

Automated Serial Port Interaction with Video Commander 3.5

This document describes the Serial Communications program in Video Commander 3.5, and specifics of using that option as a means of controlling, and querying, Video Commander from another computer or embedded controller.

Controlling Video Commander

At its simplest, the Video Commander Serial Communications program (VCComm.exe) offers a real-time command line interface into Video Commander, using the Video Commander Macro Language. An external system may issue commands to force the Video Commander system to perform certain high-level (device level) routing or control operations.

Note: IRIS does *not* recommend any external equipment trying to directly command the routers themselves. Not only are IRIS routers not designed to accept commands from multiple sources, but Video Commander would not be notified of any changes sent directly to the router.

The Video Commander Macro Language and the full set of commands are detailed in the file **vcmacro.hlp**, included with your Video Commander installation. This document also details which commands are available to the Serial Communications program; commands dealing with flow of control, time delays and server diagnostics are *not* supported in (or relevant to) Serial Communications.

In the macro language are many commands using a device name. One useful detail to note for external control is that a device may be specified by name *or* number. Thus the human-friendly command `Unlink "Main/VCR 3"` may be replaced with `Unlink #17` (possibly simpler for an external computer program). A device ID is distinguished by always starting with the number sign ("#"). Refer to the Macro Language Overview in **vcmacro.hlp** for general syntax rules.

In many cases the number is the better way to refer to a device for external control: it is faster to send (or receive); when inquiring about routes (described below), it is easier to parse a numeric response; and also, the IDs remain consistent even if a device is renamed.

An automated system using VCComm should be coded to watch for any string received that begins with the exclamation point character. This signifies any message or notification other than a successful result of a request. For example, sending a bad command in verbose response mode generates `!Command error` as the result. Some of these notifications are completely unrelated to commands and could occur at any time; the `!Off Line` message indicating Video Commander server shutdown could occur in the middle of sending a command. A complete list of responses appears at the end of this document.

Querying Video Commander

The Serial Communications program has some special commands to interrogate

high-level routes as well as initiate them.

Getting Current Routes

The first part of this is the ability to query Video Commander about the current status of a device. This allows an external program, at startup, to synchronize with what is already routed in Video Commander. The request is simply:

```
Query Device device
```

As with other commands, *device* may be a name (usually in quotes) or a number preceded by "#." The response is the list of sources currently routed to the queried destination, if any; this might look like any of the following:

```
#17=#58
#17=[V]#58,[LR]#62
"(4)"=[V]"Titler 2",[LR]"(CBS) WXYZ"
"(4)"(17)=[V]"Titler 2"(58),[LR]"(CBS) WXYZ"(62)
```

There are two options which affect what is displayed. First, is the inquiry about just one signal plane (video) or multiple planes? The first example showed just the video source for channel 4 (destination device 17); the other examples show both video and audio, with macro language breakaway syntax specifying what devices appear on which signal types. The second option is, how is a device represented? This could be a number (first two examples), a name (third example), or both (fourth example).

These options may be configured as part of VCComm setup, as described later in this document, or specified on the fly using commands. To specify the signal type(s) to query, use one of these commands:

```
Display Level level
Display Level [brkaway]
Display Level *
```

Use *level* as a name (Video, for example), *brkaway* for any valid combination of breakaway specifiers (LR, for example), or "*" for displaying all planes. In the first query example given earlier, only one plane was requested so VCComm did *not* display the breakaway specifier in the answer. For the other examples it was necessary to describe which planes were meant since several were being queried.

To specify whether name, number or both should be used in responses, change the VCComm port configuration (described later in this document), or use one of these commands:

```
Display Device Number
Display Device Name
Display Device Both
```

These commands, and the Display Level commands, affect subsequent queries.

Monitoring Routes

It is not a good use of communications time for an external program to keep querying Video Commander repeatedly, so Serial Communications also offers the ability to ask Video Commander to *announce* changes as they happen. Since this is not the primary function for which VCComm was created, the serial line user or external computer must request this behavior.

To request notifications when a device is routed or unrouted, use one of the following commands:

```
Watch Device device
Watch Device *
```

The *device* may be any single device name or number; this command may be issued any number of times to build up a list of devices the external system cares about. Alternately, the second form can be used to get notifications on *all* devices, requiring the external system to filter out the devices it doesn't care about.

For completeness, it is worth listing the reciprocal commands for *stopping* this notification process. These are:

```
Ignore Device device
Ignore Device *
```

(The Watch Device and Ignore Device commands in VCComm are *not* the same as the Watch Device and Ignore Device commands supported at the server's command line window.)

When some device or devices are set up to be watched, any route involving that device (as a source *or* destination), or an unroute where that device *was* the destination, will be announced to the serial port. Like errors and notifications described above, these notifications could arrive at any time, even in the middle of entering a command. They are always preceded by the exclamation point, and look something like this:

```
!Link #58,#17
!Unlink #17
```

The first example describes a route from device 58 to device 17; the second describes an unroute or job close in which device 17 lost its source. Note that even when a route with multiple destinations occurs, the notifications appear as separate transactions for each destination, and those involving no devices being watched will not be sent out.

The planes which are monitored, and the representation of the devices (name, number or both) are as described earlier for the Query Device command.

Setting Up VCComm

There are a number of configuration options for the VCComm program. It is worth briefly reviewing these as they relate to an external process using the serial port for interaction with Video Commander.

The port and its configuration must be set up in order to communicate properly. (Note that VCComm supports multiple ports used for this purpose on the same computer.) The options "Echo characters," "Verbose messages" and "ANSI codes" are intended mostly for human interaction through the serial line, and are not checked for an external system.



The automatic log-in is important for an external task, so the VCComm program doesn't require the external task to enter the user ID when the server restarts. (It is harder to synchronize the programs if this is the external task's responsibility.) Use the ID which should be associated with all route commands and requests.

The status feedback display section sets the initial settings which would otherwise be controlled by the Display Device and Display Level commands. For most external systems there is no reason to change these settings; numeric ID is simplest to react to, and in a case where the external system is synchronized to, for example, video routes, the selection of just that plane simplifies the notification stream *and* the job the external task must do to interpret this information.

VCComm Notifications

The following is a complete list of the notification messages sent by VCComm to the serial line, and what each one means.

?

This is a simple command error. In verbose mode, this is instead `!Command error`.

!

This is an error reported by the Video Commander server. In verbose mode, this is instead `!VC Comm error`.

`!Can't open network link`

VCComm could not open a network connection to the Video Commander server, so an apparently valid command could not be processed.

`!Command error`

The last Video Commander Macro Language command could not be interpreted, and no request was made of the server. Verbose messages must be on for this message; the brief version is simply `?`.

`!Link src,dest`

A link has occurred between the two specified devices. The format of the devices depends upon the Display Device setting currently in effect.

`!Off Line`

The Video Commander server was shut down. If verbose messages are on, the `!Server has shut down` message appears instead. Also, this message appears if the comm channel was

closed by shutting down VCComm, or as part of reconfiguring an active port; in this case, this message appears regardless of the verbose setting.

`!On Line`

The Video Commander server has just fired up, or VCComm has just started.

`!Recall State`

The Video Commander server has just executed the Recall State command, which means any number of routing changes may have just occurred simultaneously, for which `!Link` and `!Unlink` notifications were *not* broadcast. Any program seeking to track all routing changes to certain devices should query these devices again, the same as at startup.

`!Server has shut down`

The Video Commander server was shut down. Verbose messages must be on for this message; the brief version is simply `!Off Line`.

`!Unlink dest`

The specified device has just lost a source connection. The format of the device depends upon the Display Device setting currently in effect.

`!VC Comm error: nnnn`

An error was indicated by the Video Commander server. The number is a bit mask representing various common error conditions. (These error codes are included in the "Video Commander Error Codes" Technical Note.) Verbose messages must be on for this message; the brief version is simply `!.`

`!VC Comm error on network`

An error was reported by the network communications code. When this occurs it is impossible to tell whether the last command was actually executed or not. Verbose messages must be on for this message; the brief version is simply `!.`