KB00107 Terminal Concentrator Technical Differences

This article provides additional details about the technical differences between different uses of TC.

The differences can be broken down into the following categories

- Where TC is relative to the 4690 controller and files.
- Whether or not and how other applications running in the concentrator interact with TC or the sales application.
- Method of I/O device emulation or access.
- If I/O device access is provided by the TC Remote Peripheral Access Method (RPAM), the type of communication that is used between the client terminal and the TC server.
- If I/O device access is provide by RPAM, the terminal type of the client.

The information in this article applies to:

- QVS Terminal Concentrator for Windows
- QVS Terminal Concentrator for 4690

MORE INFORMATION

Where TC is relative to the 4690 controller and Files:

Remote Access - the TC box accesses files on the 4690 controller(s) as if it were up to 64 separate LAN terminals.

CSF/TC - CSF and TC can run in the same server. TC accesses the 4690 controller files through a "glue" interface to CSF.

4690/TC TC runs on the 4690 controller. TC accesses the 4690 controller files through a glue interface to 4690 terminal services layers.

Standalone/Local - all of the 4690 controller files are copied onto the TC server. TC accesses these files locally. TC is not involved in keeping the files in sync with the actual 4690 controller. Keeping the files in sync managed in batch mode by the integrator. In one case where this method was used it was used because the 4690 controller did not have the capacity to support real-time access of its files from the TC node.

Whether or not and how other applications running in the concentrator interact with TC or the sales application:

In one implementation, a customer has his own server application that runs on the TC box and communicates directly with the sales application using standard 4690 pipes. Relatively heavy user exit work was done in the application.

In another implementation, TS's Monitor/Originate pipe interface is being used in TC. The interface has been enhanced for TC by including a relative application # and terminal # in the message definitions. There is only one pair of pipes used for all messages (i.e. not a pair per terminal).

Method of I/O device emulation or access:

No physical devices (all virtual) - display outputs go to a section of the NT screen, cash drawer operation is emulated, prints all get good return codes, etc. Input can be provided by the originate pipe. Output is forked to the monitor pipe.

Devices are "remoted" using RPAM. RPAM is the more full-blown TC implementation. With RPAM the idea is that real devices exist but not on the same machine. There is a client and a server part to RPAM. The client side is known as TERMRPAM. The interface between the client and the server is through a set of defined messages. The RPAM implementation has support for switching to a backup concentrator, printer error reporting and handling, hard totals maintained in the client, provisions for an offline application in the client. Note that only devices that have been used by our customers up to now are supported. So far, this has NOT included Scale (implemented but not tested), Logo Printing, Model 3/4 Printer (implemented for demo but not fully tested), Model 4610 Printer, customer display, keyboards other than the IBM 50 key.

If I/O device access is provided by the TC Remote Peripheral Access Method (RPAM), the type of communication that is used between the client terminal and the TC server

Sockets (TCP)

4690 PRS pipes

Proprietary (such as Fujitsu)

If I/O device access is provide by RPAM, the terminal type of the client

RPAM clients have been written for NCR 7052 terminals, FSA 8770 terminals, PSI DOS terminals, IBM 4683 terminals, Windows and Windows/CE-based clients, PalmOS-based clients and Linux clients. There are 5 different API types that clients can use to interface with TC: RPAM Java Bean, RPAM COM interface, RPAM COM/XML interface, QVSRPAM.DLL interface, and RPAM message interface using sockets APIs. The last two of these are discouraged as much as possible.