above setting will disable all Vitalware functions that would result in any data changes. Users may still produce reports, sort records, etc., however data changes are not allowed.

Example 2

You have received an initial data load in Vitalware ready for checking. You do not want users to modify any data as they are only determining the integrity of the load. In order to make the reporting of issues easier, you would like users to be able to create groups so they can group records where the data does not look correct. The following Registry entries may be used:

```
Group|Default|Table|Default|ReadOnly|true
Group|Default|Table|egroups|ReadOnly|false
```

The disabling of the read-only status on egroups allows users to create and manipulate groups.

Example 3

You have created a duplicate version of your live system for access via the web. You do not want users to be able to alter any data in the web copy, however all other functionality should be available. The following Registry entries may be used:

```
Group|Default|Table|Default|ReadOnly|true
Group|Default|Table|egroups|ReadOnly|false
Group|Default|Table|eschedule|ReadOnly|false
Group|Default|Table|eexports|ReadOnly|false
Group|Default|Table|etemplates|ReadOnly|false
Group|Default|Table|efieldhelp|ReadOnly|false
```

The enabling of read/write access to the egroups, eschedule, eexports, etemplates and efieldhelp tables will allow full command functionality on a read-only system. You may want to exclude some of these tables if you want to further limit functionality. For example, removing efieldhelp will disable the updating of field level help.

Example 4

You are undertaking a clean up of the Lookup List table (eluts) so you would like to restrict updates to lookup lists to users in group `Admin`. The following Registry entries may be used:

```
Group|Default|Table|eluts|ReadOnly|true
Group|Admin|Table|eluts|ReadOnly|false
```

When the eluts table is read-only, users may not insert new values into the table. This means **all** lookup lists are treated as read-only. Users may find this annoying as it may make it difficult to save records.

Example 5

You have just received an allocation of staff to add field level help for all modules in the system. You do not want the staff to alter any other data apart from field level help. You have created a group called `Volunteers` and placed the allocated staff in the group. The following Registry entries may be used:

```
Group|Volunteers|Table|Default|ReadOnly|true
Group|Volunteers|Table|efieldhelp|ReadOnly|false
```

In order for the volunteers to be able to create the field level help for all modules, you would also need to allocate the `daEditHelp` operational privilege for group `Volunteers`.

Index