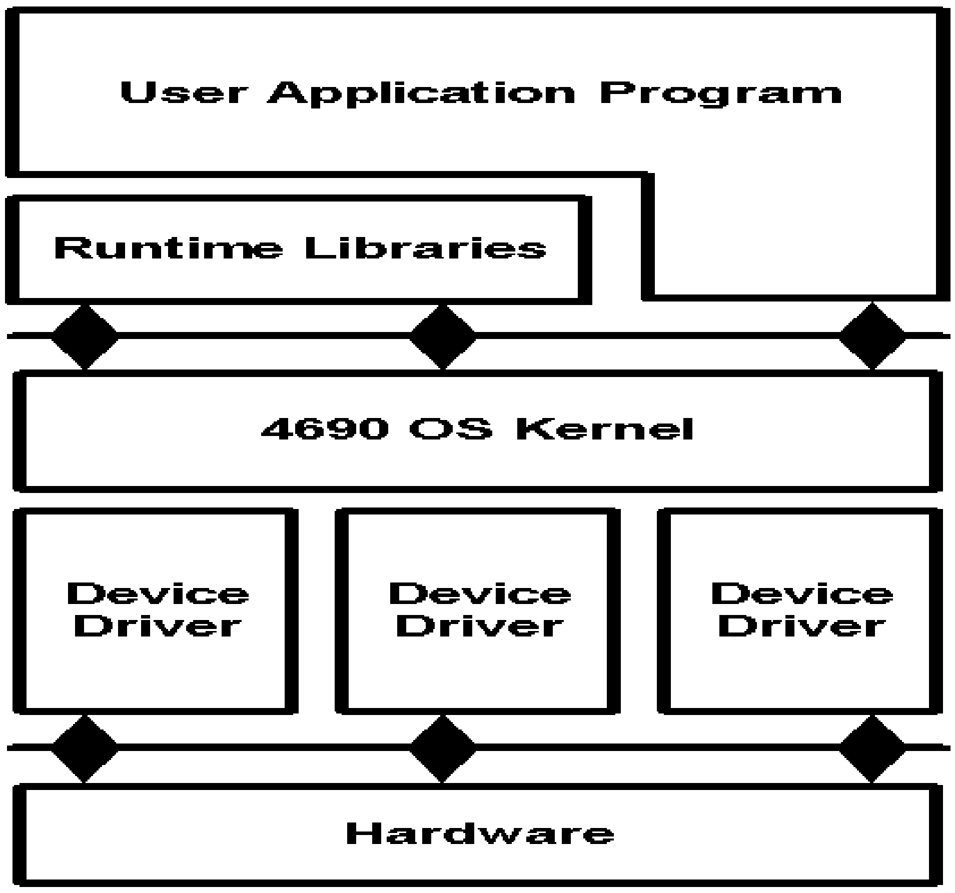
Terminal Concentrator for Windows Technical Overview

What is Terminal Concentrator?

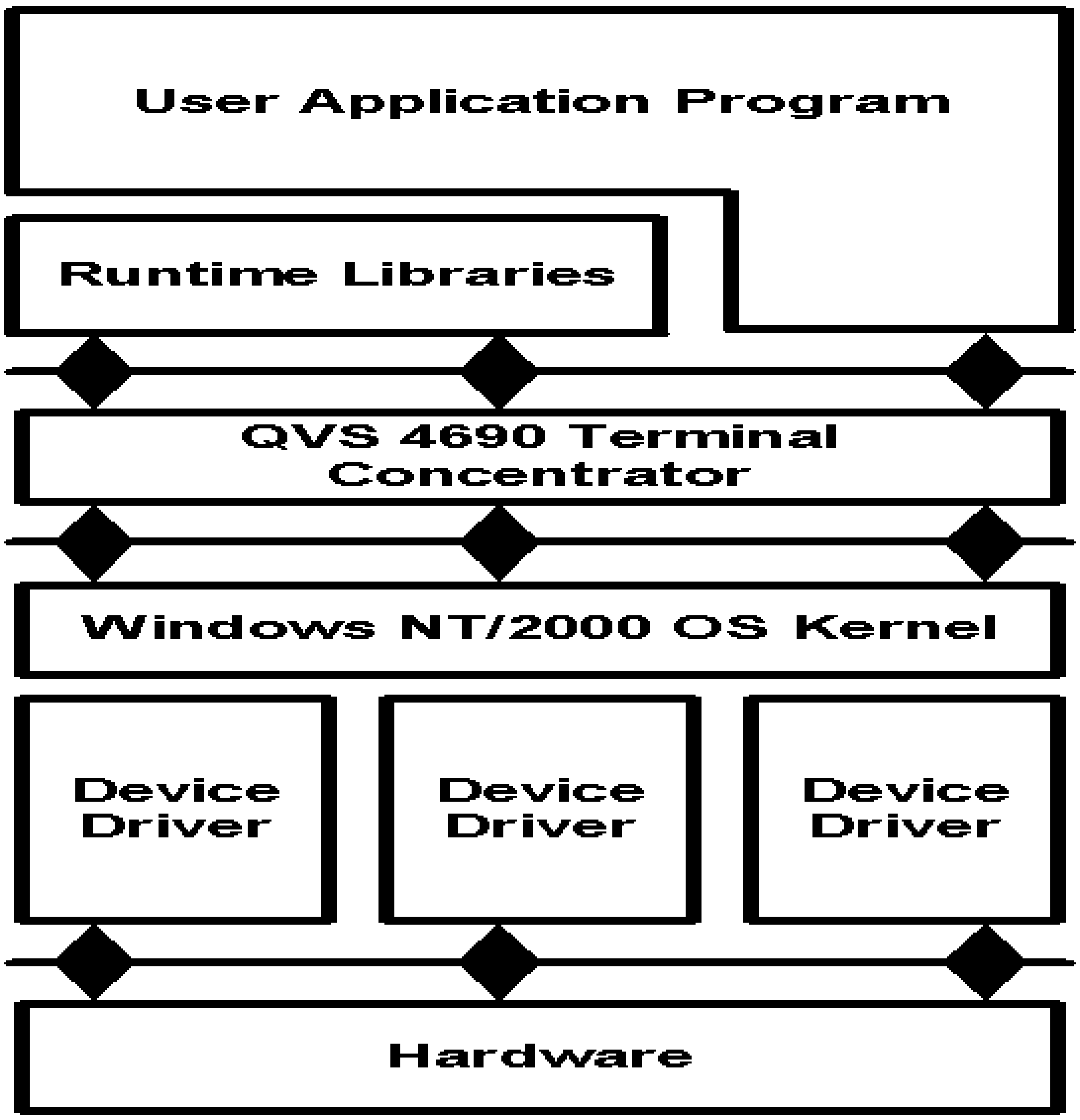
- TC is a 4690 OS Emulator running on Windows NT/2000
- Provides the OS services required to run a 4690 terminal sales application on Windows NT/2000
- Allows users to run up to 64 instances of the terminal sales application on a single server
- Supports thin client POS terminals through remote device I/O protocol called "Remote Peripheral Access Method" (RPAM)
  (All device level I/O is remoted out to a separate process or computer. It's the responsibility of the remote process to actually perform the requested I/O operation and return the results back to the terminal sales application through the RPAM interface)
- Supports client any device for which an RPAM I/O catcher can be written. QVS has written I/O catchers for devices running Windows, Windows CE, Linux, and PalmOS

How does Terminal Concentrator work?

Native 4690 Environment



Terminal Concentrator Environment



- Implemented as a group of cooperating user mode Windows processes and a kernel mode component that allows QVS to load and execute a 4690 .286 load format executable under Windows

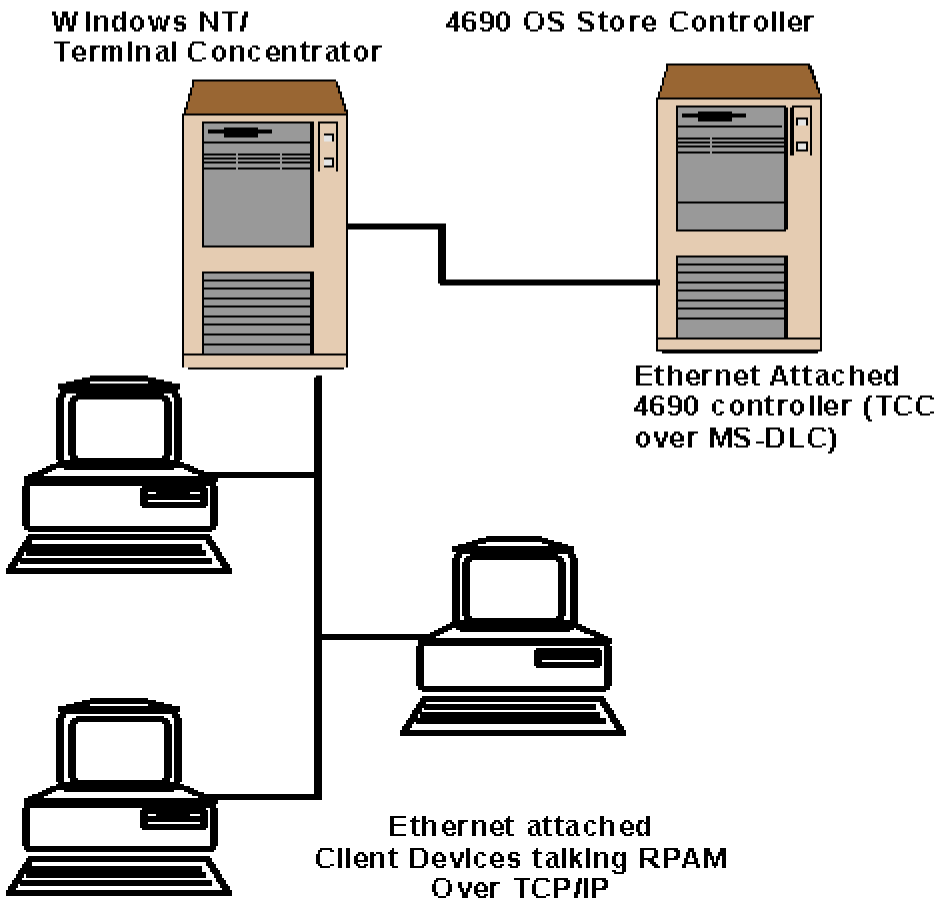
- Utilizes Windows sockets API to communicate with RPAM client devices connected on a traditional Ethernet or wireless Ethernet LAN

- Communicates with an Ethernet attached 4690 controller using DLC protocol or through shared memory with CSF in integrated TC/CSF environments

- Traps all 4690 OS calls made by the terminal sales application (.286)

- Carries out the requested service using native Windows OS primitives (file access, LAN access, etc).
- Returns the results back to the .286 the same way 4690 OS would.

Typical TC Installation (Connected to a remote Store Controller)



TC Configurations

- Integrated with CSF
  In this mode TC runs in the same box as IBM's 4690 Controller Services Feature. Allow you to run a complete store environment in 1 box without the need to have a separate 4690 controller

- Connecting to a Remote 4690 controller
  In this mode TC connects to a remote store controller (running 4690 OS or CSF) over a LAN using the DLC protocol.

- Standalone Mode
  No 4690 controller or CSF is required. TC uses native Windows NT/2000 functionality to access POS files. Advantage is TC runs self contained. Disadvantage is no 4690 controller functionality such as Background Sales is available. Good for limited functionality demos/development environments

Installing and Running TC

- Installing TC is relatively straightforward
  No 4690 controller or CSF is required. TC uses native Windows NT/2000 functionality to access POS files. Advantage is TC runs self contained. Disadvantage is no 4690 controller functionality such as Background Sales is available. Good for limited functionality demos/development environments

- Main Installation question revolves around how TC will communicate with the 4690 controller (or CSF)

- Connecting to a remote store controller (Note: You must have DLC protocol installed on the TC server prior to installing TC in this configuration)

- Running Integrated with CSF

- Number of terminals. QVS supplied license key required to run more than 4 instances of the terminal sales application.

- Operator display type. Relates to what type of physical display the terminal sales application expects to use as the system display.   Either 2x20 POS display or full screen (CBASIC VDISPLAY device).

- TC typically runs as a Windows NT/2000 service application
  (In a production environment you almost always want to run TC as a service but it can also be run as a standard Windows NT user mode program. This is often more convenient in lab/development environments)

- Provides for automatic restart of Terminal Concentrator in the event of an abend. Can be enabled and disabled from the Start menu.

- Provides a mechanism whereby historical generations of trace log information can be archived

- Allows TC to be started even when no user is logged on to the system (Automatic service start)

- QVS has implemented mechanism in the service start/stop activities that allow user to add custom commands and applications.
  (You can run applications or commands before and or after TC is started)

TC Problem Determination

- TC has extensive tracing facilities that allow QVS to quickly analyze problems

- TC tracing has a minimal impact on performance. It’s completely practical to leave TC tracing turned on in lab/development environments

- Enabling Logging at TC IPL
  You can enable TC logging using the Start menu option to enable logging at next IPL. You must stop and restart TC before logging will be activated)

- Enabling logging when starting TC from the command line
  Use the '-t' command line option when running TC from the command line. Has the same effect as enabling logging from the Start menu.

- Toggling TC logging from the command line.
  Useful for situations where a great deal of activity must be executed to recreate a problem. Allows you to run without tracing on up to the point where a failure occurs. Then you can turn tracing on to capture the problem. Can cut down considerable on the size of the trace files generated.

- Type of trace files generated by TC
  It's important to have a basic understanding of the trace files generated by TC so you can be sure to send a complete, readable set of logs to QVS when needed.

- TC generates .bin, .db and .pan files.
  .bin files are the compressed binary log files that hold the trace information generated by TC.
  .db files are the message database files required by the QVS log formatter to convert the .bin files into readable text files.
  .pan files are files generated by the system when an application or a TC process abends.

- Without the corresponding .db file, the .bin files are unreadable. When sending logs you must include a .db with the bin files.

- .pan files are generated when an abend occurs regardless of whether TC was in trace mode or not.   Contains critical system information regarded why and where the abend occurred as well as a small trace buffer for each process in the system.
  .pan files can also be generated from the command line using the "–k1" command line switch.

- Each TC process logs trace information to a separate log file.

- You can selectively enable tracing for a specific process using the "-txxx" command line switch. 'xxx' represents the process number you wish to trace.

Interpreting TC Log Files

- TC trace logs are used most often by QVS personnel to aid during problem determination

- While it's not strictly required that a third party developer be able to interpret the log information it can be useful to find out what an application is actually doing.

- Formatting log files
  TC logs are compressed binary files that must be formatted by the log formatter before then can be read.
  To format the logs you run the program 'e46dumpf' with the '-a' command line switch

- E46DUMPF output files.

e46dumpf creates .blg files from .bin files. These are human readable text files containing the TC trace logs. Depending on the size of the .bin files you may end up with more than one text output file. In these situations .bl1 … .bl4 files may be generated as well. Most recent log information is contained in the .blg file with successfully older trace information in the .bl1 through .bl5 files.

Basics of Reading Logs

- One process per log file
- Main TC screen tells you which log file a given instance of the terminal sales application is using.
- Handles in TC trace files.
  All I/O by a terminal sales application is done through handles.
- Finding the handle for a given device or file.
- Search for where the application opened the device or file.
- Look at the return code from the open command. This is the file handle for the device or file.
- Tracking application use of a given handle.
- Application will use the handle for all reads and writes to the device as well as 4690 OS special calls.

Lab Issues

- Lab setups sometimes require special TC configuration settings to allow multiple concentrators and multiple store controllers to coexist on the same LAN.
- When multiple controllers are present, TC allows you to configure which controller to use by setting store number.
- To specify that you want TC to connect to a particular store you set the “storeNum” variable in the QCDIFILE.
- If you set store number for a concentrator you also need to set it for clients which will be communicating with that instance of TC.
- Under Windows you use the registry.
  On the client machine to specify a store number value using \HTLM\Software\QVS\TSF_STORE_NUM registry key.
- Java clients specify store number through the RPAMApp object
- It’s also possible to control which instance of TC you would like a terminal to connect to.
- Often useful when multiple developers will be accessing TC a single instance of TC.
- Terminal access is controlled using the “tcPrimaryServer” setting in QCDIFILE in conjuction with condition directives
- Example:
  Terminals 1-20 should connect to one instance of TC while all other terminals should connect to another instance of TC
  The first TC machine would have tcPrimaryServer configured as follows in it’s QCDIFILE:

```
#if TERM_NUM >=1 && TERM_NUM <= 20 then
   tcPrimaryServer=”1”
#else
   tcPrimaryServer=”0”
#endif
```
The second instance of TC could be configured as:
```
#if TERM_NUM >20 then
   tcPrimaryServer=”1”
#else
   tcPrimaryServer=”0”
#endif
```

- Unless otherwise specified, the default action for TC is to respond to any and all clients.
  To aid setting up a lab environment, TC client programs will abend when the try to connect to TC in an environment where there is more than one TC primary server configured

Overview of Programming with TC

- Two major APIs supported
- COM API for Windows environments
- COM API available as in-process or out-of-process server
- Java API for Windows and Linux clients
- PalmOS library
- Java and COM APIs are very similar
- RPAMTerminal object
  Provides the application with a high level interface to the device level I/O performed by a 4690 sales application. Allows the client application to feed input to the 4690 application and monitor and respond to output performed by the 4690 sales application.
- RPAMTerminal object contains sub-objects representing the various physical and logical devices used by the terminal sales application

ANDisplay/ANDisplay2

CashDrawer

IOProcessor

MSR

Printer

TermControl

Tone

TotRet

VDisplay1/VDisplay2

- RPAMApp Object
  The application object contains system-wide settings that control all RPAMTerminal objects instantiated. You can use the application object to change the default behavior of the RPAM interface layer.

- Accessing POS files on the controller using the POSFile and POSKeyed file objects