pitney bowes

Customer Engagement

EngageOne® Vault

Version 7.5 Service Pack 1

Vault Installation

Table of Contents

1 - The Vault suite of products

The components that make up Vault support storage, display, management and processing of composed documents in electronic environments.

In this section

Overview

Vault Server is the document repository and forms the hub of the Vault environment. This server component can compress, store and manage documents that have been composed in a wide range of output formats including those specifically produced by the Generate environment.

Mobile Vault is a Windows based component that allows documents to be read from a local copy of the document Vault.

Vault service

Vault Service family of products provide a comprehensive range of access and display mechanisms for documents stored in Vault. They are primarily aimed at users within the corporate environment; typically customer service or other front-line personnel.

Vault Service Client provides an intuitive, high performance Windows based interface to the documents stored within Vault. It is an executable that can be easily rolled-out to desktops as required. This can be supplemented with Vault Service Reprint Admin which administrators can use for compressed stream viewing and document export capabilities.

Rendering Engine allows users to build a customized interface to Vault and optionally allow you to integrate the document display function into an existing web server environment. This is based on a set of API functions that communicate with Vault server and return rendered documents as required.

Vault clients

The rollout of Vault client applications such as Vault Service Client, and Vault Service Reprint Admin involves installing the required modules from the distribution media onto the appropriate workstations.

Release content

Software for each supported platform is provided as a set of computer files. Within the files, there are Vault guides in PDF. The products supplied to you will depend on the Vault licenses you hold. Additionally, you will only be able to install products for which you have a valid license keycode. If you believe that you have not received the correct material or keycodes please contact your Vault supplier.

About Vault installations

This manual explains how to install Vault Server, and Vault clients. It is intended for system administrators and professional services personnel who are responsible for installing Vault.

These instructions assume that you are familiar with the Microsoft Windows and UNIX operating systems.

About server configurations

There are several server components in the Vault environment of which you may have one or many installed. Some servers have non-Vault software pre-requisites. The hardware supporting the server configuration should reflect your needs for performance and security but all machines need to be able to communicate freely using TCP/IP sockets. All installations will include:

- Vault Server which compresses, stores and manages the datastreams containing the documents required for archiving and display via one of the Vault Service Clients. Your installation may have one or many Vaults.
- In a Windows environment it is recommended that Windows Server Edition is used for Vault in order to achieve the security provisions and remote management features that are integral to these platforms. Ensure that you apply the most up to date service packs and security fixes.

Integrating with Generate

Many of the components that form Generate are optional and can reside on a range of platforms or servers which are not necessarily connected. As a result, file based methods of exchanging key information between the components are used particularly when Vault components are part of the environment.

License keycodes

The functions and features that are available with all Vault software are controlled by an XML license file. Your XML license file information will be supplied to you (separately from other release material) typically as an email attachment.

2 - Installing Vault and clients

This chapter provides information on installing and configuring your Vault environment. Vault modules in this release are supplied for use with Microsoft Windows and UNIX (Linux, AIX and Solaris) platforms.

In this section

Overview

In a Microsoft Windows environment it is recommended that Microsoft Windows Server Edition (64-bit) is used for Vault in order to achieve the security provisions and remote management features that are integral to these platforms. Ensure that you apply the most up to date service packs and security fixes.

Refer to **Software requirements** for supported operating systems.

Hardware requirements

The hardware and network configuration that supports the Vault environment is at the discretion of the individual installation.

The performance of servers that make up the environment is often a critical issue and you must ensure that the configuration is optimized to meet the expectations of your user community. Disk and memory requirements will depend on the size and complexity of your Vault applications and on what other software products are running. As with most software, the more resources you have available the more quickly and efficiently applications will run.

> **Note:** If a virtual environment is used, it is the responsibility of the customer to ensure that the virtual environment, in its entirety, is up to the demands of Vault software. Due to the variable nature of virtual environments, performance problems or concerns that result from using a virtual environment are the responsibility of the virtual environment administrators to manage such that sufficient host resources are allocated in order to ensure adequate business-level performance.

More detailed recommendations are available on the following website:

www.doc1supportnet.com

Testing

Correctly planning the hardware is a critical step in a successful Vault installation. This can be complicated by a number of factors:

- The wide variety of configurations in which Vault can be deployed.
- The wide variety of data types Vault can process and their varied characteristics.
- The difficultly in simulating the volume and load of a production environment.

To help refine your design it is highly recommended you set up a test environment to help model the characteristics of the production deployment.

For best results:

- Ensure you load samples of each of the data formats you intend to load in production.
- Load a good volume of data.
- Load data from different cycles, runs, and dates.

You need to determine:

- The approximate size of the all the data you intend to load in production.
- The number and type of Vault processes needed to handle the volume and load of production.
- The correct configuration settings for profiles, databases, and etc.
- The appropriate number of processors and memory for each server.
- Approximate network utilization between servers.

Processor architecture

Vault consists of a number of different components including a Microsoft Windows client and several server processes.

The Microsoft Windows client can run on practically any recent x86 processor including:

- Intel Core i3
- Intel Core i5
- Intel Core i7
- AMD A-Series
- AMD Phenom
- AMD Althon
- AMD Sempron

Vault server processes include:

- e2serverd (Vault Server)
- e2loaderd (Vault Loader)
- e2renderd (Vault Rendering Engine)
- e2routerd (Vault Router)
- indexerd (Vault Indexer as a Service)

These processes can run on the following architectures:

- x86 running Microsoft Windows or Linux
- SPARC running Solaris
- Power running AIX

Examples of these include:

- Nehalem based Intel Xeons
- Sandy Bridge based Intel Xeons
- AMD Opterons
- SPARC T3
- SPARC64 VII+
- UltraSPARC T2
- POWER7

Memory and processor cores

The memory and process cores needed on a Vault machine will depend on the number and type of Vault processes deployed there.

For an approximate estimate use the following values:

- e2loaderd: 1-2 cores, 1-2 GB
- e2serverd: 2-4 cores, 1-2 GB
- e2renderd: 4-8 cores, 2 GB
- e2routerd: 2-4 cores, 1 GB
- indexerd: 2-4 cores, 3-4 GB

A theoretical single machine deployment using one e2loaderd, e2serverd, and e2renderd should have 7-14 cores and 4-8 GB of memory.

A situation where there is more load typically involves adding more e2renderd instances often grouped together using e2routerd. You may have something such as the following:

Machine 1	**Machines 2 and 3**	**Machine 4**
e2loaderd e2serverd 3-6 cores 2-4 GB	2 x e2renderd 8-16 cores 4 GB	e2routerd 2-4 cores 1 GB

In general it is preferable to favor fewer, faster cores to more, slower cores.

Storage

Choosing a good storage layout is an important part to configuring Vault because:

- Vault installations usually store vast quantities of data.

- The storage system has a significant effect on the performance of Vault deployment.

While not a SQL server, choosing a storage system for Vault has many similar considerations.

The size of data normally requires a large number of drives which increases the potential rate of drive failures. You will need to choose some method of data protection such as using RAID 5 or RAID 10 drive arrays.

Vault server has a number of specific directories with performance consequences. Keeping these on isolated volumes is one way to increase the total number of I/O operations the server can handle at once.

Typically you would separate the directories into the following groups: index, data, and support. The index directory is used in searching for and linking to documents. It typically experiences very heavy random read and write disk I/O. The data directories consist of docdata and pagedata or storage, depending on the configuration. You normally have heavy sequential writes and very heavy random reads to this data. It is sometimes reasonable to further separate this area into even smaller volumes (either by docdata/pagedata or by storage/year). The remainder of the directories serve various purposes that are normally not as critical to performance.

Avoid

- Using storage that might not have a reliable connection to the server.
- Virus scanners, in particular real time virus scanners.
- Using a 32-bit operating system. A 64-bit operating system will generally have a higher threshold for kernel memory.

Network

Vault processes and clients communicate over TCP/IP. Often large transfers of data occur between machines:

- Transferring raw jobs to Vault server.
- Large Postscript headers.
- Documents with a large number of pages.

Aim for high bandwidth between servers (1 gigabit or even 10 gigabit capable servers connected via switches).

Use the test environment to model how much your deployment actually needs.

Avoid performance sapping issues such as:

- Bad cables (for example, crossed pairs)
- Duplex mismatches

Backup and recovery

Vault installations contain vast amounts of critical data. Ensure that there are plans in place to handle backup and recovery of the database.

The indexes that are at the root of the data must be backed up correctly. To ensure the proper consistency of the index backup, the process should make use of the `indexbackup.adm` flag file to create a consistent index copy to `server\backup` and then back up that copy.

> **Note:** Avoid storing the indexes on remote storage (redirection, NAS, iSCSI, etc.). The indexes are particularly sensitive to storage failures.

Backup programs must not lock Vault files or open them in exclusive mode. Doing so can prevent Vault processes from accessing your data properly.

Software requirements

Microsoft Windows support

Vault Service Clients (Vault Reprint Admin, and Mobile Vault) are supported on the following operating systems:

- Microsoft Windows 7 Service Pack 1
- Microsoft Windows 8

Vault and Render servers are supported on the following:

- Microsoft Windows Server 2008 R2 Service Pack 1
- Microsoft Windows Server 2008 Service Pack 2
- Microsoft Windows Server 2012 and Windows Server 2012 R2
- Microsoft Windows Server and Datacenter 2016 versions that support 32 bit applications.

> **Note:** that the Windows 2016 Nano Server does not support 32 bit applications.

The following is supported:

- 32-bit x86
- 64-bit x64 (runs as a 32-bit process)

It is highly recommended that you use a 64-bit operating system to host Vault. Vault servers can put considerable stress on the kernel resources of 32-bit versions of Microsoft Windows. This can be exacerbated by using antivirus software, the use of bad drivers or even large amounts of memory (which requires more kernel space for page tables).

Note: Vault is not supported on Itanium Processor Architecture.

UNIX support

Vault server and the Rendering Engine are supported on Linux, Solaris on SPARC, and AIX platforms. The requirements for each are as follows:

Linux

Linux distribution (32-bit or 64-bit x86) with kernel version greater than or equal to 2.6.18, and glibc version greater than or equal to 2.3. This includes Red Hat Linux and Suse Linux.

Note: Running on a 64-bit Linux version will require the installation of 32-bit compatibility libraries.

AIX

IBM AIX 6.1, and 7.1 with C++ runtime libraries (xlC.rte and xlC.aix50.rte filesets). The component xlC.aix61.rte must be at level 9.0.0.5 or higher. The AIX XL C/C++ runtime must be version 9.0.0.5 or higher. The xlC supplemental runtime for aix50 (xlC.sup.aix50.rte) at level 9.0.0.1 or higher.

To determine the current level of the installed xlC components, run the following AIX command: lslpp -l | grep xlC

Solaris on SPARC

Sun Solaris 10 (for 64-bit SPARC).

Microsoft Windows installation

There are three installation scripts, including the required script for installing the Vault server components and two optional scripts for installing Vault Windows service clients.

Note: Installing under microsoft windows requires administrator access rights.

These scripts run under Powershell 2.0 and later versions. Note that for Windows 7, Windows 8, Windows Server 2008 Windows Server 2012 and Windows Server 2016, Powershell is bundled with Windows.

Vault Windows installation material is contained in `InstallSet.zip` file. The actual filename will reflect the product name (Vault), the version number (7.2.0.22 for example), the word InstallSet, and a ZIP filetype. Extract the contents of this ZIP file to your chosen install directory, for example c:\vault-temp, at this point the following files will be available:

- `install-admin.bat` - the Windows admin service client
- `install-desktop.bat` - the Windows desktop service client
- `install-vault.bat` - Vault system servers components

Installing Vault server (required)

There are four files required for Vault system installation:

- `install-libs.psm1` - the powershell common module/functions of the installation
- `install-vault.ps1` - the powershell main script for the installation
- `install-vault.bat` - the main install batch command file
- `install-vault-config.txt` - the install configuration

Configuring Vault installation

The parameters used to configure Vault installation are specified in `install-vault-config.txt`.

Refer to the information below for details of this configuration file:

```
#####################################################
#parameters for licence
#
# description of parameters:
#
# [licence_path] : is for the licence info root directory,
#     should include the public key file and licence file;
#
# [licence_file] : is for the licence file name
#
# [Environment] : is for the evironment ID, should match the value inside
 licence file
#
# If you don't know the Environment ID from a license file, you can use
 the
# scripts of (print-lic.bat) to display the license information:
# for example: print-lic.bat <full-path-license-file>
#

licence_path=c:\\licence
licence_file=VaultKeys.xml
Environment=1
```

```
#####################################################
#parameters for source files
#
# description of parameters:
#
# [source_path] is for the root path/dir which is including
#     Vault system files/resources/sub-directories which are used to
install,
#     the default value is the current working directory if not setup
or empty,
#

#source_path=C:\\test\\9.9M9p0012-InstallSet-Vault


#####################################################
#parameters for installation
#
# description of parameters:
#
# [program_group] : is for the group entry name in Windows START menu,
 default is "Vault"
#
# [install_path] : is for the target/destination directory to install
Vault system
#     if not defining install_path, or it is empty
#     then use the path of "C:\\Program Files (x86)\\PBBI CCM\\Vault"
#
# [install_servers] : is for settings of servers components, can be
below values
#     "server" : is for server components (e2serverd, e2laoderd,
indexerd);
#     "render" : is for e2renderd (Rendering engine) components;
#     "router" : is e2routerd components;
#
# [install_clients] : is for windows client components, can be below
values
#    "admin"   : is for Vault Windows Admin service client
#    "desktop" : is for Vault Windows Desktop service Client
#

program_group=Vault

#install_path=c:\\test\\install

install_servers=server, render, router

install_clients=admin, desktop


#####################################################
#parameters for IP addresses and ports
```

```
#
# description of parameters:
#
# [ip_server] : is for the ip address of e2serverd
# [ip_loader] : is for the ip address of e2loaderd
# [ip_render] : is for the ip address of e2renderd
# [ip_router] : is for the ip address of e2routerd
#
# [port_server] : is for the port number of e2serverd
# [port_loader] : is for the port number of e2loaderd
# [port_render] : is for the port number of e2renderd
# [port_router] : is for the port number of e2routerd
#

ip_server=127.0.0.1
ip_loader=127.0.0.1
ip_render=127.0.0.1
ip_router=127.0.0.1

port_server=6001
port_loader=6002
port_render=6003
port_router=6004


##################################################
#install server/render/router as a service
#
# description of parameters:
#
# [install_service] : set up if install servers(e2serverd, e2loaderd,
indexerd,
# e2renderd, e2routerd) as Windows service;
#     true : install the servers as a windows service; otherwise, not
install them
# as services
#     the default value for [install_service] is true if don't setup or
 empty
#
# [stop_running_service] : set up if stop running service of servers
(e2serverd,
# e2loaderd, indexerd, e2renderd, e2routerd)
#     true : stop running services if applicable; otherwise not stop
the running
# service, the users need manually stop it;
#     the default value for [stop_running_service] is false (not stop
the running
# service) if don't setup or empty;
#
# [delete_existing_service] : set up if delete the existing services
for
# (e2serverd, e2loaderd, indexerd, e2renderd, e2routerd)
#     true : delete the existing services; otherwise not delete the
```

```
existing
# services, the users need manually delete them;
#     the default for [delete_existing_service] is false (not delete
the existing
# services) if don't setup or empty;
#

install_service=true

stop_running_service=true

delete_existing_service=true


####################################################
#parameters for Windows service clients
#
# description of parameters:
#
# [primaryHost_admin] : is for the host name / ip address of admin
service client
# [serviceIP_admin]   : is for the ip address of server which the client
 will connect to
# [servicePort_admin] : is for the port number of server which the client
 will connect to
#
# [primaryHost_desktop] : is for the host name / ip address of desktop
 service client
# [serviceIP_desktop]   : is for the ip address of server which the
client will connect to
# [servicePort_desktop] : is for the port number of server which the
client will connect to
#
primaryHost_admin=localhost
serviceIP_admin=localhost
servicePort_admin=6001

primaryHost_desktop=localhost
serviceIP_desktop=localhost
servicePort_desktop=6001
```

Installing Vault

Follow the steps below to install Vault:

1. Open a command prompt using the **Run as administrator** option.
2. Go to your install directory into which Vault system files were extracted, (for example, c:\vault-temp).
3. If you wish to change the default values, open `install-vault-config.txt` in Notepad and change values as required. Care should be taken when editing `install-vault-config.txt`.

Here are the settings:

[Environment]	Environment ID, should match the value inside licence file. If you don't know the Environment ID from a license file, use the `print-lic.bat` scripts to display the license information. For example: print-lic.bat c:\vaultlic\VaultKeys.xml
[ip_server]	IP address of e2serverd
[ip_loader]	IP address of e2loaderd
[ip_render]	IP address of e2renderd
[ip_router]	IP address of e2routerd
[port_server]	Port number of e2serverd
[port_loader]	Port number of e2loaderd
[port_render]	Port number of e2renderd
[port_router]	Port number of e2routerd
[primaryHost_admin]	Host name/IP address of admin service client
[serviceIP_admin]	Host name/IP address of server which the client will connect to
[servicePort_admin]	Port number of server to which the client connects
[primaryHost_desktop]	Host name/IP address of desktop service client
[serviceIP_desktop]	IP address of server to which the client connects
[servicePort_desktop]	Port number of server to which the client connects

4. Run `install-vault.bat` from your install directory (for example, `c:\vault-temp\install-vault.bat`).
5. In case an error condition is encountered, the appropriate error message will be displayed at the command prompt or from a log file (`install-vault-YYYYMMDD-HHMMSS.log`). Follow the information provided in the error messages to resolve the issue and retry.
6. On successful installation the following message is displayed at the command prompt: Vault system installation complete.

7. Restart your system, Vault is ready to use.

Installing the Admin Service client (optional)

To install the Windows Admin Service client follow these steps:

1. Open a command prompt using the **Run as administrator** option.
2. Go to your install directory into which Vault system files were extracted, (for example, c:\vault-temp).
3. If you wish to change the default values, open install-admin.bat in Notepad and change values highlighted below:

   `set serviceIP=localhost` - server's iP address the client will connect to

   `set primaryHost=localhost` - server's IP address the client will connect to

   `set servicePort=6001` - server's port number the client will connect to

   `set targetdir=C:\Program Files (x86)\PBBI CCM\Vault` - the target install root path

   Care should be taken when editing `install-admin.bat`.
4. Run `install-admin.bat` (for example, `c:\vault-temp\install-admin.bat`)
5. In case an error condition is encountered, the appropriate error message will be displayed at the command prompt or from a log file (`install_vault_admin.log`). Follow the information provided in the error messages to resolve the issue and re-try.
6. On successful installation the following message is displayed at the command prompt: Vault admin client installation complete.
7. Restart your system, the Windows Admin Service client is ready to use.

Installing Desktop Service client (optional)

1. Open a command prompt using the **Run as administrator** option.
2. Go to your install directory into which Vault system files were extracted, (for example, c:\vault-temp).
3. If you wish to change the default values, open `install-desktop.bat` in Notepad and change values highlighted below:

   `set serviceIP=localhost` - server's iP address the client will connect to

   `set primaryHost=localhost` - server's IP address the client will connect to

   `set servicePort=6001` - server's port number the client will connect to

   `set targetdir=C:\Program Files (x86)\PBBI CCM\Vault` - the target install root path

Care should be taken when editing `install-admin.bat`.

4. Run `install-desktop.bat` (for example, `c:\vault-temp\install-desktop.bat`)
5. In case an error condition is encountered, the appropriate error message will be displayed at the command prompt or from a log file (`install_vault_desktop.log`). Follow the information provided in the error messages to resolve the issue and re-try.
6. On successful installation the following message is displayed at the command prompt: Vault desktop client installation complete.
7. Restart your system, the Desktop Service client is ready to use.

Optimizing file system performance

Microsoft Windows deals with the recording of short file names for compatibility with much older software that is turned on by default on the operating system. This functionality can be changed by running a single DOS command on the server where Vault is installed.

Note that you will notice a performance improvement only with a large number of files (over 300,000) in relatively few folders where a lot of the filenames start with similar names. Not having 8.3 filenames available will prevent the use of old applications such as Word 2.0 and Excel 4.0.

Since Vault typically has filenames beginning with CCYYMMDD-HHMMSS, many files often do have the same first 'n' characters (specifically, the year: 2006, and even the following characters only use numbers, reducing the variability available in the first 8 characters. This causes the operating system to perform many searches to find the next available shortened filename to use on a given file.

To disable this functionality, run the DOS command prompt and type the following:

```
C:\> FSUTIL behavior set disable8dot3 1
```

Installing Vault in small memory environments

If Vault is being installed in a small memory Windows environment (less than 4 GB), the memory footprint of the Indexer as a Service can be minimized by tuning the cache size in the initialization parameters.

This is done by creating a file named `indexerd.ini` in the Vault "server" directory. The contents of the file should include the following lines:

```
[Indexer1]
CacheSize=256
QueueSize=10000
```

Upgrading an existing e2 Vault installation on Microsoft Windows

To upgrade an existing e2 Vault installation on Microsoft Windows:

1. Back up the existing installation and data before doing the upgrade.
2. Stop Vault services or Vault running applications.
3. If the version of Vault that will be installed is same as the existing one, copy the executable files.If the versions are different, remove the existing installation and then follow the above installation sections.

   Note: if you are using vault indexer as a service release 6.0, the `index.dr2` file created under release 6.0 will not be compatible with the release 6.1 or higher of the indexer as a service.

   A full reindex of your data using release 6.1 or higher will be required to make your document index visible with release 6.1.

   The recommended method is to rename your existing release 6.0 `index.dr2` file and allow vault to create a new empty index.dr2 file. the reindex operation will then populate the newly created `index.dr2` file.

UNIX installation

Linux

The following procedure uses the GNU "tar" command to install Vault software on Linux.

Note: Before installing under Linux, verify that “SELinux” support is not set to the “Enforcing” mode. Vault may not install or run correctly if SELinux is set to this mode.

1. Log on as root.
2. Copy the Vault installation tarball to your directory. The name will usually be constructed of the product name (Vault), the release number (7.2.0.20 for example), and the operation system version (Linux) and ends with ".tar.gz".
3. Decompress the installation tarball:

   ```
   tar -zxvf <name-of-the-install-tarball>.tar.gz
   ```

4. Run the install script:

```
cd <name-of-the-tarball>
./install.sh
```

Note: If you are installing on a 64 bit Linux system, please verify that the following prerequisite libraries are installed on your system:

- `glibc.i686`
- `libstdc++.so.6`

If these libraries are missing, the Vault executable will fail to run.

General information

1. Vault software will be installed to `/opt/PBBI CCM/Vault` which is the installation root directory and contains the following sub-directories:
   - `lib`
   - `render`
   - `router`
   - `server`
2. The Vault repository server can be configured through the INI configuration files in the repository server's root directory:
   `/opt/PBBI CCM/Vault/server`
   Vault ADM server's download directory is:
   `/opt/PBBI CCM/Vault/server/download`
3. Please note that on UNIX platforms, Vault ADM server only monitors the download directory for files that have the ".done" extension (for example, `MyFile.afp.DONE`, `MyFile.jrn.DONE`, `MyFile.rpk.DONE`). Files with all other extensions are ignored and will not be processed by Vault server.
   The recommended way to transfer printstream files (and their associated journals and resource packs) is to first transfer the files to the ADM server's download directory without the ".DONE" extension. Once the files have been transferred/copied successfully, they should then be renamed and have the ".DONE" extension appended:

```
(example)
    root:[~/]# for i in MyFile.afp MyFile.jrn MyFile.rpk;
    > do
    > cp $i /opt/PBBI CCM/Vault/server/download/$i && mv /opt/PBBI
CCM/Vault/server/download/$i /opt/PBBI
CCM/Vault/server/download/$i.DONE
    > done
```

Post-install configuration

Licence file and Environment information has to be added or updated before Vault servers can be run, as the installation package does not supply a Licence file.

After successfully completing the installation, copy the Licence file (for example, `VaultKeys.xml`) to Vault repository server's root directory (`/opt/PBBI CCM/Vault/server`) and update Vault repository server's `server.ini` configuration file:

```
[Licence]
LicenceFile=<Path to keycode/licence file>
Environment=<environment from keycode/licence to select>
```

The LicenceFile key can be set to either an absolute path and filename:

```
(for example)
LicenceFile=/opt/PBBI CCM/Vault/server/VaultKeys.xml
```

or a path/filename relative to the repository server's root directory:

```
(for example)
LicenceFile=VaultKeys.xml
```

Starting and stopping Vault servers

Vault servers can be started or stopped using vault RC (run control) script that is installed by the installation script:

`/etc/init.d/vault`

Once Vault software has been installed and configured (Licence file, profiles, etc.) Vault servers can be controlled as follows:

(to start Vault servers)

`/etc/init.d/vault start`

(to stop currently running Vault servers)

`/etc/init.d/vault stop`

> **Note:** this script only facilitates manual startup and shutdown of Vault servers.

If you are not using the Indexer as a Service, you can use `/etc/init.d/vault_noindexer` script in place of the `/etc/init.d/vault` script. The `vault_noindexer` script will not start or stop the Indexer as a Service.

Uninstalling Vault

1. Ensure that all Vault servers have been stopped: `/etc/init.d/vault stop`

2. Manually uninstall Vault software by removing:
   - the installation root directory (including all binaries, libraries, configuration files, and ingested data): `rm -rf /opt/PBBI CCM/Vault`
   - Vault RC scripts:

```
rm -f /etc/init.d/vault
rm -f /etc/init.d/vault_noindexer
```

AIX

The following procedure uses the AIX “gunzip” and “tar” commands to install Vault software on AIX:

1. Log on as root.
2. Copy the Vault installation tarball to your directory. The name will usually be constructed of the product name (Vault), the release number (7.2.0.20 for example), and the operation system version (Linux) and ends with ".tar.gz".
3. Decompress the installation tarball:

   ```
   /bin/gunzip -dc <name-of-the-install-tarball>.tar.gz | /bin/tar -xf -
   ```

4. Run the install script:

```
cd <name-of-the-tarball>
./install.sh
```

Note: The install script will prompt for confirmation that the dependencies (C++ runtime libraries for AIX) have already been installed before continuing with the installation.

General information

1. Vault software will be installed to `/opt/PBBI CCM/Vault` which is the installation root directory and contains the following sub-directories:
   - `lib`
   - `render`
   - `router`
   - `server`
2. The Vault repository server can be configured through the INI configuration files in the repository server's root directory: `/opt/PBBI CCM/Vault/server/download`

   Vault ADM server's download directory is: `/opt/PBBI CCM/Vault/server/download/opt/`

3. Note that on UNIX platforms, the Vault ADM server only monitors the download directory for files that have the ".DONE" extension (for example, `MyFile.afp.DONE`, `MyFile.jrn.DONE`, `MyFile.rpk.DONE`).

   Files with all other extensions are ignored and will not be processed by the Vault server.

   The recommended way to transfer print files (and their associated journals and resource packs) is to first transfer the files to the ADM server's download directory without the ".DONE" extension. Once the files have been transferred/copied successfully, they should then be renamed and have the ".DONE" extension appended:

   Example

```
root:[~/]# for i in MyFile.afp MyFile.jrn MyFile.rpk;
    > do
    > cp $i /opt/PBBI CCM/Vault/server/download/$i && mv    /opt/PBBI
 CCM/Vault/server/download/$i /opt/PBBI
CCM/Vault/server/download/$i.DONE
    > done
```

Post-install configuration

Licence file and Environment information has to be added and/or updated before Vault servers can be run, as the installation package does not supply a Licence file.

After successfully completing the installation, copy the Licence file (for example, `VaultKeys.xml`) to the Vault repository server's root directory (`/opt/PPBI CCM/Vault/server`) and update Vault repository server's `server.ini` configuration file:

```
[Licence]
LicenceFile=<Path to keycode/licence file>
Environment=<environment from keycode/licence to select>
```

The LicenceFile key can be set to either an absolute path and filename:

```
LicenceFile=/opt/PBBI CCM/Vault/server/VaultKeys.xml
```

or a path/filename relative to the repository server's root directory:

```
LicenceFile=VaultKeys.xml
```

Starting and stopping Vault servers

Vault servers can be started or stopped using the Vault RC (run control) script that is installed by the package:

```
/etc/rc.d/vault
```

Once the Vault package has been installed and configured (Licence file, profiles, etc.) Vault servers can be controlled as follows:

To start Vault servers:

```
/etc/rc.d/vault start
```

To stop currently running Vault servers:

```
/etc/rc.d/vault stop
```

Note:

- This script only facilitates manual startup and shutdown of Vault servers.
- If you are not using the indexer as a Service, then you can use the `/etc/rc.d/vault_noindexer` script instead of the `/etc/rc.d/vault` script. The `vault_noindexer` script will not start or stop the indexer as a service

Uninstalling Vault

1. Ensure that all Vault servers have been stopped: `/etc/rc.d/vault stop`
2. Manually uninstall Vault software by removing:
   - The installation root directory (including all binaries, libraries, configuration files, and ingested data): `rm -rf "/opt/PBBI CCM/Vault"`
   - Vault RC scripts:

```
rm -f /etc/rc.d/vault
rm -f /etc/rc.d/vault_noindexer
```

Solaris on SPARC

The following procedure uses the Solaris “gunzip” and “tar” commands to install Vault software on Solaris:

1. Log on as root.
2. Copy the Vault installation tarball to your directory. The name will usually be constructed of the product name (Vault), the release number (7.2.0.20 for example), and the operation system version (Linux) and ends with ".tar.gz".
3. Decompress the installation tarball:

   ```
   /bin/gunzip -dc <name-of-the-install-tarball>.tar.gz | /bin/tar -xf -
   ```

4. Run the install script:

```
cd <name-of-the-tarball>
./install.sh
```

General information

1. Vault software will be installed to `/opt/PBBI CCM/Vault` which is the installation root directory and contains the following sub-directories:
   - `lib`
   - `render`
   - `router`
   - `server`
2. The Vault repository server can be configured through the INI configuration files in the repository server's root directory: `/opt/PBBI CCM/Vault/server`

   Vault ADM server's download directory is: `/opt/PBBI CCM/Vault/server/download`
3. Note that on UNIX platforms, the Vault ADM server only monitors the download directory for files that have the ".DONE" extension (for example, `MyFile.afp.DONE`, `MyFile.jrn.DONE`, `MyFile.rpk.DONE`).

   Files with all other extensions are ignored and will not be processed by Vault server.

   The recommended way to transfer print files (and their associated journals and resource packs) is to first transfer the files to the ADM server's download directory without the ".DONE" extension. Once the files have been transferred/copied successfully, they should then be renamed and have the ".DONE" extension appended:

   Example

```
root:[~/]# for i in MyFile.afp MyFile.jrn MyFile.rpk;
    > do
    > cp $i "/opt/PBBI CCM/Vault/server/download/$i" && mv "/opt/PBBI
 CCM/Vault/server/download/$i" "/opt/PBBI
CCM/Vault/server/download/$i.DONE"
    > done
```

Post-install configuration

Licence file and Environment information has to be added and/or updated before Vault servers can be run, as the installation package does not supply a Licence file.

After successfully completing the installation, copy the Licence file (for example, `VaultKeys.xml`) to the Vault repository server's root directory (`/opt/PBBI CCM/Vault/server`) and update Vault repository server's `server.ini` configuration file:

```
[Licence]
LicenceFile=<Path to keycode/licence file>
Environment=<environment from keycode/licence to select>
```

The LicenceFile key can be set to either an absolute path/filename:

```
LicenceFile=/opt/PBBI CCM/Vault/server/VaultKeys.xml
```

or a path/filename relative to the repository server's root directory:

```
LicenceFile=VaultKeys.xml
```

Starting and stopping Vault servers

Vault servers can be started or stopped using the Vault RC (run control) script that is installed by the package:

```
/etc/init.d/vault
```

Once the Vault package has been installed and configured (Licence file, profiles, etc.) Vault servers can be controlled as follows:

To start Vault servers:

```
/etc/init.d/vault start
```

To stop currently running Vault servers:

```
/etc/init.d/vault stop
```

Note:

- This script only facilitates manual startup and shutdown of Vault servers.
- If you are not using the Indexer as a Service, you can use the `/etc/init.d/vault_noindexer` script to start and stop Vault instead of the `/etc/init.d/vault` script. The `vault_noindexer` script will not start or stop the Indexer as a Service.

Uninstalling Vault

1. Ensure that all Vault servers have been stopped: `/etc/init.d/vault stop`
2. Manually uninstall Vault software by removing:
   - The installation root directory (including all binaries, libraries, configuration files, and ingested data): `rm -rf "/opt/PBBI CCM/Vault"`
   - Vault RC scripts:

```
rm -f /etc/init.d/vault
rm -f /etc/init.d/vault_noindexer
```

Failure recovery on UNIX

In the event that Vault experiences a failure (crash, abnormal exit due to mis-configuration, etc.) on UNIX, the following procedure can be used to recover and restart Vault servers after a such an occurrence:

1. Ensure that all Vault servers have been stopped either by running the Vault RC (run control) script for your platform or by manually stopping the servers (for example, "`ps -ef|grep -i e2`", "`kill -TERM <Vault server PIDs>`", etc.).

   If you are running the Indexer as a Service, you must manually stop it as well (for example, "`ps -ef | grep -i indexerd`", `kill -TERM <Vault indexerd PID>`"
2. Remove any orphaned files left behind for example, core file (`/opt/PBBI CCM/Vault/server/core`) or PID files (`/var/opt/vault/run/*.pid`), etc.
3. If the failure condition was due to a configuration error (for example, "the compressed block size is too small to handle this data"), update Vault server's relevant configuration to correct the error.
4. Restart Vault servers.

Upgrading an existing e2 Vault installation on UNIX

1. Unpack the install package (tar -zxvf …)
2. Stop the existing vault system
   a) `su root` (enter root password when prompted)
   b) `cd /etc/init.d`
   c) `./e2vault stop`
3. Backup the following files in the existing Vault install path (`/opt/PBBI CCM/Vault`)
   a) `server/e2serverd`
   b) `server/e2util`
   c) `server/e2loaderd`
   d) `render/e2renderd`
   e) `router/e2routerd`
   f) `lib`
4. Backup all files in `server/tools`
5. As root, copy the equivalent new files from the installation directories
   a) `server/e2serverd`
   b) `server/e2util`
   c) `server/e2loaderd`
   d) `render/e2renderd`
   e) `router/e2routerd`
   f) All files from `server/tools`
   g) All files from `lib`
6. Restart Vault.
   a) `su root` (enter root password when prompted)

b) `cd /etc/init.d`

c) `./e2vault start`

Note: This upgrade process does not install the Indexer as a Service.

3 - Configuring and running Vault

The main component of Vault is the document repository which compresses, stores and manages documents composed in a wide range of output datastreams including most of those produced within the Generate environment.

In this section

Overview

Under normal circumstances the processes required by the repository are automatically invoked when the operating system is started. Before you can start loading documents into the repository you will need to define the properties of the incoming documents. Refer to **Creating application profiles** on page 43 for further information. This section is intended to provide information on configuring, loading and troubleshooting repository processing.

Another component which makes up Vault is the Mobile Vault which is a Microsoft Windows based application that allows documents to be read from a local copy of an Vault database. Refer to the "Vault User Guide" for more information.

Before you configure

The output datastream passed to Vault must be in one of the supported formats such as AFPDS, Metacode, or Postscript. Before any datastream is usable within Vault you must provide or create an index for the documents it contains.

The storage in vault is broken into several areas:

Page data:

- compressed and stored in .drp files
- shared across whole the server

Document data:

- includes properties such as `doc.date`
- includes pointers to pages in the .drp files
- is compressed and stored in .drd files
- shared across whole the server.

Index data:

- each database has its own indexes.
- made up of the customer table (.drr/.drt) and index files (.dri/.dru).
- allows controls that let you specify which documents are visible in a database.
- if using authentication, databases can be restricted to specific users or groups.index entries are made up of:
  - fields from the document, such as `doc.data`.
  - pointers to the document records in the .drd files or customer records in the customer table.

Note: Loading resources via HIP file is not supported for Postscript.

- Datastreams produced by Generate are the preferred source as the environment provides a simple method of creating an acceptable index known as an Interchange Journal (DIJ).
- For datastreams not created by Generate you will need to configure Vault to identify the appropriate index information directly from the contents of the datastream.

The settings related to a particular application are stored as sections within the profiles.ini file. Each section is known as an application profile and contains:

- the expected type of output datastream and index.
- the name template by which files related to the application are identified by the ADM.
- document handling parameters.

If required, you can move or rename the file location. In some scenarios you can configure Vault to store documents from particular applications in different databases so that you can refine access control.

Security

Enabling SSL for Vault servers

SSL can be enabled for Vault servers by modifying their respective configuration files. Assuming that you already have the SSL private key (for example, `e2vault-server.key`) and the corresponding SSL certificate (for example, `e2vault-server.crt`), enable SSL for Vault servers as follows:

Note: The insertion of chevron markers (`>>>>`) denote the newly added lines for SSL.

1. Vault server: modify the `server\e2serverd.ini` file.

```
[server1]
 service=*:6001
>>>> ssl=1
>>>> sslcertificate=e2vault-server.crt
>>>> sslprivatekey=e2vault-server.key
 [connection1]
>>>> ssl=1
```

2. Loader server: modify the `server\e2loaderd.ini` file.

```
 [server1]
 service=*:6002
 [connection1]
 service=localhost:6001
>>>> ssl=1
>>>> sslcertificate=e2vault-server.crt
>>>> sslprivatekey=e2vault-server.key
```

3. Rendering engine: modify the `render\e2renderd.ini` file.

```
[server1]
 service=*:6003
>>>> ssl=1
>>>> sslcertificate=e2vault-server.crt
>>>> sslprivatekey=e2vault-server.key
 [connection1]
 service=localhost:6001
>>>> ssl=1
```

4. Vault Router server: modify the `router\e2routerd.ini` file.

```
 [router1]
 # Number of rendering engines to use
 count=2
 [server1]
 # Hostname and port that e2routerd listens on for incoming connections

 service=*:7003
>>>>  ssl=1
>>>>  sslcertificate=/opt/e2vault-server.crt
>>>>   sslprivatekey=/opt/e2vault-server.key
 # First of two rendering engines to use
 [connection1]
 service=127.0.0.1:6003
>>>>  ssl=1
 # Second of two rendering engines to use
 [connection2]
 service=127.0.0.1:6004
>>>>  ssl=1
```

With the above changes made, once Vault servers are restarted, SSL will be enabled and used for all network communications between the servers and for communications to/from the server by other systems or entities (Perl web client, Java Service web client, and etc., along with the API sets, such as .NET API and Java API).

Using `uclient.exe` and `loader.exe`

When enabling SSL, if you wish to use the 'uclient.exe' and 'loader.exe' you need to add 2 lines to client.ini.

```
 [installer]
>>>> ssl=1
 primary=e2vault2
 [connection1]
>>>> ssl=1
 service=e2vault2:6001
 serverlicence=1
```

Generating an SSL certificate for use with Vault

1. Use the openssl executable/binary located in the server\tools folder of your Vault install for generating the SSL key and certificate for Vault.
2. Set the OPENSSL_CONF environment variable to the full path and location of the openssl.cnf configuration file which is also located in the server\tools folder of your Vault install:

   UNIX example

   ```
   export OPENSSL_CONF=/opt/PBBI CCM/Vault/server/tools/openssl.cnf
   ```

   Microsoft Windows example

   ```
   set OPENSSL_CONF=C:\Program Files\PBBI
   CCM\Vault\server\tools\openssl.cnf
   ```

3. Generate a new SSL key-certificate pair to use with Vault as follows:
   a) Change directory into the folder containing the openssl executable/binary (see above).
   b) Generate an RSA Private Key (example below creates a 4096-bit key):

      ```
      openssl genrsa -out e2vault-server.key 4096
      ```

   c) Generate a CSR (Certificate Signing Request)

      ```
      openssl req -new -key e2vault-server.key -out e2vault-server.csr
      ```

   d) Generate a Self-Signed SSL Certificate

      ```
      openssl x509 -req -days 365 -in e2vault-server.csr -signed
      e2vault-server.key -out e2vault-server.crt
      ```

      You can change the validity of the generated SSL certificate as desired; the above example makes the certificate valid for 1 year.

      Once the steps above have been completed successfully, you will have the following files:

      - e2vault-server.crt (Self-signed SSL certificate for Vault server)
      - e2vault-server.csr (Certificate Signing Request that was used to create the self-signed certificate above)
      - e2vault-server.key (RSA Private Key that was used to self-sign the SSL certificate above)

4. Copy the `e2vault-server.crt` and `e2vault-server.key` files into the directory containing the e2serverd and e2loaderd executables (for example, `\some\path\server\`), as well as the directory containing the e2renderd executable (for example, `\some\path\render\`).

Configuration steps

Carry out the following steps when configuring your installation of the document Vault:

1. Configure document storage settings, refer to **Storing documents** on page 36
2. Configure directory paths, refer to **Configure directory paths** on page 38
3. Provide index information, refer to **Provide index information** on page 40
4. Create application profiles, refer to **Creating application profiles** on page 43
5. Provide font and image resources, refer to **Provide font and image resources** on page 45
6. Start the necessary Vault processes, refer to **Start Vault processes** on page 47

Compression

It is important to note that the maximum size of an individual output datastream file that can be stored in Vault after compression should not exceed 4GB. If the file is larger than 4GB, consider splitting the incoming datastream into multiple files.

Storing documents

Output datastreams and their associated files are loaded into the repository by the Automated Data Manager feature (ADM) which polls the download directory for incoming data.

The download directory is defined in the `server.ini` file as follows:

```
[Paths]
DownloadPath=path
```

If the files are being generated on a machine remote from Vault you will need to use a transfer protocol such as FTP and file sharing. You must ensure that all text translation features are disabled when transferring data. Note: caution is needed with file transfer in that failed transfers directly to download might lead to fragments being ingested.

Note: Datastream and HIP files should be handled as binary.

Any information, warning or error messages generated by the ADM are stored in a log file which has the default location of:

```
<drpath>\server\log\process.<timestamp>.<processid>.log
For example: e2loaderd.20120130.080827.4148.log
```

Naming download files

Files that appear in the ADM download directory should have names that conform to one of the entries in the [`FileMap`] section of `profiles.ini`. In this way they are mapped to an application profile within the ini that specifies the type of data and index expected and any other custom document or application requirements. Refer to **Creating application profiles** on page 43 for details.

Note: It is important that the download file names be unique enough so that they are not confused for another.

It is important to note that for Generate created datastreams the base name used for both the datastream itself and the associated DIJ must be the same.

The requirements are:

- The base job name is the same for the stream and journal (if using journals).
- The base job filenames must be unique so that streams and journals for multiple jobs are not confused.
- The search fragments in the [filemap] are tested in order and the first fragment found determines the profile.

Output datastreams will be identified providing their base name matches an application profile. Data files must end with a recognized extension, such as:

.afp	AFPDS
.mtc	Metacode
.ps	PostScript
.xml	HTML pages contained within an XML 'pak' construct as produced by Generate

It is also recommended that file have the following format:

```
YYYYMMDD-HHMMSS-doctype
```

Where the first part of the file name is the date and time in a most-to-least significant order – year month day, hour minute second. This is of particular benefit when viewing a listing of your files which will always be shown in date order when sorted by name. 'doctype' indicates the application – use a short name such as BILL14 as this is repeated for all documents of this type that share the same profile – resources, format, page options, etc. Ensure that the names are as unique as possible to prevent failed files from having problems in the work directory.

For example:

```
20020131-114532-bizbill.afp
```

Downloading remote files

A remote file can be downloaded without having to copy or move the file to the download directory. ADM is capable of retrieving a remote file over the network, provided it is both shared and accessible to the account ADM is using. Under Microsoft Windows, files accessed this way should be specified using UNC syntax: `\\servername\sharename\path\filename`

To download a remote file, create a file with the .indirect extension and place it in the download directory. Indirect files may stand in for print streams, journals or resource packs. Enter the full path to a file in another location (`<drpath>`). Vault will treat the .indirect file as if it was the remote file when placed in the download directory. The remote file does not get deleted after processing is complete (only the .indirect file is deleted). You also do not have to copy or move the file to the download directory, Vault will read it from where it is. Indirect files may stand in for print streams, journals or resource packs.

> **Note:** e2serverd.exe must be able to access the file given the account the service logs in as. this also applies when you redirect a directory to another location or set access controls on the existing Vault tree.

Configure directory paths

Vault consists of several working directories. These include the repository itself, a log directory, diagnostic tools, various working directories, and the download directory. By default these are created as sub-directories under the main installation directory name.

Defining the download directory

The download directory is a key part of the mechanism that loads document datastreams and associated resources. It is continually polled by Vault Loader for new files which are then processed and stored in the repository as appropriate.

By default the download directory will be located in `<drpath>\server\download` you can reconfigure the download directory to any location that is directly accessible by the repository. You may, for example, want to isolate it from the rest of the repository directories as a security measure. You can then allow general access to the download directory while leaving the other directories in a more secure environment. You may also want to redirect subdirectories to improve disk performance. For example, you can place the index directory, download directory, pagedata directory, and etc., on separate drive arrays to increase the number of simultaneous operations by sending requests to different hardware sub-systems.

Under normal circumstances the other directories that make up the repository are not user configurable. You should, however, ensure that these have a suitable level of protection against unauthorized access.

To move the download directory:

1. Create or identify the required directory.

   If you are not using the default location for the download directory you will need to create a new directory within your file system. This is *not* created by the repository.

2. Open the server initialization file (`server.ini`) for edits.

   `<drpath>\server\server.ini`

3. Insert or change the DownloadPath setting.

   The required syntax is:

   ```
   [Paths]
   DownloadPath=dlpath
   ```

   Where `dlpath` is the is the path name that will be polled for new documents. You will need to create the [Paths] section if it does not already exists in the ini file. Refer to the "Vault Customizing Guide" for detailed information on `server.ini` settings.

4. Stop and restart the e2loadered service.

Provide index information

Documents are stored in Vault in a structure that makes it easy to select and browse the required documents. This is done by organizing documents in a similar way to the traditional paper filing system – sorting documents together by a unique identifier into logical folders. Once a single customer has been selected, all the documents that have been archived for that customer can be quickly and easily browsed.

Typically the unique identifier, or primary key, would be the customer account number as this is normally a unique, non-recycled number that clearly and positively identifies a specific customer.

> **Note:** Important: if your primary key is not unique or is recycled, contact your Vault supplier to resolve this issue before proceeding.

Other information, or secondary keys can be linked with the primary key, such as the customer name, address, phone number, social security number. These keys provide the index and can be used to search for a document, for example, the account number for 'Joe Smith' or the individual at '123 Water Street'. Any information can be a key, whatever helps in searching for a document.

Providing the index information

If you are creating the documents to be archived using Generate then the index information is provided in a separate file that is loaded into Vault along with the output datastream. This file is known as a standard or XML journal (Document Interchange Journal).

You specify the type of index being used for an application as part of the profiles initialization file. See **Creating application profiles** on page 43 for details.

The DIJ index process

A DIJ is an XML construct that can contain all the references with which it is possible to search for documents within vault. The DIJ is defined as a standard Journal when creating the application using the Designer. All the indexible references can be provided using references to fields within the application data to be used in the production environment or by other objects such as constants and environment settings. One entry per document is added to the DIJ when it is created by the GEN component of Generate.

Important: Special care is required if you intend to manipulate the order of pages within the datastream after it has been rendered by Generate. You may be using StreamWeaver or a third party tool to merge or reorder pages within datastreams. Where this is the case you must ensure

that a new DIJ file is created that reflects the amended datastream. StreamWeaver provides commands that allow you to read and write DIJ records and individual elements as required. If this is not feasible, consider storing the documents in Vault before any such post processing. Another consideration is the page signatures that generate marks pages with so that you can double check the journal to stream page mapping in the XML journal based modes.

> **Note:** Refer to the designer users guide for creating a DIJ object and the production guide for specifying the file to receive the DIJ output.

Once generated, the DIJ should be passed to Vault along with the datastream to which it pertains by placing both files in the ADM download directory. The base name of both files must match the relevant entries in the [`FileMap`] section of `profiles.ini` for this to happen. Ensure the following conditions are met:

- Stream and journal must share the same base name (`20120101-invoice.afp`, `20120101-invoice.jrn`)
- There are appropriate file map entries to map the file name to the correct profile.

ADM automatically creates Vault index entries from the information held in the DIJ. By default the index is assumed to contain four basic references (keys) which are displayed to the end user when searching for a document.

These are:

- Customer account number (primary key)
- Customer name
- Customer address
- Document date

The following are the default indexes:

account	to customer	required	visible
name	to customer	optional	visible
address	to customer	optional	visible
invlink	to document	required	visible
guid	to document	optional	hidden
iguid	to document	optional	hidden

Note that the various individual address elements that are configured as part of the DIJ object in the Designer (address line 1-7 and Postal Code) are concatenated together to form the single Customer address index element by ADM.

If required, you may amend the attributes that are made available to client systems by using the Index and Render keywords:

- `indexN=` entries in `profiles.ini` can change how keys are created.
- `indexN=` entries in `database.ini` can alter search behavior.
- `RenderN=` entries in `database.ini` are used to specific search output columns.

Refer to the "Vault Customizing Guide" for detailed information on alternative methods of indexing.

Working with non-Generate created data

If you are using non-Generate created date data you will need to configure Vault to identify the appropriate index information directly from the contents of the datastream as it is loaded. Using this method may limit the flexibility of your application as redesigning a document may result in the content of the output datastream no longer being compatible with the index generation criteria specified originally. You will also need to understand the relevant datastream protocol so that you can specify the elements to be searched.

> **Note:** Non-journal methods may be useful for loading legacy data that was generated before the production of journals was possible in your environments.

The supported non-indexing methods are as follows:

Journal: uses a text file that contains the index information that has been extracted from the datastream in a prescribed format. Note that the journal model doesn't scrape data from the stream as the genericXXX modes do and so it is not sensitive to layout or print encoding changes.

GenericAFP: uses TRN Transparent Data commands (TRN is a transparent data command) within AFPDS files to indicate the start of a command sequence which completes with the text string required for indexing.

GenericTLE: values within AFP Tag Logical Element (TLE) records provide the index information.

GenericMetacode: searches for binary patterns within Metacode streams to determine the start and end of index text. Refer to the Vault Customizing Guide for further information on alternative methods of indexing.

StreamWeaver: added <tags> and edit assistants in the Visual Engineer product to assist in making the XML journal files.

Output from Generate

When working with output from Generate you must always supply the following resources to the download directory as a group:

- The output datastream file (containing the actual documents)
- The DIJ file that provides the index into the documents.
- The fonts and image resources referenced within the datastream.

Font and image resources are stored in the HIP file as created by a Designer publishing task. Fonts and image resources can be:

- manually loaded.
- loaded via resource packs (which can be dropped into download).
- via `ExtractResources=` (for inline AFP resources).
- resource packs should (ideally) be loaded before print streams that use them (otherwise it will delay the loading of those streams).

They are made available to the download directory where they are matched to the appropriate datastreams using internal identifiers stored in the associated DIJ. A HIP containing new or updated font and image resources can be passed to the ADM polling location as soon as it becomes available, but for consistency you may want to make it available at the same time as the application output files to which they relate.

Creating application profiles

You may intend to store documents related to one or many applications within Vault. The settings that govern how each application is handled within the Vault environment are defined in the profiles.ini file, a default version of which is installed with Vault software.

`profiles.ini` is made up of multiple sections. There must always be a [FileMap] section plus as many 'application profiles' sections as needed by your installation.

The [`FileMap`] section is used to identify which application profile is to be applied to incoming file. For example, how the ADM will process files found in the download directory. For each application profile you should specify one or more file name templates that identify the files belonging to that application. In the following example all files that include the string '' in any part of their basename will be associated with the application profile `MyApp` as shown below:

```
[FileMap]
=MyApp

[MyApp]
Format=AFP
Documents=XMLJournal
```

In this example the following files would be associated with the named profile `MyApp` profile: `new.afp`, `.afp`, `new.jrn`, `newapp.afp` and so on. Note that templates apply to both datastreams and their associated DIJ, standard and XML journal index files where applicable.

You may specify the application keyword (`MyApp` in this example) multiple times to associate additional file name templates if necessary.

The primary function of the application profiles themselves is to identify the type of output datastream expected (the Format keyword) and the method by which documents are to be indexed (Documents).

You can also use them to define a specific document database and resource file location for the application and include other settings that provide formatting information not included within the datastream itself. Refer to the Vault Customizing Guide for a full listing of `profiles.ini` settings.

Extended example of profiles.ini

```
[FileMap]
Stat=Statements
=Default

[Statements]
Documents=XMLjournal
Format=Postscript
Stream=1
LengthDelimited=0
TapeBlockFormat=1
MarginX=0
MarginY=0
Tray=group1.wmf
PageBreak=0
SkipHeaderPages=1

[Default]
Documents=GenericTLE
Format=AFP
Database=UnknownDocs
```

Databases and access rights

A database is a collection of documents within Vault. Documents are included in a particular database by reference names that are associated with them when they are loaded into Vault. Documents are added to the default database unless otherwise configured.

You can apply one or more database references to documents either at the application level using `profiles.ini` or globally using `server.ini`. You can also specify multiple databases or * to read a list of databases from the document record attributes.

To configure an application specific database:

Add the database keyword to an application profile section of `profiles.ini` . The name associated with the keyword becomes a path name within the Vault environment so you should restrict its length to 16 characters and avoid using special characters. Do not use the names 'Default' or 'Error' which are reserved.

For example:

```
[MyApp]
Database=SpecialReports
```

For further information about the profiles initialization file refer to **Creating application profiles** on page 43. Refer to the Vault Customizing Guide for a full listing of `profiles.ini` settings.

If database references are in place and you are using the Rendering Engine to create a custom client for Vault you can code appropriate access control methods using these settings.

To control access to databases, for Vault client applications, you can set up Microsoft Windows authentication for the databases in `server.ini`. For more information, refer to the Server initialization file section in the "Vault Customizing Guide".

Provide font and image resources

If you are creating the documents using Generate the resources can be made available by passing the HIP file to the ADM to be loaded into Vault.

Resources can be stored either in a default location or in a specific directory, known as a resource set. Note that there is a default resource set called “default”.

For Generate created documents additional sub-directories are created for the resource set and are not user configurable.

To change the default resource location:

Set the following keyword in `server.ini`

```
[Production]
ResourceSet=<resdir>
```

Where:

`ResourceSet=`	The meaning is different for XML journals, in this case it is the name of the template resource set (which gets copied into the one the xml journal process automatically creates).
`<resdir>`	A directory in `<drpath>\server\distrib\` for the resource set. The default is Default.

It is important to note that this keyword should not normally be changed once you start loading resources to Vault.

To configure an application resource set:

Set the following keyword in `profiles.ini`

```
[<Profile>]
ResourceSet=<resdir>
```

Where:

`<Profile>`	the application profile being amended.
`<resdir>`	directory in `<drpath>\server\distrib\` for the resource set.
`<drpath>`	directory where the repository was installed

Make Generate resources available

A typical Generate application will have all its resources bundled into its HIP file as part of the application Publish process.

New resources can be made available to Vault at any time simply by placing the appropriate HIP file in the ADM download directory.

Note that the resources are stored in sub-directories within this location, one per application version as determined by a unique identifier within the HIP file.

Other methods of loading resources

For datastreams with embedded resources a range of utilities is provided to extract them for use with Vault. Refer to the "Vault Customizing Guide" for further information on Vault utilities.

> **Note:** Resource files can be added while vault is running, but ensure that you restart Vault after changes are made to `server.ini` and `profile.ini`

Alternatively you may have the resources already available as independent files perhaps copied from your printer and browser environment. In both cases, store them either within the appropriate resource set or the default location by copying them into the appropriate Vault directory.

The Document Interchange Journal

A Document Interchange Journal (DIJ) is an index of documents contained within an output datastream file created by Generate. Every output datastream created by Generate that is intended to be stored in Vault. For example, documents for use with Service components must be accompanied by a DIJ file.

A DIJ is an XML file and is configured as part of a publication design in the Designer. You will need to specify parameters for the DIJ that allow Vault to identify the intended recipient of each document using an account number and its version using a 'statement date'. See “Interfacing with Vault” in the "Generate Users Guide" for details on creating and configuring a DIJ file as part of a Designer application design.

At production time Generate writes an entry to the DIJ file for each document it processes. If you use PCE or another post-composition process to merge output datastreams or to add/remove documents, you will need to ensure that the associated DIJ is updated to reflect such changes.

When creating a DIJ the relevant vendor, payment and document control information will normally be created automatically within the Generate environment.

In order to present documents correctly when they are viewed, the font and image resources with which the application was designed must be made available in Vault. Refer to the "Vault Customizing Guide" for information on working with fonts. Topics discussed include Font Embedding, Font Substitution, and PDF enhancement.

Non-Generate created print streams

If you are using non-Generate created documents or where you need to supplement existing resources manually, you will need to use an alternative method for loading resources into Vault. There are two possible routes: to extract embedded resources from a datastream or to place resource files manually in the appropriate vault directory.

Start Vault processes

In previous versions, all processes were started under a single service. The server, loader, render, router, and indexing (Indexer as a Service) processes are separate services and will appear in the services configuration as:

- Vault Server
- Vault Loader
- Rendering Engine
- Router

- Indexer as a Service

Separating the processes makes it easier to stop and start individual processes using standard Microsoft Windows tools, for example: "`net stop e2loaderd`" would stop ADM but not server or render.

On Microsoft Windows

For starting each service (or you can use the services applet in administrative tools):

- `net start e2serverd`
- `net start e2loaderd`
- `net start e2renderd`
- `net start e2router`
- `net start indexerd`

Starting manually

In a testing scenario the two main Vault processes can, if required, be started manually. When run in this mode all output from a process is displayed in the command window from which it was started. It is not recommended that processes be started manually in a production environment; always use the monitoring service, refer to the previous section for details.

Note:

- `e2loaderd` can be started manually with `-f` or `-b`
- `-b` is a batch mode that exits when there is nothing left to do
- `-f` starts the full loader in the foreground

manual start commands:

- `e2serverd -f`
- `e2loaderd -f`
- `e2loaderd -b`
- `indexerd -f`

Executable name changes

The following executables used from versions 5.3 and earlier have been renamed:

Old Name	New Name
render	`e2renderd.exe`
server -s	`e2serverd.exe`
server -a	`e2loaderd.exe`
server -c/-h/-i	`e2util.exe`

Configuring Encryption at Rest

If you plan using the Vault Encryption at Rest feature, you should plan for the deployment, installation and configuration of a encryption key server. See "Encryption at Rest " section in the Vault Customizing Guide for details.

Customizing the Vault environment

Customization of the Vault environment and the way client systems interact with the repository is controlled by a series of initialization files (INIs). A set of default INIs are created as part of Vault installation and for most installations the majority of the settings within the INIs can be left at their default values.

For information on customizing the Vault environment using the initialization files, refer to the "Vault Customizing Guide".

Working with Unicode data

In order to successfully load Unicode data into Vault you will need to provide either a Unicode enabled standard journal or an XML journal. Additionally, you will need to configure the indexes to support Unicode data.

Unicode journals

To specify a journal in Unicode enabled standard format, add the following profile options in `profiles.ini`:

```
[someprofile]
Documents=ujournal
JournalCodePage=<codepagename>
```

Where <`codepagename`> is the encoding used for the journal (for example, UTF-8, gb18030, windows-1252, shift_jis78, ISO-8859-1, etc.). The format of the journal has the ability to use a wide variety of encodings. This in turn allows you specify data in languages other than English. To specify a journal in XML format, add the following profile option in `profiles.ini`:

```
[someprofile]
Documents=uxmljournal
```

Note that the encoding is specified in the header of the XML file as per the XML standard.

Unicode indexes

There are two parts to each database in Vault: A single customer table and several indexes. Some indexes, like the account number index, are required for basic Vault operations, others are optional. Refer to the Vault Customizing Guide for information on custom indexing, this part of the configuration has remained unchanged. However the ability to use a Unicode enabled form of the customer table and/or indexes is now available. The Unicode customer table permits storage of names and addresses in any language. Unicode indexes let you search attributes containing data in any language. To use the normal customer table and index formats, no additional settings are required.

To use the Unicode customer table and index formats, with a single sort order, add the following database option in `database.ini`:

```
[somedatabase]
LanguageDefault=<sortorder>
```

To use a mix of normal and Unicode customer table and index formats or multiple sort orders, add the following database options in `database.ini`:

```
[somedatabase]
LanguageDefault=* or <sortorder>
LanguageN= * or <sortorder
```

Where:

`LanguageDefault` can either be * indicating normal customer table, indexes are normal by default or `<sortorder>` indicating a Unicode customer table, indexes will be Unicode using the specified sort order by default.

`LanguageN` can either be * indicating index N is normal or `<sortorder>` indicating index N is Unicode and uses the specified sort order.

Sort orders

The sort order specification supports a number of options but in most cases you can specify a minimum of options to get appropriate behavior. The typical form of the sort order is as follows:

`L<language>_R<region>_AS`

Examples:

Len_RUS_AS	English (United States)
Lzh_RCN_AS	Chinese (China)
Lzh_RSG_AS	Chinese (Singapore)
Lja_RJP_AS	Japanese (Japan)
Lko_RKR_AS	Korean (South Korea)
Lth_RTH_AS	Thai (Thailand)

Note that "_AS" is an option that reduces the significance of whitespace and punctuation which makes searching easier.

You can get a list of registered locales by running `e2util -xl` from the server directory. More detail on the collator options can be found here:

http://userguide.icu-project.org/collation/concepts

Notices

3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com