

Topic Sheet: CPC Vault Operations

Copyright © 2007 - ASC Process Systems, Inc.

Rev: 9/6/07

TABLE OF CONTENTS

1) General information.....	4
What is the vault?.....	4
Why use the vault?.....	4
Where should the vault be installed and run?.....	4
Setup Vault as DCOM.....	5
2) Vault Application	6
Installing the Vault.....	6
How to launch the Vault	6
Vault operations screen	6
File – Exit.....	7
File – Save.....	7
File – Open	7
Users	7
Last Connect.....	7
Last Comm	7
# of elements	7
Element selection	7
Element update	7
Element Contents	7
Remove	7
Remove All	8
3) Vault object	9
Vault Properties.....	9
Enabled.....	9
RemotePC	9
Key.....	9
Value.....	9
ObjectPath	9
Vault Reference.....	10
Status.....	10
Enabled.....	10
Vault methods	10

Connect	10
Disconnect	10
StoreValue	10
RetrieveValue	10
StoreObjectProps	10
RetrieveObjectProps	11
ClearAll	11
GetCount	11
4) Using the Backup object with the Vault.....	11
Backup Properties.....	11
Enabled.....	11
SourceObject.....	11
Filename	11
Interval	12
LastTime	12
NoChildren.....	12
LastTime	12
NoChildren.....	12
PropertyList.....	12
VaultPath	12
How to store/retrieve specific properties of a single object.....	12
How to store/retrieve specific properties of all child objects	13

1) General information

This document describes the new Vault capability in CPC and Flextime.

What is the vault?

The vault is a separate application that acts as a repository or storage for object properties and information on CPC/Flextime systems. Unlike Objects.g and other types of backup file storage, the Vault is designed to act in a real-time capacity without the overhead of disk operations.

Why use the vault?

The vault should be used in the following situations:

1. Use the vault if you want multiple CPC/Flextime systems to share data. Because the vault can be accessed by any system, information stored on the vault can be read and/or changed by any system on the network.
2. Use the vault if you want to store real-time information for restoration after system crash. Unlike the Objects.g file which is normally saved every 1-2 minutes, the vault can be used to store specific object properties every second or as they change. This data can then be loaded immediately on startup to provide bumpless restoration of the running process.

Where should the vault be installed and run?

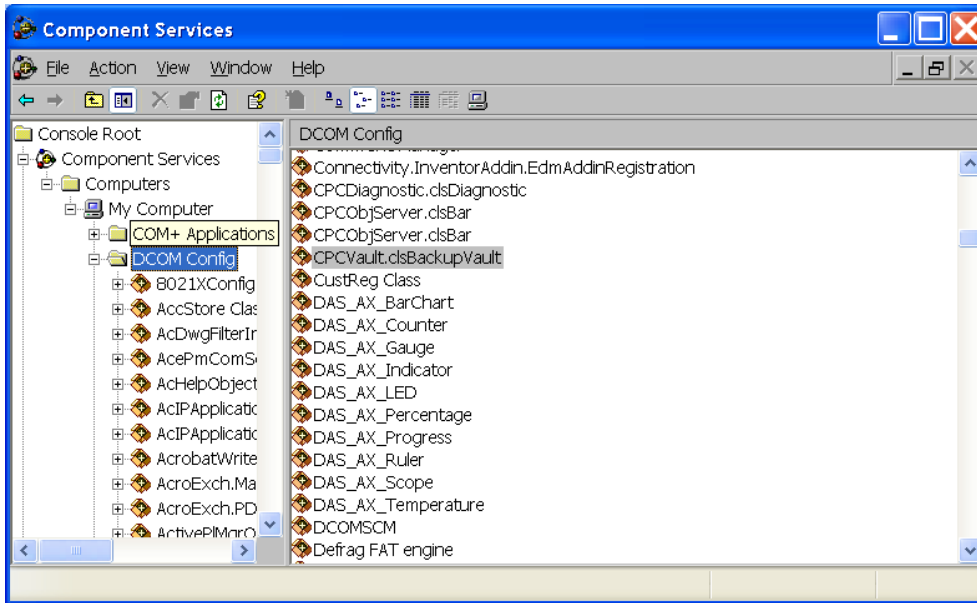
You only have to install Vault on equipment/systems that need the Vault.

On those systems, it is recommended to install the Vault on all system computers, although the Vault only needs to be running on a targeted PC. The Vault is run on the backup PCB of a dual-PC system or on the main PCA computer for a single-PC system.

Setup Vault as DCOM

If you intend on accessing a Vault on any remote computer (ie. PCB), than you'll need to setup the Vault as a DCOM application. Setup the Vault the same way as setting CPC for DCOM settings.

- 1.) Click **Start** button and then **Run** and then enter an run **dcomcnfg**
- 2.) Once the form is visible, expose the CPCVault object below.



- 3.) Right-click, select Properties, and then setup DCOM on this object
 - a. General tab, select AuthenticationLevel = "None?"
 - b. Location tab, select "Run this Application on This Computer"
 - c. Security Tab, select Custom on all options and Add "Everyone" on each section and click "Allow" on all options for the Everyone user.
 - d. Identify Tab, selection "Interactive User"

2) Vault Application

The following section describes the backup Vault used to collect information during a Flexitime session.

Installing the Vault

The vault application includes a Setup application that installs the vault in C:\Program Files\CPC Vault folder.

Because the vault is a DCOM application, its application (Vault.exe) will not be visible in the Programs selection from the Start menu. If you want to run the Vault, you can place it in the Startup folder or move a shortcut to the Desktop.

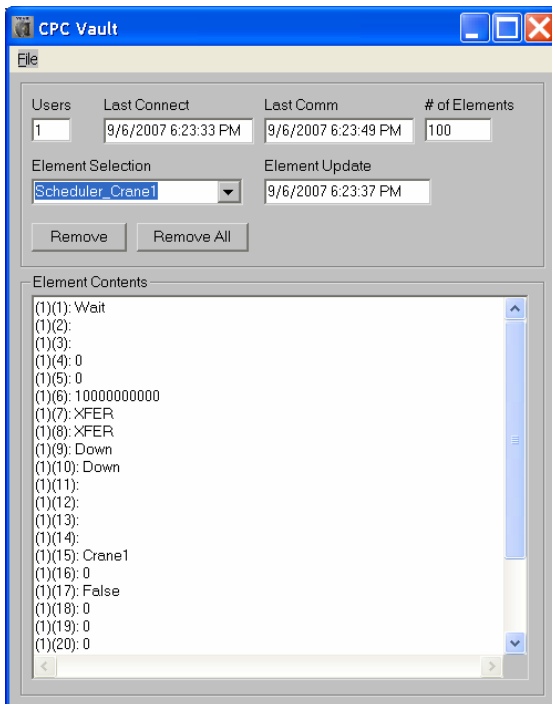


How to launch the Vault

You can either make a desktop shortcut to the CPC Vault.exe application or you can place that shortcut in the Startup folder for automatic launch on PC startup.

Vault operations screen

The following form shows the CPC Vault application.



File – Exit

To exit and shutdown the Vault, select File and then Exit.

File – Save

To save the currently stored information to a disk file, select File and then Save. Enter a filename and then press Save. The file extension is .vlt.

File – Open

To load from a previously saved disk file, select File and then Open to select the saved file. Click Open to load the data.

Users

This defines the number of connections to the Vault. Typically this will be “1” indicating that the PCA is connected.

Last Connect

This is the time of last connection to the vault.

Last Comm

This is the time of last communication to the vault. This will change every few seconds during real-time saving mode.

of elements

This indicates the number of data arrays and/or data elements current stored in the Vault. In normal operation, this value will be between 1000 and 1500 elements.

Element selection

This combo box holds the list of all elements stored in the vault. Each element is stored with a name key. You can scroll through this combo to view each element’s values.

Element update

This textbox shows the time of last update to the specifically selected element.

Element Contents

This element contents show the information stored for the specific element name. If the data stored is a single number or text, than that text will be displayed.

If the information was stored as a single-dimensioned array, than the information will be shown next to the index of the array. (ie. (5))

If the information was stored as a two-dimensioned array, than the information will be shown next to the double index of the array (ie. (1)(1)).

Remove

This button is used to remove a specific element. For most instances, this should not be used.



ASC Process Systems
14062 Balboa Blvd.
Sylmar, CA USA
www.aschome.com

Remove All

This button is used to remove all stored data. It is usually only used during testing and validation tasks.

3) Vault object

If the Vault application is installed and running on the local or remote PC, you can setup the Vault object in CPC.

Tankline	Name	Vault
SecurityManager	Enabled	True
Server	RemotePC	
Vault	Key	StationTimes
ClientDefault	Value	<ARRAY>
ActiveClients	ObjectPath	
Forms	VaultReference	
EventLog	Status	Ok
FileCopying	Connect	False
Recordsets	Disconnect	False
Production	StoreValue	False
Database	RetrieveValue	False
ICS1	StoreObjectProps	False
ICS2	RetrieveObjectProps	
ICS3	ClearAll	False
ICSTasker	GetCount	False
PLC		
Inhibit		
BackupSystem		

The Vault object is the communication object between a CPC/Flextime application and the Vault application.

Vault Properties

The following properties are provided:

Enabled

Enables the Vault operations

RemotePC

If you want to access the local Vault on the local PC, than leave this blank. If you want to access a remote PC's vault, than either enter that PC's name or its fixed IP address.

Key

This the named key that is used to store or retrieve the data from the Vault. Every element or array stored to the vault is stored with a key. Keys must not be a number, but can include any alpha-numeric characters. (ie. "Data1" or "TanklineStations", or ".Tankline.Cranes.Crane1")

Value

This is the information stored or retrieved from the vault. If the data was an array, than the property will show "<Array>".

ObjectPath

This is a CPC/Flextime path targeting the object properties to store. This is used for methods **StoreObjectProps** or **RetrieveObjectProps**.

Vault Reference

This property is currently unused.

Status

This shows the status of the last method call. Some of the entries are:

- Unable to connect – error during Connect method
- Connected – Connect method was successful
- Disconnected – Disconnect method was successful
- Ok – storage or retrieve was successful
- Invalid Key – retrieval failed because data does not exist for that key
- No connection – storage or retrieve methods couldn't find the connection
- Nothing to delete – key does not exist during a Delete method call.

Enabled

Enables the Vault operations

Vault methods

The following properties are provided:

Connect

Enables the Vault operations. Status will show "Connected" if successful.

Disconnect

Enables the Vault operations. Status will show "Disconnected" if successful.

StoreValue

This method stores the information in **Value** to the Vault using named key set in the **Key** property.

RetrieveValue

This method retrieves the named key information from the Vault using the **Key** property. The result is returned to the **Value** property.

StoreObjectProps

This method stores an array of all properties for the object selected in the **ObjectPath** property using the named key set in the **Key** property. *Note: This method stores all object's properties, including QuickScript text. If you want to store only specific properties for an object, than you should utilize the Backup object along with the Vault.*

RetrieveObjectProps

This method retrieves an array from the Vault using the named key set in the **Key** property. The returned array is automatically stored to the object targeted by the **ObjectPath** property. *Note: This method retrieves all object's properties, including QuickScript text. If you want to retrieve only specific properties, than you should utilize the Backup object along with the Vaultt.*

ClearAll

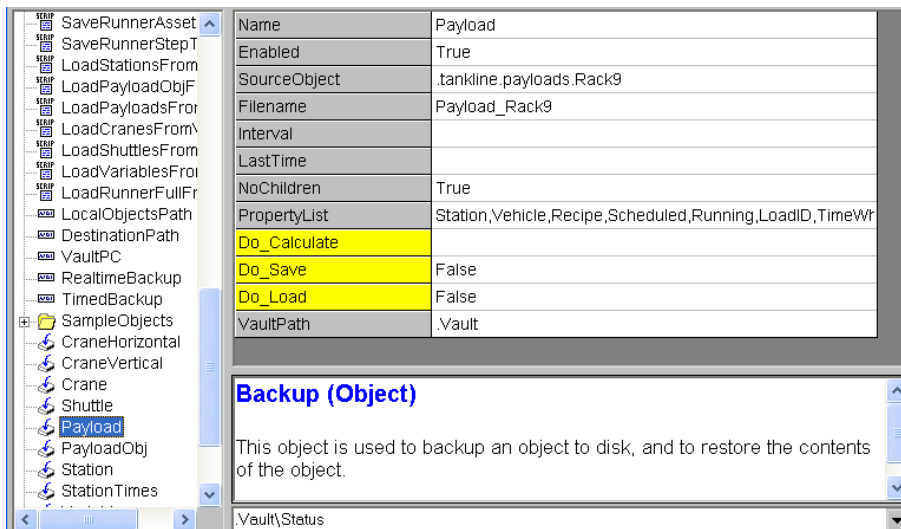
This method clears all stored information on the targeted Vault.

GetCount

This method retrieves the Vault's data storage element qty and returns the number to the Value property.

4) Using the Backup object with the Vault

CPC's Backup object is not only used to store information to disk files, but can also be used to store specific properties and information to the Vault.



Backup Properties

The following properties are provided:

Enabled

Enables the Backup operations

SourceObject

Path to object (Nochildren = true) or parent object (Nochildren=false)

Filename

When using the Vault, this becomes the named key for storage and retrieval.

Interval

This time is only used if you want the information stored at a predefined interval. It is used in conjunction with Do_Calculate method.

LastTime

This property is used by the system during Do_Calculate methods to determine when the next timed interval is.

NoChildren

This property is used to indicate whether you want to store/retrieve children or the parent object.

LastTime

This property is used by the system during Do_Calculate methods to determine when the next timed interval is.

NoChildren

If this property is True, than the Backup looks only at the targeted object and it's properties. If this property is False, than the Backup will store/retrieve only the children object's and their properties.

PropertyList

This is a list of properties to store and retrieve. If this is blank, than all properties for the object will be stored and retrieved. Note that in most cases you should define a limited list of properties in order to speed up the communications with the vault. In practice, you should not store and retrieve quickscripts properties, methods, the Name property, and any other property that is unchanging.

VaultPath

This should target the Vault object. It is normally ".Vault". Note that if this is blank, than the backup object will save to a file targeted by Filename.

How to store/retrieve specific properties of a single object

You can use the backup object to store specific properties of a specific object to the vault. This operation will store a single array of properties defined by the PropertyList .

In this case:

- VaultPath = path to Vault object (usually .Vault)
- SourceObject = Targeted object path
- Filename = key name to store information
- Interval, Lasttime = blank
- NoChildren = True
- PropertyList = comma delimited list of properties to store

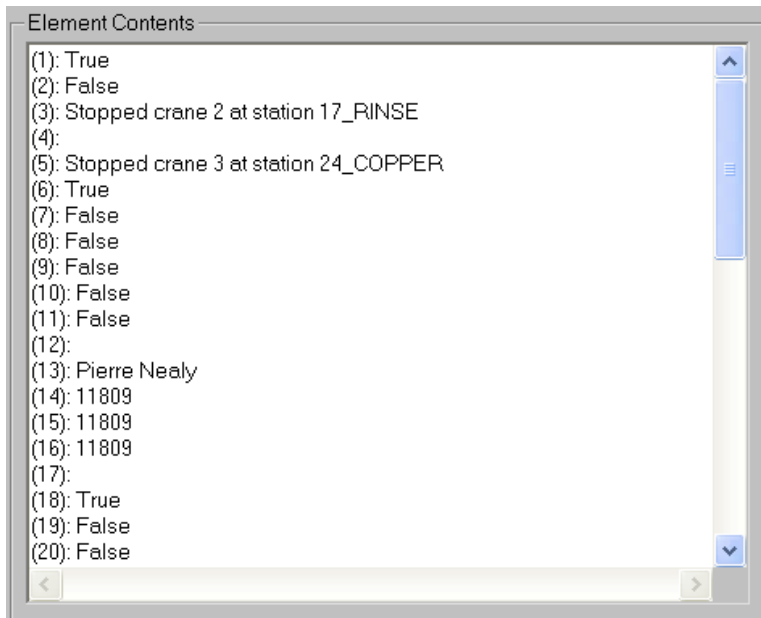
How to store/retrieve specific properties of all child objects

You can use the backup object to store specific properties of children objects (ie. Variables). This operation will store a 1-dimensioned array containing 1-dimensioned arrays of properties.

In this case:

- VaultPath = path to Vault object (usually .Vault)
- SourceObject = Targeted object path
- Filename = key name to store information
- Interval, Lasttime = blank
- NoChildren = False
- PropertyList = comma delimited list of properties to store

If the property list is a single property (ie. Value), than this operation will store only a 1-dimensioned array of that one property. For example, the following shows an example of storing the Value properties of 20 variables (SourceObject = .Tankline.Variables):



If the property list denotes multiple properties, than the operation will store a 1-dimensioned array containing 1-dimensioned arrays of property values. In this case, the Vault will only show the properties for the first object and last object. For example, the following shows an example of storing the Name and ElapsedTime properties of 71 station objects (SourceObject = .Tankline.Stations).

