

Switcher Products



Protocols Manual

Software Version

KEMA Quality

CERTIFICATE

Certificate Number: 510040.001

The Quality System of:

Grass Valley USA, LLC and its Grass Valley Affiliates

Headquarters:

**400 Providence Mine Road
Nevada City, CA 95945
United States**

**15655 SW Greystone Ct.
Beaverton, OR 97006
United States**

**Brunnenweg 9
D-64331 Weiterstadt
Germany**

**Kapittelweg 10
4827 HG Breda
The Netherlands**

**2300 So. Decker Lake Blvd.
Salt Lake City, UT 84119
United States**

Including its implementation, meets the requirements of the standard:

ISO 9001:2008

Scope:

The design, manufacture and support of video and audio hardware and software products and related systems.

This Certificate is valid until: June 14, 2012
This Certificate is valid as of: December 23, 2010
Certified for the first time: June 14, 2000

H. Pierre Sallé
President
KEMA-Registered Quality

The method of operation for quality certification is defined in the KEMA General Terms And Conditions For Quality And Environmental Management Systems Certifications. Integral publication of this certificate is allowed.

KEMA-Registered Quality, Inc.
4377 County Line Road
Chalfont, PA 18914
Ph: (215)997-4519
Fax: (215)997-3809
CRT 001 042108

Accredited By:
ANAB



a  **DEKRA** company



Switcher Products

Protocols Manual
Software Version

Contacting Grass Valley

International Support Centers	France 24 x 7	+800 8080 2020 or +33 1 48 25 20 20	United States/Canada 24 x 7	+1 800 547 8949 or +1 530 478 4148
Local Support Centers (available during normal business hours)	Asia	Hong Kong, Taiwan, Korea, Macau: +852 2531 3058 Indian Subcontinent: +91 22 24933476 Southeast Asia/Malaysia: +603 7805 3884 Southeast Asia/Singapore: +65 6379 1313 China: +861 0660 159 450 Japan: +81 3 5484 6868		
		Australia and New Zealand: +61 1300 721 495	Central/South America: +55 11 5509 3443	
		Middle East: +971 4 299 64 40 Near East and Africa: +800 8080 2020 or +33 1 48 25 20 20		
	Europe	Belarus, Russia, Tadzikistan, Ukraine, Uzbekistan: +7 095 2580924 225 Switzerland: +41 1 487 80 02 S. Europe/Italy-Roma: +39 06 87 20 35 28 -Milan: +39 02 48 41 46 58 S. Europe/Spain: +34 91 512 03 50 Benelux/Belgium: +32 (0) 2 334 90 30 Benelux/Netherlands: +31 (0) 35 62 38 42 1 N. Europe: +45 45 96 88 70 Germany, Austria, Eastern Europe: +49 6150 104 444 UK, Ireland, Israel: +44 118 923 0499		

Copyright © Grass Valley USA, LLC. All rights reserved.

This product may be covered by one or more U.S. and foreign patents.

Grass Valley Web Site

The www.grassvalley.com web site offers the following:

Online User Documentation — Current versions of product catalogs, brochures, data sheets, ordering guides, planning guides, manuals, and release notes in .pdf format can be downloaded.

FAQ Database — Solutions to problems and troubleshooting efforts can be found by searching our Frequently Asked Questions (FAQ) database.

Software Downloads — Download software updates, drivers, and patches.



END-OF-LIFE PRODUCT RECYCLING NOTICE

Grass Valley's innovation and excellence in product design also extends to the programs we've established to manage the recycling of our products. Grass Valley has developed a comprehensive end-of-life product take back program for recycle or disposal of end-of-life products. Our program meets the requirements of the European Union's WEEE Directive, the United States Environmental Protection Agency, and U.S. state and local agencies.

Grass Valley's end-of-life product take back program assures proper disposal by use of Best Available Technology. This program accepts any Grass Valley branded equipment. Upon request, a Certificate of Recycling or a Certificate of Destruction, depending on the ultimate disposition of the product, can be sent to the requester.

Grass Valley will be responsible for all costs associated with recycling and disposal, including freight. However, you are responsible for the removal of the equipment from your facility and packing the equipment to make it ready for pickup.



For further information on the Grass Valley product take back system please contact Grass Valley at + 800 80 80 20 20 or +33 1 48 25 20 20 from most other countries. In the U.S. and Canada please call 800-547-8949, and ask to be connected to the EH&S Department. Additional information concerning the program can be found at: www.grassvalley.com/about/environmental-policy

001187401

Contents

Section 1 — Tally Protocol	9
Introduction	9
Kalypso System Overview	9
Zodiak System Overview	9
Kayenne System Overview	10
Tally Background Information	10
Tally Types	11
Output Tally	11
On-air Tally	11
Look-ahead Tally	11
Iso Tally	12
Selection Tally	12
Tally Calculation Basics	12
Keyer Modifications	12
Transition Mixer Modifications	13
Internal and External Processing Loops	13
Tally System	13
Names	14
Tally Relay Outputs	14
Serial Tally	15
Changing Tally Port Serial Settings	15
Kalypso & Zodiak Contribution Tally Protocol	16
Source IDs	17
ME Contribution Information	18
External Processing Contribution Information	19
Still Store Contribution Information	20
Output Contribution Information	21
Kalypso Systems	21
Zodiak Systems	22
Source Names	23
Update	25
Command Codes and Instance Summary	25
Message Structure	26
Communication Specifics	26
Message Parsing and Processing	28
Kayenne Contribution Tally Protocol	30
Message Structure and Summary	30
Source IDs	31
ME Contribution Information	32
eDPM Contribution Information	35
Image Store Contribution Information	36
Output Status	36
Source Names	37
Update	38
Message Parsing and Processing	39
Section 2 — Editor Protocol	41
Kalypso/Zodiak Editor Protocol	41

Introduction	41
Serial Data Word Description	41
Editor Protocol Access	42
Break Character	42
Address Byte	44
Command/Message Block Structure	44
Byte Count	45
Effects Address Byte	45
Command Code Byte	45
Status Replies	46
Error Detection	46
Editor Commands	47
Introduction	47
Command Usage	48
Application Examples	48
Source Bus (C1 – C4) Commands	50
Kalypso Effects Addresses	51
Zodiak Effects Addresses	52
Kalypso Source Numbers	53
Zodiak Source Numbers	54
Source Select (C0) Command	55
Kalypso Effects Addresses	55
Zodiak Effects Addresses	57
Preview Bus (E2) Command	57
Source Numbers	57
Split Key (E4, E5, E6, E7, E9 and EA) Commands	58
Kalypso Effects Addresses	58
Zodiak Effects Addresses	58
Source Numbers	59
Pushbutton Select and Control (C6, C7 and FB) Commands	59
Kalypso Effects Addresses	60
Zodiak Effects Addresses	60
Kalypso Pushbutton Numbers	61
Zodiak Pushbutton Numbers	62
Wipe Pattern (C8) Command	63
Kalypso Effects Addresses	63
Zodiak Effects Addresses	63
Wipe Numbers	64
Transition Mode (CA) Command	65
Kalypso Effects Addresses	65
Zodiak Effects Addresses	65
Mode Byte — M/E	66
Mode Byte — DSK (Zodiak Only)	66
Auto and Key Transition Rate (CC and CD) Commands	67
Kalypso Effects Addresses	68
Zodiak Effects Addresses	68
Transition Rate	68
Learn and Recall E-MEM Register (DA and DB) Commands	71
Kalypso Effects Addresses	72
Zodiak Effects Addresses	73
Learn Mode Byte	73
Recall Mode Byte	74
Registers	75
4000 Bit-Mask Format	75

Kalypso Bit-Mask Format	76
Zodiak Bit-Mask Format	77
Save and Load Data (5F and DF) Commands	78
Effects Addresses	78
Reg Number	78
Name	78
Save and Load Status (6D and ED) Commands	82
Effects Addresses	82
Status Message	82
Timeline Control (4E, 4F, CE, and CF) Commands	83
Kalypso Effects Addresses	83
4000 Bit-Mask Format	84
Kalypso Bit-Mask Format	84
Zodiak Bit-Mask Format	85
Data Field Format	86
All Stop (F2) Command	88
Kalypso Effects Addresses	88
Zodiak Effects Addresses	89
Software Version (6C and EC) Commands	90
Effects Address	90
Model Number	90
Version Number	90
Switcher Model Features	91

Section 3 — Peripheral Bus II Protocol 93

Introduction	93
Overview	93
System Description	93
E-MEM System Interaction (Learns, Recalls)	94
Trigger Interactions	95
Command Timing and Frame Accuracy	96
Protocol Description	96
Hardware Interface	96
Command Structure	97
Controlled Device Identification	98
Commands	99
Learn Command	99
Recall Command	100
Trigger Command	101
Query Command	102
Query Response	102
Read Command	103
Write Command	104
Write Response	104

Section 4 — DPM CPL Protocol 105

Introduction	105
General Background about Switcher/DPM Integration	105
Switcher Configuration	106
General Protocol Notes	106
Message Timing	107
Init/Online	108

Source Selection	109
Near/Far and Front/Back	109
Source Hold	110
Tally	110
Activity Check	111
Running Effects.	112
Subscription.	112
Message Format	113
Message Tokens	114
Set	115
Subscribe	115
Parameter Service	116
Unsubscribe	116
Issue	117
Parameter Article	118
Send Event	119
Return Code.	120
Parameters.	121
Other Command Specifics	122
Transmission Media and Protocols	123
Examples	124
Section 5 — Router Protocol	129
Introduction.	129
Native Protocol	129
Commands	130
Section 6 — Still Store Image File Format	131
Introduction.	131
Still Store Hardware	131
Image Storage Mechanism	131
Metadata File Format Versions and Kalypso Software Releases.	133
Video Conversion.	133
Shaped Video	134
SD Image Data File Format.	136
SD Full Size File Format / Size Calculations	137
SD Fenced Image File Size Calculations	138
HD Image Data File Format	139
HD Full Size File Format / Size Calculations.	140
HD Fenced Image File Size Calculations.	141
Image Metadata File Format.	141
Line Rate.	142
Still Store Output Control	143
Output Crop and Position	144
Output Freeze Mode.	144
Mark In/Out	145
Loops.	145
Key Offset.	145
Index.	147

Tally Protocol

Introduction

This section describes the tally mechanisms available on the Kalypso and Kayenne Video Production Center and the Zodiac Digital Production Switcher systems. These systems (generically called “switcher”) are each equipped with a classic tally relay system, but also have a serial tally interface to provide more in-depth status than simple on/off air information. Technical details of the serial tally protocol used to report tally information are presented later in this section.

Kalypso System Overview

A Kalypso system provides 64 tally relays that close in response to tally statuses, and an additional 64 relays may be added as an option. The tally serial port also outputs current system tally information.

Kalypso systems are available in 4-ME and 2-ME models. The MEs are identical to one another, with one acting as a PGM PST bank. Each ME has four keyers, two background buses, two utility buses, two preview outputs, and four program outputs.

A Kalypso system has 46 programmable system outputs (Outputs 39 and 40 are not programmable). An aux bus output can be configured as a video/key pair.

The Kalypso Still Store option has two inputs and eight outputs.

Zodiac System Overview

A Zodiac system provides 32 tally relays. The tally serial port also outputs current system tally information.

Zodiac systems are available in a 3-ME model (three fully functional MEs, one acting as PGM PST) and a 2.5-ME model (2 MEs and PGM PST). Each ME has four keyers, two background buses, one utility bus, one preview output, and one program output. The PGM PST-DSK bank is equipped

with three downstream keyers (on the 3-ME model an additional four ME keyers are available, making a total of seven), and has two program and two preview outputs.

Many of the Zodiac system's 26 outputs are dedicated, and 13 aux buses are available. Zodiac system output pairing is different from that on Kalypso systems.

The Zodiac Still Store has two inputs and four outputs.

Kayenne System Overview

A Kayenne system provides 24 tallies relays per installed ME board. Smaller 4-RU Kayenne frames support up to 48 relay tallies. Larger 8-RU Kayenne frames support up to 96 relay tallies.

Kayenne systems have a modular design available in different configurations, ranging in size from 1.5ME to 4.5MEs. From a protocol perspective, a Kayenne can have up to 5MEs (which includes the ME50). Kayenne systems have 6 keyers per ME.

Kayenne systems also support an Image Store capable of 6 input and output channels, and a floating eDPM effects system.

Kayenne Source IDs are in the range of 1-146 (Kalypso was limited to 128).

The message sync algorithm employed is different from Kalypso because source IDs can have values greater than 128. The size of the contribution tally is 303 bytes (Kalypso was 223 bytes).

Tally Background Information

This subsection describes what a switcher does in terms of tally calculations, and may be useful for persons designing a tally computer that uses serial tally information.

The tally system of a production switcher provides status to devices and their operators. The most basic status is whether or not the device is on air, but other statuses can also be provided by the tally system. For example, the Kalypso system has four program outputs on its PGM PST bank. The tally system is not affected when one of the additional outputs is simply a "clean feed" version of the main program output. However, customized feeds which depart from the program output are possible. These customized outputs may include sources which are not present on the main program output. For example, the switcher may automatically substitute a "private" camera for a wide shot. The tally system provides ways to properly tally these special situations.

Tally Types

Tally requirements can go beyond simple on/off air indication for each source. This section describes the types of tally information which can be provided via the switcher tally relays or the serial protocol.

The switcher performs five tally calculations: on-air tally, plus four other customer specified calculations. Each calculation may be customized by selecting the tally type and which buses contribute to the tally.

There are three basic types of tally calculations and two specific derivatives for the “output tally” type.

Output Tally

Output tally indicates which sources contribute to a specific output or group of outputs (maximum of four outputs). On-air and look-ahead tally are specific cases of output tally.

Output tally needs a starting point for tally calculations. The customer can select one or more outputs which will be included in the tally calculation. Any sources which appear on those outputs will be tallied.

On-air Tally

On-air tally indicates whether a source is on or off air. This information is most often used to provide camera tally so the camera operator and the talent know when the camera is included on the program output(s) of the production switcher. It is a special case of output tally. It is provided as a separate type to ease tally configuration.

The switcher has more than one output which may be on air (e.g., all four program outputs of the PGM PST bank may have different source contributions). On air tally includes sources which contribute to any of these outputs. On-air tally had no configuration. It tallies any source which contributes to any of the four PGM PST program outputs.

Look-ahead Tally

Look-ahead tally is similar to On-air tally but is based on the selection made on the switcher’s main PRESET bus. It is intended to indicate which sources will be on air if a program-preset transition is performed. Look-ahead tally is also a special case of output tally. It is provided as a separate type to ease tally configuration. Look-ahead tally has no configuration. It tallies any source which contributes to the look-ahead state for any of the four program outputs of the PGM PST bank.

Iso Tally

Iso tally indicates which sources contribute to a specific point in the switcher video path whether or not those sources contribute to the program output. For example, an ME iso tally indicates what sources contribute to one or more of an ME's program outputs. Iso tally is similar to output tally except that the tally calculation is specified in terms of the ME outputs not on physical outputs of the switcher. In fact, the ME outputs do not have to be mapped to physical outputs in order to generate ME Iso tally.

Selection Tally

Selection tally indicates what source is selected on a particular bus or buses. Selection tally does not follow reentries or factor in bus visibility. The buses need not be on air for the sources to contribute to this information.

Selection tally requires specifying which buses of the switcher should be included in the tally calculation.

Tally Calculation Basics

Tally status provides information about switcher sources based on the video composite at some point in the video path. This section describes the mechanisms used within the switcher to calculate tally relay closures. The information is also useful when interpreting the "contribution tally" data described later.

Many sources may be combined at each ME of the switcher and ME banks can be reentered. In order to calculate tally status, the video and key paths must be traced backwards from the video output through all MEs, keyers and external processing equipment (e.g., a DVE) to the primary sources.

At each stage in the video path, tally calculations must take into consideration whether or not a particular bus contributes to the composite and what source is selected on the bus. If the selection is a primary source, the end of that branch has been found. However, if the selection is a reentry (either another ME or internal (e.g., frame store) or external processing (DVE), the trace must continue in order to find sources which contribute to that processing block's output.

Calculations must include both video and key paths through the switcher.

Keyer Modifications

Keyers modify the video and key signals and affect tally calculations.

The keyer can either use the incoming video or replace it with a matte. In addition, the key signal can be derived either from the incoming video,

incoming key signal or a wipe pattern generator. The video input contributes to the key if it is used as either the fill or as the key source. The key input contributes to the key if it is used as the key source.

An opacity setting of 0% negates all contributions of the keyer.

The processed cut signal or the masking signals may also change video contribution. If the result of key processing produces a completely transparent (i.e., cut signal is 0 for the entire raster), the video signal is not contributing to the key output. This calculation is normally not considered since it requires real time sampling of the cut signal and existing key processor ASICs do not include the calculations.

All of these factors are combined into a single flag within the contribution tally packet.

Transition Mixer Modifications

The transition mixers modify the contribution of all inputs to an ME bank. If the mixer makes an input transparent, there is no contribution to the mixer output for that input.

Separate contribution flags are provided for each path into a mixer. Each mixer (the Kalypso system has four per ME, for example) has its own set of contribution flags in the contribution tally packet.

Internal and External Processing Loops

Images processed by DVEs, frame stores or other devices must also receive accurate tally. Since these devices have the ability to make their incoming video invisible, their current state must be included in the switcher tally calculations. For example, if the DVE positions an image completely off screen or the frame store is frozen, the on-air tally for that source should go away (assuming it isn't on air via some other path).

Tally System

There are many inputs, outputs and paths through a production switcher. The tally system must be capable of determining which sources contribute to a specified output regardless of contributions to other outputs. In essence, there may be multiple tally paths through the switcher just as there are multiple video paths. It is essential to maintain tally independence back through each video processing path of the switcher and external gear (e.g., DVE).

For this reason, each device which processes video must build a “contribution map” which indicates which of its inputs are visible on or contributes to each output. Because the video path through the switcher is known, the tally system can then trace each output back through all paths and find all contributing sources. The set of contributing sources for each output is likely to be different, but can be tallied independently.

With this mechanism, most tally modes are a degenerate case of Output Tally. On-air tally is Output Tally for the PGM PST Program output. Look-ahead tally is Output Tally for the PGM PST Preview output.

If more than one program output exists, for example, the four programmable clean feed outputs of the PGM PST bank, contributions can be combined after tracing each path from output back to all inputs.

Names

All tally information is number based to keep messages short and minimize communication time. Source names are provided through a separate mechanism because they change less frequently and require a higher bandwidth to transfer.

Source names name the source IDs including reentries.

Tally Relay Outputs

The Kalypso system provides 64 tally outputs as part of the standard system. A second card can be added for a total of 128 tally relays within the frame. The Zodiak system has 32 tally relays with no additional cards available.

On a Kalypso or Zodiak system, tally relays can be configured for different tally calculations in groups of 32. For example, on a Kalypso system one group can provide on-air tally while another group can provide bus iso tally. Each tally relay within the group can be associated with one of the 128 sources. In the previous example, on-air tally can be set up for any 32 of the 128 Kalypso sources using the first group. The second group could provide bus iso tally for the same set of 32 sources or a completely different set. In many situations, 32 on-air closures will be sufficient. When this is not the case, additional groups can also be configured for on-air tally calculations and mapped to different sources. By using all four tally groups for on-air tally, all 128 Kalypso sources can be tallied.

Note At the time of publication, switcher tally relays were hard-coded to correspond one-to-one to system sources. The tally relay for any source visible on any of the outputs of the PGM PST bank will close. Programmable tally relays, as referenced here, are under development. Check the documentation of your current switcher software version to determine what tally relay programming capabilities are available.

Serial Tally

The switcher provides a serial tally interface that provides the necessary information to trace any tally path within the switcher.

The serial protocol is based on RS-422 asynchronous serial communications at 76.8 kb, 8 data bits, 1 stop bit, and no parity as a default. Other baud rates (9600, 19,200, 38,400, 57,600, and 115,200) and odd or even parity selections can be selected for contribution tally but the lower rates reduce throughput and hence latency in reporting tally. 76.8 kb/field with no parity yields about 128 characters per field and was assumed for all transmission times unless otherwise noted. At 9600 baud and even or odd parity, the link only supports 14 characters per field, making the link virtually useless for real time tally (contribution tally would require 16 fields to transmit). See [Table 10 on page 26](#) for specific timing information.

The serial tally interface uses binary protocols with easily identifiable message boundaries. All communication is from the switcher frame to the external device. Multiple listeners can be connected to this port. External devices should not transmit information on either pair of the link. The switcher does not listen to this port, it only speaks.

The contribution tally information is currently output to port 5 on the Kalypso system, and port 2 on the Zodiak system. The ability to configure switcher serial ports for different purposes is currently under development.

Changing Tally Port Serial Settings

At the time of publication, the switcher does not have a menu to change serial port settings. The contribution tally baud rate can be changed by editing the text file "mfPorts.cfg" located in the root of the frame's hard disk. The file can be FTPed to a workstation, edited with any text editor and FTPed back. The frame can then be reset to activate the new baud rate. If the file is not recognized during frame boot up, a default will be created replacing an existing one. The default uses 76.8 Kb.

The following sample indicates what the file should look like for 115.2 Kb on a Kalypso system.

```
PORT5:115200,N,8,1 ; (Port:X:BaudRate,Parity,DataBits,StopBits)
```

PORT5: identifies which port will be affected by the remaining information on the line. Currently only the tally contribution port (port 5 on Kalypso, port 2 on Zodiak) can be set using this mechanism.

Respectively, the remaining comma separated information is:

Baud Rate: 9600, 19200, 38400, 57600, 76800 or 115200

Parity: N for none, O for odd or E

Number of Data Bits: must be 8

Number of Stop Bits: 1 (typical) or 2.

The semicolon indicates a comment separator causing the rest of the line to be ignored by the parser.

Kalypso & Zodiak Contribution Tally Protocol

Contribution tally makes it possible to determine which sources and processing blocks (an ME or DVE) contribute to the image at any point in the video path. On-air tally, for example identifies which sources contribute in some way to the main program output of the switcher. ME 2 iso tally starts at ME 2's main program output.

Refer to *Tally Calculation Basics* on [page 12](#) and *Tally System* on [page 13](#) for more details.

Contribution tally consists of approximately 220 bytes of information. Because bandwidth is limited to approximately 128 bytes per field at 76.8 kb, this information is broken into smaller messages. This allows sending changes at a higher priority than unchanging information. Even with prioritized updates, there is no guarantee that all changes will be communicated within one field. See [Table 10 on page 26](#) for specific timing information.

Contribution messages indicate which sources are selected on buses feeding the processing block (or output) and which of those inputs contribute to a processing block's output. In the cases where a processing block has several outputs, a separate contribution map is included for each output. Contribution tally for external processing blocks such as a DVE also lists the source IDs for this block's output(s) so that reentry paths through external devices can be identified and followed.

Contribution message codes start at 0x8 with an instance ID greater than or equal to one. This insures that command character value is always above 128 so it is recognized as a command code. Command codes 0xE and 0xF are not used but are reserved.

Contribution information for each processing block is sent as a separate message. Information for switcher outputs is sent in six messages of eight outputs each. All portions of the system are included in the contribution dump even if they are inactive. Inactive blocks will contain data which indicates inactive status for all inputs to the processing block. In addition, that block's outputs will never appear as selections on any bus of the switcher.

Source IDs

Valid source IDs used in this protocol range from 1-128. Source IDs in the range of 93 - 128 originate within the switcher and never change. On Kalypso systems, source IDs in the range of 1 - 92 represent external sources, including any external processing blocks. Zodiac systems have the same range of external source IDs, but have fewer external physical inputs.

A source value of 0 indicates the input or bus is not in use, or to indicate that an external DVE is being used in an effects send loop. In this case, the DVE's contribution information is included in the ME's contribution information and additional tally calculations are not necessary. No further tally can be calculated for source IDs of zero, and it should not be counted as a primary input.

Tally is source based. Which physical inputs are used is not factored into tally information.

Table 1 summarizes source IDs for ME reentries and internal sources. Note that some sources are not available on some switcher models.

Table 1. Source IDs

Source ID	Reentry	Source ID	Reentry
93	ME 1 PGM A	94 ^a	ME 1 PGM B
95 ^a	ME 1 PGM C	96 ^a	ME 1 PGM D
97	ME 1 PVW A	98 ^a	ME 1 PVW 2
99 ^b	ME 2 PGM A	100 ^{a b}	ME 2 PGM B
101 ^{a b}	ME 2 PGM C	102 ^{a b}	ME 2 PGM D
103 ^b	ME 2 PVW A	104 ^{a b}	ME 2 PVW 2
105 ^{b c}	ME 3 PGM A	106 ^{a b}	ME 3 PGM B
107 ^{a b}	ME 3 PGM C	108 ^{a b}	ME 3 PGM D
109 ^{b c}	ME 3 PVW A	110 ^{a b}	ME 3 PVW 2
111	Pgm-Pst PGM A	112	Pgm-Pst PGM B
113 ^a	Pgm-Pst PGM C	114 ^a	Pgm-Pst PGM D
115	Pgm-Pst PVW A	116	Pgm-Pst PVW 2
117 ^a	Test Signal	118	Black
119	Background 1	120	Background 2
121	Still Store 1	122	Still Store 2
123	Still Store 3	124	Still Store 4
125 ^a	Still Store 5	126 ^a	Still Store 6
127 ^a	Still Store 7	128 ^a	Still Store 8

^a Zodiac Systems do not have these sources.

^b 2-ME Kalypso Systems do not have these sources.

^c 2.5-ME Zodiac Systems do not have these sources.

ME Contribution Information

Kalypso systems have up to 4 MEs. Each ME has 12 input buses (4 video/key pairs, 2 backgrounds and 2 utility buses), 4 program outputs (PGM A – PGM D) and two preview outputs. PVW A is always associated with PGM A. PVW 2 is selectable as the preview for PGM B, C or D. Preview output contribution depends on the preview mode selected and active keyer “hold to previews” or “show keys”. Contribution information is provided for all six real outputs and for the look-ahead state of the four program outputs. “LAP” contribution always indicates what the program contribution information would be following a main transition.

Zodiak system MEs have only one utility bus, and only one program and one preview output. Contribution information for these will be generated, but none will be generated for utility 2 which is not present. Contribution information for missing inputs and outputs will always be false.

On a Zodiak 2.5-ME system, PGM PST-DSK has three downstream keyers and no utility buses, and two program and two preview outputs. Zodiak PGM PST-DSK is reported as Pgm-Pst. It will only report keys 1-3 as contributing to the composite, and only show PGM A and B outputs active. ME 3 is not active.

On a Zodiak 3-ME system, PGM PST feeds the downstream keyer and is reported as ME 3. The DSK is reported as Pgm-Pst, and has three downstream keyers, no utility buses, and no B bus. The DSK has two program and two preview outputs.

The Keyer buses “in use” flags indicate whether or not the cut and fill would be visible if the keyer was contributing to an output. The Contribution flags indicate whether or not the keyer contributes to the output. These flags need to be combined (ANDed) to determine if a keyer cut or fill bus is actually contributing to the output. This was done to limit size of the message without eliminating information.

The source information indicates which source is feeding the ME inputs. These may be primary sources or reentries. Sources and in use flags apply to all outputs from an ME, but contribution information may be different for each output. Refer to [Table 2 on page 19](#).

Table 2. ME Contribution

b7	b6	b5	b4	b3	b2	b1	b0	Notes				
ME Contribution (0x8)				ME ID				ME IDs 1 – ME 1 2 – ME 2 3 – ME 3 4 – Pgm-Pst	Source IDs			
ME Contribution (0x8)				ME ID					ME 1	ME 2	ME 3	Pgm-Pst
Key 1 Fill Source (1 – 128)												
Key 4 Cut	Key 4 Fill	Key 3 Cut	Key 3 Fill	Key 2 Cut	Key 2 Fill	Key 1 Cut	Key 1 Fill	Keyer buses in use				
Key 1 Cut Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PGM A Contribution	93	99	105	111
Key 2 Fill Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PGM B Contribution	94	100	106	112
Key 2 Cut Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PGM C Contribution	95	101	107	113
Key 3 Fill Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PGM D Contribution	96	102	108	114
Key 3 Cut Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	LAP A Contribution				
Key 4 Fill Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	LAP B Contribution				
Key 4 Cut Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	LAP C Contribution				
A Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	LAP D Contribution				
B Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PVW A Contribution	97	103	109	115
Utility 1 Source (1 – 128)												
Key 4	Key 3	Key 2	Key 1	Util 2	Util 1	B	A	PVW 2 Contribution	98	104	110	116
Utility 2 Source (1 – 128)												

External Processing Contribution Information

External video processing blocks such as DVEs must be included for accurate tally calculations. This protocol supports up to two external processing blocks, each with up to four input pairs and six outputs.

External processing blocks use primary inputs for reentry paths. The output source IDs identify these paths. These source IDs should be treated just like an ME reentry. Output contribution maps and reentry source IDs are provided for the 6 outputs.

The input mode flags indicate whether the B side of an input pair is an independent video signal or linked to the A side as the input’s key. Refer to [Table 3](#).

Table 3. External Processing Contribution

b7	b6	b5	b4	b3	b2	b1	b0	Notes
External Proc Contribution (0x9)				Ext Proc ID				Ext Proc ID 1 – DPM 1 2 – DPM 2
External Proc Contribution (0x9)				Ext Proc ID				
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 1 Contribution
Output 1 Source (1 – 128)								Output 1 reentry ID
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 2 Contribution
Output 2 Source (1 – 128)								Output 2 reentry ID
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 3 Contribution
Output 3 Source (1 – 128)								Output 3 reentry ID
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 4 Contribution
Output 4 Source (1 – 128)								Output 4 reentry ID
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 5 Contribution
Output 5 Source (1 – 128)								Output 5 reentry ID
Input 4B	Input 4A	Input 3B	Input 3A	Input 2B	Input 2A	Input 1B	Input 1A	Output 6 Contribution
Output 6 Source (1 – 128)								Output 6 reentry ID
0				Input 4B	Input 3B	Input 2B	Input 1B	Input Mode 0 – Video 1 – Key
Input 1A Source (1 – 128)								
Input 1B Source (1 – 128)								
Input 2A Source (1 – 128)								
Input 2B Source (1 – 128)								
Input 3A Source (1 – 128)								
Input 3B Source (1 – 128)								
Input 4A Source (1 – 128)								
Input 4B Source (1 – 128)								

Still Store Contribution Information

The Kalypso internal still store has 8 outputs. Outputs can be configured in pairs as two independent outputs or as a video-key pair. The Output Mode flags identify the configuration for each output pair.

The Zodiac internal still store has only four outputs, so still store contribution information will only be valid for outputs 1-4. Contribution information for outputs 5-8 will always be false.

The Kalypso and Zodiak internal still store has two inputs. They can be configured as two independent video inputs or as a video-key pair. The Input Mode flag indicates which mode the inputs are operating in.

The Record flags indicate whether or not the input is in use. If an input is grabbing a frame or recording an animation, the bit will be 1. An active record flag is essentially a beginning point for tally regardless of still store output usage since the indicated source is being recorded on the still store's hard disk. Refer to [Table 4](#).

Table 4. Still Store Contribution

b7	b6	b5	b4	b3	b2	b1	b0	Notes
Still Store Contribution (0xA)				1				Only one instance of Still Store contribution
Still Store Contribution (0xA)				1				
Input 1 Source (1 – 128)								
Input 2 Source (1 – 128)								
0	Output 7/8 Mode	Output 5/6 Mode	Output 3/4 Mode	Output 1/2 Mode	Input Mode	Input 2 Rec	Input 1 Rec	Mode 0 – Video-video 1 – Video-key

Output Contribution Information

Kalypso Systems

Kalypso has 46 programmable outputs (Outputs 39 and 40 are dedicated to Black and Test and report these source IDs). An output can be programmed as either an aux bus or as a dedicated output for an ME, etc. From a tally standpoint, the only difference is what source is selected on the output. A dedicated output won't change sources while an aux bus might. That is, the tally protocol won't indicate what the output is being used for (dedicated output or aux bus) and the receiver of the information shouldn't care. It may be necessary to identify which physical outputs are starting points for tally calculations.

Output status is communicated in 6 messages of 8 outputs each. For example, status for output 10 would be sent as output 2 in the second block.

The on-air flags indicate whether or not the switcher considers the output to be on air. This information can be hard coded, or may be based on status from some external device. For example, the Pgm-Pst PGM A output is the main program output and is normally assumed to be on air. The feed to an iso recorder may utilize a GPI input (running status via the switcher-VTR control interface) so that output is on air only when the VTR is in record. Tally calculations may choose to ignore these flags.

Outputs used as aux buses may be paired in order to deliver a video-key pair to an external device. The Output Mode flags indicate whether the two

buses of a pair are independent (video-video) outputs or are being used as a video-key pair. Refer to [Table 5](#).

Table 5. Output Contribution

b7	b6	b5	b4	b3	b2	b1	b0	Notes
Outputs Status (0xB)				Block ID				Block ID 1 – Outputs 1 - 8 2 – Outputs 9 - 16 3 – Outputs 17 - 24 4 – Outputs 25 - 32 5 – Outputs 33 - 40 ^a 6 – Outputs 41 - 48 ^b
Outputs Status (0xB)				Block ID				
Output 1 Source (1 – 128)								
0	0	0	0	Out 7/8	Out 5/6	Out 3/4	Out 1/2	Mode 0 – Video-video 1 – Video-key
Output 2 Source (1 – 128)								
Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	On-air
Output 3 Source (1 – 128)								
Output 4 Source (1 – 128)								
Output 5 Source (1 – 128)								
Output 6 Source (1 – 128)								
Output 7 Source (1 – 128)								
Output 8 Source (1 – 128)								

^a Outputs 39 and 40 are not programmable and are always Black and Test, respectively.

^b Outputs 41 through 48 are the effects send outputs.

Zodiak Systems

Many Zodiak outputs are hard wired to ME outputs. These outputs will not be reported in tally contribution information. The remaining outputs are programmable as aux buses (single buses or aux bus pairs). Zodiak output pairing is different from Kalypso. Note that aux bus numbers may not correspond to aux output numbers since this assignment is configurable. Output messages are organized to put Zodiak's effects send buses on the same output IDs as used for Kalypso effects send buses. Zodiak aux outputs 6-13 (Zodiak's effects send outputs) are reported as outputs 41-48 (Kalypso's effects send outputs). Zodiak's switched preview is also reported as Output 6. ([Table 6](#)).

Table 6. Zodiac Output Assignments

Output	Signal
1	Aux Output 1 (A side if paired with aux output 2)
2	Aux Output 2 (B side if paired with aux output 1)
3	Aux Output 3 (A side if paired with aux output 4)
4	Aux Output 4 (B side if paired with aux output 3)
5	Aux Output 5 (can't be paired)
6	Switched Preview
7-8	unused (reported as source ID 0)
9-40	unused, not reported
41	Aux Output 6 (effects send - A side if paired with aux output 7)
42	Aux Output 7 (effects send - B side if paired with aux output 6)
43	Aux Output 8 (effects send - A side if paired with aux output 9)
44	Aux Output 9 (effects send - B side if paired with aux output 8)
45	Aux Output 10 (effects send - A side if paired with aux output 11)
46	Aux Output 11 (effects send - B side if paired with aux output 10)
47	Aux Output 12 (effects send - A side if paired with aux output 13)
48	Aux Output 13 (effects send - B side if paired with aux output 12)

Source Names

Switcher source names are not really part of tally, but often provide valuable information to devices which utilize switcher tally information. Source names are limited to twelve characters and each is sent in its own message along with the source's ID. Source names are not necessarily unique. That is, two sources could have the same name.

The switcher supports an optional nickname for each source as well as the full source name. Nicknames are a shortened version of the full source name intended for displays with limited space. These are blank if undefined in which case the full source name should be used.

The switcher supports the notion of aliasing source names, i.e., replacing generic names for names more specific to the intended usage of the source. When aliases are activated in suite preferences, the tally system reports aliases. Otherwise, the contribution tally reports engineering source names. The name set ID indicates which name set is being reported.

Characters are limited to 7-bit ASCII. Names shorter than 12 characters are left justified and null (0) filled. The message format is indicated in [Table 7](#).

Table 7. Source Names

b7	b6	b5	b4	b3	b2	b1	b0	Notes
Source Name (0xC)				Nameset ID				Nameset ID 1 – Name & Nickname 2 – Alias
Source Name (0xC)				Nameset ID				
Source ID (1 – 128)								
0	Name Character 1							Left-most character
0	Name Character 2							
0	Name Character 3							
0	Name Character 4							
0	Name Character 5							
0	Name Character 6							
0	Name Character 7							
0	Name Character 8							
0	Name Character 9							
0	Name Character 10							
0	Name Character 11							
0	Name Character 12							Right-most character
0	Nickname Character 1							Left-most character
0	Nickname Character 2							
0	Nickname Character 3							
0	Nickname Character 4							
0	Nickname Character 5							
0	Nickname Character 6							Right-most character

Update

Update messages inform the receiver of critical state changes. Initialize indicates the start of switcher tally processing (the beginning of life). Data Consistent indicates that all tally packets are up to date. Since updates for one change may span several packets, to avoid tallying sources improperly, tally calculations should only be performed when data is consistent (all updates have been sent). See [Table 8](#). Refer to [Command Codes and Instance Summary](#) for the use of these messages.

Table 8. Update Messages

b7	b6	b5	b4	b3	b2	b1	b0	Notes
Update (0xD)				Update ID				Update ID 1 – Initialize 2 – Data Consistent
Update (0xD)				Update ID				

Command Codes and Instance Summary

Refer to [Table 9](#) for a list of the valid command codes and valid instance IDs for each command, and message lengths.

Table 9. Command Code and Instance

Command Code	Instances	Message Length	
		Data Only	With Command
ME Contribution (0x8)	1 – ME 1 2 – ME 2 3 – ME 3 4 – Pgm-Pst	23	25
External Proc Contribution (0x9)	1 – DPM 1 2 – DPM 2	21	23
Still Store Contribution (0xA)	n/a	3	5
Outputs Status (0xB)	1 – Outputs 1 - 8 2 – Outputs 9 - 16 3 – Outputs 17- 24 4 – Outputs 25 - 32 5 – Outputs 33 - 40 6 – Outputs 41 - 48	10	12
Source Name (0xC)	1 = Name & Nickname 2 = Alias	19	21
Update (0xD)	1 – Initialize 2 – Data Consistent	0	2
unused (0xE)			
unused (0xF)			

Message Structure

Messages start with a message code. Message codes are repeated twice in order to provide positive identification of the message start. Data bytes could match a valid message code, however the messages have been laid out such that no two consecutive data bytes have values above 128. Since all message code characters are above 128, it is not possible for message data to duplicate the repeated message code.

Communication Specifics

When the switcher begins sending tally information, it will issue an Initialize command, then send all tally packets as fast as possible. After all packets have been sent a Data Consistent message is sent.

During idle times (no changes), one background contribution update will be sent at the frequency specified in [Table 10](#) followed by a Data Consistent message. All instances of all message codes are always sent regardless of which portions of the switcher are currently active.

The source name set is sent one name at a time. All 128 source names are sent in order and the process repeats.

Background contribution updates and the source name messages are spaced out and interleaved as shown in [Table 10](#). Remaining serial bandwidth is used for contribution change updates, if any.

When changes occur, many messages may be required to communicate the change. At the end of the stream of packets, a Data Consistent message will be sent. The switcher will insure that data is inconsistent for no longer than indicated in [Table 10](#).

Table 10. Communication Specifics

Category	Field Rate	Baud Rate					
		9,600	19,200	38,400	57,600	76,800	115,200
Cycle period (fields)	60 Hz	20	10	5	4	3	2
Background contribution refresh (fields)		260	130	65	52	39	26
Background contribution refresh (sec)		4.329	2.165	1.082	0.866	0.649	0.433
Source name refresh (fields)		853	427	213	171	128	85
Source name refresh (sec)		14.208	7.104	3.552	2.842	2.131	1.421
Max data inconsistent time (sec)		0.333	0.167	0.083	0.067	0.050	0.033
Cycle period (fields)	50 Hz	16	8	4	3	2	2
Background contribution refresh (fields)		208	104	52	39	26	26
Background contribution refresh (sec)		4.160	2.080	1.040	0.780	0.520	0.520
Source name refresh (fields)		683	341	171	128	85	85
Source name refresh (sec)		13.653	6.827	3.413	2.560	1.707	1.707
Max data inconsistent time (sec)		0.320	0.160	0.080	0.060	0.040	0.040

The cycle period is the time (in fields) it takes to send one background contribution update and three source name messages. Remaining time within a cycle is used to send contribution changes. If no changes need to be sent, the link goes idle during a cycle.

By design, the cycle period is also the time it takes to send all contribution messages. This is the maximum time contribution information will be inconsistent. Data is inconsistent when tally changes occur and more than one message must be sent to communicate all changes.

Background contribution refresh is the time it takes to completely send all contribution messages as background updates. This tells the tally system designer how long it will take for their system to sync up to the switcher should they connect after the switcher is up and idle.

Source name refresh is the time it takes to send all source name messages.

Message Parsing and Processing

Message code recognition and parsing can use the following algorithm:

```

messageCode = INVALID
lastChar = 0
bytesReceived = 0
dataConsistent = FALSE

repeat forever
currentChar = next byte on input stream

// message header - set up for new message
if (currentChar > 128 && currentChar == lastChar)
messageCode = currentChar >> 4 // message code is 4 MSBs
instance = currentChar & 0xF // instance is 4 LSBs
bytesReceived = 0

if (messageCode == UPDATE)
switch (instance) // instance is really update type
case INITIALIZE:
invalidateAllData ()
dataConsistent = FALSE
break
case DATA_CONSISTENT:
if ( allDataValid ())
dataConsistent = TRUE
processTally ()
break
end switch

// prevent tally processing while receiving tally updates
// but source name updates shouldn't prevent tally calcs
else if (messageCode < SOURCE_NAME)
dataConsistent = FALSE

// assemble message starting with message code and instance ID
if (messageCode != INVALID)
messageBuffer[bytesReceived++] = currentChar
if (bytesReceived == messageLength (messageCode) + 1)
copyTally (messageBuffer, messageCode, instance)
messageCode = INVALID // end of message
else
// waiting for valid message header - discard char
lastChar = currentChar
end repeat

onSerialTimeout ((76,800 / baudRate) * 100mS)
invalidateAllData ()
dataConsistent = FALSE

```

This routine looks for two consecutive characters greater than 128. The message type is extracted from the 4 most significant bits of this character. An instance ID is extracted from the 4 least significant bits of this same character. Instance IDs indicate specific information for identical objects (e.g., MEs). The meaning of instance (if any) for a command is indicated in the command descriptions above.

Each message type has a known message length. `messageLength` returns the number of data characters in the message. Message data lengths are specified in the command descriptions above.

Characters are assembled in a buffer including the message code and instance ID. When all characters have been received, `copyTally` is called to move the completed message out of the message buffer to somewhere it can contribute to tally calculations.

This algorithm assumes the program keeps track of valid receipt of each and every contribution tally data block (every instance of all messages). It is not appropriate to process tally data if all data blocks are not up to date. When a data block is received, it is marked as valid (possibly by `copyTally`). The only thing which invalidates any data block is `invalidateAllData` which invalidates all data blocks. This only occurs on loss of communication or when the switcher sends the Initialize message.

`allDataValid` is routine which checks all the data block flags to make sure all the data is valid. If it is, tally calculations can be performed. If not, they must wait until all blocks have been received. Since the switcher sends a Data Consistent message after every background update and after a stream of changes, this is the appropriate time to check the data flags and determine if processing should be permitted.

`processTally` is routine which performs the desired tally calculations. An alternative is to perform tally calculations in a different part of the program but only if `dataConsistent` is true. Specific tally calculations depend on the application and are beyond the scope of this document. In practice, you wouldn't need both `dataConsistent` and `processTally`.

Failure of the communications link for even a short period of time may allow tally updates to be missed and the receiver's view of the data to be out of date. For this reason, some mechanism (represented by `onSerialTimeout` ()) should be implemented which detects lack of communication activity. If this occurs, tally calculations should be suspended and all data blocks should be marked as invalid. Tally calculations can resume after all data blocks have been refreshed and a Data Consistent message has been received.

At least one background update will be sent on a regular basis. The frequency of this update is a function of the link baud rate as shown in the table above. The inactivity timeouts listed in the table are based on 120% of the update sequence length for 50 Hz operation (since it has a longer field time (20 mS)).

Kayenne Contribution Tally Protocol

This section describes the Contribution Tally Protocol that applies to Kayenne switchers. This protocol has a modified form of the earlier Kalypso and Zodiac switcher protocols.

Contribution tally makes it possible to determine which sources contribute to the image at any point in the video path. On-air tally, for example, identifies which sources contribute in some way to the main program output of the switcher.

Contribution messages indicate which sources are selected on buses feeding each processing block (ME, eDPM, etc.) and which of those inputs contribute to each output of a processing block. In the case where a processing block has several outputs, a separate contribution map is included for each output.

Contribution information for each processing block is sent as a separate message. All portions of the system are included in the contribution dump even if they are inactive. Active blocks will be updated whenever their tally information changes.

Message Structure and Summary

Messages start with two message code bytes followed by the message data.

- The first message code byte has 0xF in the upper nibble and a command code in the lower nibble.
- The second message code byte has 0xF in the upper nibble and an instance number in the lower nibble.

The previous Kalypso protocol used identical two message codes in sequence to indicate start of message. This algorithm depended on the fact that message codes were always greater than 128 and source ID's were 128 or less so that two message codes in sequence was unique. This new method allows for source ID's up to and including 240 (0xF0).

Message code bytes always have their most significant 4 bits set on. The command code is in the least significant 4 bits of the first message code byte. The instance ID is in the least significant 4 bits of the second message code byte. The instance ID is 1 based and is never 0.

The new sync algorithm is to detect two consecutive bytes with 0xF in the upper nibble. This allows message data to use all eight bits. To make the sync pattern unique, messages have been organized so that bytes containing contribution map data are always separated with bytes containing source ID data. This makes detecting the beginning of messages possible and preserves efficient use of message data bytes for encoding tally information.

All messages are of fixed length. Each message type has its own characteristic length (see Table 11). Note that a complete contribution tally now consists of 303 bytes instead of 223 for Kalypso. Because the bandwidth of a serial connection is limited to approximately 128 bytes per field, the tally information is broken into smaller messages summarized in the table below. Even with prioritized updates, there is no guarantee that all changes will be communicated within one field.

Table 11. Kayenne Contribution Tally Messages

Command Code	Instance Number	Message length per instance	Message length all instances	Running Subtotal
0x8 (ME contribution)	1 = ME 1	41	205	205
	2 = ME 2			
	3 = ME 3			
	4 = ME 4			
	5 = ME 5			
0x9 (DPM contribution)	1 = eDPM	14	14	219
0xA (SS contribution)	1 = Image Store	10	10	229
0xB (Output status)	1 = Outputs 1-8	12	72	301
	2 = Outputs 9-16			
	3 = Outputs 17-24			
	4 = Outputs 25-32			
	5 = Outputs 33-40			
	6 = Outputs 41-48			
0xC (Source Name)	1 = Name & Nickname (1-146)	21	N/A	N/A
0xD (Update)	1 = Initialize Kalypso protocol	2	2	303
	2 = Data Consistent			
	3 = Initialize Kayenne protocol			
0xE (unused)				
0xF (unused)				

The message structure for Kayenne differs from the Kalypso/Zodiak protocol in the following ways:

- Different message sync mechanism.
- Different length for most messages with corresponding different length for total contribution tally information.

Source IDs

Valid source IDs used in this protocol for Kayenne range from 1-146. A source value of 0 indicates the input or bus is not in use. Source IDs in the range 1-96 represent external sources. Source IDs in the range 97-146 originate within the switcher and are fixed.

Note Contribution tally is based on engineering sources, not logical source aliases. Which physical inputs are used is not factored into tally information.

Table 12. Kayenne Source IDs

Src ID	Reentry	Src ID	Reentry	Src ID	Reentry
97	ME 1 PGM A	98	ME 1 PGM B	99	ME 1 PGM C
100	ME 1 PGM D	101	ME 1 PVW A	102	ME 1 PVW 2
103	ME 2 PGM A	104	ME 2 PGM B	105	ME 2 PGM C
106	ME 2 PGM D	107	ME 2 PVW A	108	ME 2 PVW 2
109	ME 3 PGM A	110	ME 3 PGM B	111	ME 3 PGM C
112	ME 3 PGM D	113	ME 3 PVW A	114	ME 3 PVW 2
115	ME 4 PGM A	116	ME 4 PGM B	117	ME 4 PGM C
118	ME 4 PGM D	119	ME 4 PVW A	120	ME 4 PVW 2
121	ME 50 PGM A	122	ME 50 PGM B	123	ME 50 PGM C
124	ME 50 PGM D	125	ME 50 PVW A	126	ME 50 PVW 2
127	eDPM ch 1 video	128	eDPM ch 1 key	129	eDPM ch2 video
130	eDPM ch2 key	131	eDPM ch3 video	132	eDPM ch3 key
133	eDPM ch4 video	134	eDPM ch4 key	135	Test Signal
136	Black	137	White	138	Background 1
139	Background 2	140	Black Key	141	Image Store out 1
142	Image Store out 2	143	Image Store out 3	144	Image Store out 4
145	Image Store out 5	146	Image Store out 6		

The source IDs for Kayenne differ from the Kalypso/Zodiak protocol in the following ways:

- Includes ME 50, no ME is designated PGM-PST.
- Includes eDPM outputs, which are internal sources.
- Max source ID is 146 (0x92) which overlaps with Message codes. Kalypso max source ID of 128 did not overlap with Message code which is one factor which contributed to the need for the new message synchronizing scheme. This new scheme allows for source ID's up to a max of 239 (0xEF).

ME Contribution Information

Kayenne systems have up to 5 MEs. Each ME has 16 input buses (6 video/key pairs, 2 background and 2 utility buses), 4 program outputs (A-D) and two preview outputs. PVW A is always associated with PGM A. PVW 2 is the preview of C.

Contribution information is provided for all six real outputs and for the look-ahead state of the 4 program outputs. The "LAP" contribution always indicates what the program output contribution would be following a main transition.

The keyer buses “in use” flags indicate whether or not the cut and fill would be visible if the keyer was contributing to an output. These flags need to be combined (ANDed) with the contribution bits to determine if a keyer cut or fill bus is actually contributing to the output.

The source information indicates which source is feeding the ME inputs. These may be primary sources or reentries. Sources and “in use” flags apply to all outputs of an ME, but contribution information may be different for each output.

This differs from the Kalypso/Zodiak protocol in the following ways:

- Separate bytes for fill and cut “in use” flags.
- Separate bytes for key and background bus contributions.
- Byte order is different, especially source IDs.
- No ME is designated PGM-PST (except as indicated in the Logical ME Unassigned Bytes in the ME Contribution Tally).

See [Table 13 on page 34](#).

Table 13. ME Contribution Tally

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Offset
Command Code 1 (0xF)				ME Contribution (0x8)				Message code	0
Command code 2 (0xF)				1 = ME 1 (B)				Message code	1
				2 = ME 2 (C)					
				3 = ME 3 (D)					
				4 = ME 4 (A)					
				5 = ME 50 (50)					
A Source (1-146)									2
0	0	Key 6 Fill	Key 5 Fill	Key 4 Fill	Key 3 Fill	Key 2 Fill	Key 1 Fill	Keyer Fill buses in use	3
B Source (1-146)									4
0	0	Key 6 Cut	Key 5 Cut	Key 4 Cut	Key 3 Cut	Key 2 Cut	Key 1 Cut	Keyer Cut buses in use	5
U1 Source (1-146)									6
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PGM A Key contribution	7
0	0	0	0	U2	U1	B	A	PGM A Bus contribution	8
U2 Source (1-146)									9
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PGM B Key contribution	10
0		0	0	U2	U1	B	A	PGM B Bus contribution	11
Key 1 Fill Source (1-146)									12
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PGM C Key contribution	13
0	0	0	0	U2	U1	B	A	PGM C Bus contribution	14
Key 1 Cut Source (1-146)									15
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PGM D Key contribution	16
0	0	0	0	U2	U1	B	A	PGM D Bus contribution	17
Key 2 Fill Source (1-146)									18
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	LAP A Key contribution	19
0	0	0	0	U2	U1	B	A	LAP A Bus contribution	20
Key 2 Cut Source (1-146)									21
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	LAP B Key contribution	22
0	0	0	0	U2	U1	B	A	LAP B Bus contribution	23
Key 3 Fill Source (1-146)									24
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	LAP C Key contribution	25
0	0	0	0	U2	U1	B	A	LAP C Bus contribution	26
Key 3 Cut Source (1-146)									27
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	LAP D Key contribution	28
0	0	0	0	U2	U1	B	A	LAP D Bus contribution	29
Key 4 Fill Source (1-146)									30
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PVW A Key contribution	31
0	0	0	0	U2	U1	B	A	PVW A Bus contribution	32
Key 4 Cut Source (1-146)									33
0	0	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	PVW 2 Key contribution	34
0	0	0	0	U2	U1	B	A	PVW 2 Bus contribution	35
Key 5 Fill Source (1-146)									36
Key 5 Cut Source (1-146)									37
Key 6 Fill Source (1-146)									38
Key 6 Cut Source (1-146)									39

Table 13. ME Contribution Tally - (continued)

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Offset
0	Suite ID			Logical ME ID				Logical ME Assignment	40
	1=Suite 1			0=PGM PST					
	2=Suite 2			1=ME 1					
	0=Not Assigned			2=ME 2					
				3=ME 3					
			4=ME 4						

The Logical ME Assignment byte (offset 40) represents the logical ME assigned to the physical ME identified in the message code type (offset 1). The logical assignment is by suite, hence the suite ID in bits is b4-b6 and the logical ME ID in bits is b0-b3. These may be used to display the logical ME ID associated with a specific ME contribution tally.

eDPM Contribution Information

The Kayenne includes an eDPM and so has no facility for effects send to external DVEs. Since the outputs of the eDPM are fixed internal sources (like ME reentry) and are included in the Source ID table above, they do not need to be part of this message data. Output contribution maps and input source ID's are provided for 4 channels and 4 video/key (A/B) outputs. The Output A contribution maps reflect the contribution of the A side of each input. The Output B contribution maps reflect the contribution of the B side of each input.

Table 14. eDPM Contribution Tally

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Off set
Command Code 1 (0xF)				DPM Contribution (0x9)				Message code	0
Command Code 2 (0xF)				eDPM (1)				Message code	1
Input 1A Source (1-146)									2
Input 1B Source (1-146)									3
Input 4B	Input 3B	Input 2B	Input 1B	Input 4A	Input 3A	Input 2A	Input 1A	Output 1 A Contribution	4
Input 2A Source (1-146)									5
Input 2B Source (1-146)									6
Input 4B	Input 3B	Input 2B	Input 1B	Input 4A	Input 3A	Input 2A	Input 1A	Output 2 A Contribution	7
Input 3A Source (1-146)									8
Input 3B Source (1-146)									9
Input 4B	Input 3B	Input 2B	Input 1B	Input 4A	Input 3A	Input 2A	Input 1A	Output 3 A Contribution	10
Input 4A Source (1-146)									11
Input 4B Source (1-146)									12
Input 4B	Input 3B	Input 2B	Input 1B	Input 4A	Input 3A	Input 2A	Input 1A	Output 4 A Contribution	13

This differs from the Kalypso/Zodiak protocol in the following ways:

- There is one eDPM with 4 output channels
- The A and B contributions are in separate bytes. This is the one place in the new protocol where extra bytes are used to allow the sync byte(0xFF) to be unique. It would have been possible to combine A and B input contributions into one byte but that would leave the possibility that a data byte would be 0xFF, the same as the sync byte.

Image Store Contribution Information

The Kayenne still store is configured into three channels with two inputs and two outputs each. Inputs and Outputs can be configured as two independent video signals or as video-key pairs. The mode for each channel input and output is indicated by mode flags. The record flag indicates whether or not each input is in use.

Table 15. Image Store Contribution Information

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Off set
Command Code 1 (0xF)				Image Store Contribution (0xA)				Message code	0
Command Code 2 (0xF)				Image Store (1)				Message code	1
Input 1A Source (1-146)									2
Input 1B Source (1-146)									3
Input 2A Source (1-146)									4
Input 2B Source (1-146)									5
Input 3A Source (1-146)									6
Input 3B Source (1-146)									7
0	Ch 3 Rec	Ch 2 Rec	Ch 1 Rec	0	Ch 3 mode	Ch 2 mode	Ch 1 mode	Input Mode (0 = VV, 1 = VK) and record state	8
0	0	0	0	0	Out 3 mode	Out 2 mode	Out 1 mode	Output Mode (0 = VV, 1 = VK)	9

This differs from the Kalypso/Zodiak protocol in the following ways:

- There are three channels (6 outputs) for the Image Store.
- The input modes and output modes are in separate bytes.
- The Image Store tally is channel oriented.

Output Status

Kayenne has 48 programmable outputs. An output can be programmed as either an aux bus or as a dedicated output for an ME. From a tally standpoint, only the source selected in each output is significant.

The on-air flags indicate whether the switcher considers the output to be on-air. Outputs used as aux buses may be paired in order to provide a video/key pair to an external device. The mode flags indicate whether the two buses of a pair are independent (video/video) outputs or are being used as video/key pair.

Table 16. Kayenne Output Status

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Off set
Command Code 1 (0xF)				Output Status (0xB)				Message code	0
Command Code 2 (0xF)				Block ID 1 = Outputs 1-8 Block ID 2 = Outputs 9-16 Block ID 3 = Outputs 17-24 Block ID 4 = Outputs 15-32 Block ID 5 = Outputs 33-40 Block ID 6 = Outputs 41-48				Message code	1
0	0	Mode 3/ 4	Mode 1/ 2	Out 4 onair	Out 3 onair	Out 2 onair	Out 1 onair	Mode (0 = VV, 1 = VK)	2
0	0	Mode 7/ 8	Mode 5/ 6	Out 8 onair	Out 7 onair	Out 6 onair	Out 5 onair		3
Output 1 source (1-146)									4
Output 2 source (1-146)									5
Output 3 source (1-146)									6
Output 4 source (1-146)									7
Output 5 source (1-146)									8
Output 6 source (1-146)									9
Output 7 source (1-146)									10
Output 8 source (1-146)									11

This differs from the Kalypso/Zodiak protocol in the following ways:

- The modes and source ID bytes are in different order.
- The output modes and on air state for outputs 1-4 are in separate bytes than the same data for output 5-8.

Source Names

Switcher source names are not really part of tally, but often provide valuable information to devices which utilize switcher tally information. Source names are limited to 12 characters and each is sent in its own message along with the source ID. Source names are not necessarily unique. The name pro-

vided is the engineering source name. The nickname is an engineering source nickname unique to serial tally and is limited to 6 characters.

Table 17. Kayenne Source Names

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Off set
Command Code 1 (0xF)				Source Name (0xC)				Message code	0
Command Code 2 (0xF)				1 = Name + Nickname				Message code	1
Source ID (1-146)									2
0	Name Character 1						Left-most character	3	
0	Name Character 2							4	
0	Name Character 3							5	
0	Name Character 4							6	
0	Name Character 5							7	
0	Name Character 6							8	
0	Name Character 7							9	
0	Name Character 8							10	
0	Name Character 9							11	
0	Name Character 10							12	
0	Name Character 11							13	
0	Name Character 12						Right-most character	14	
0	Nickname Character 1						Left-most character	15	
0	Nickname Character 2							16	
0	Nickname Character 3							17	
0	Nickname Character 4							18	
0	Nickname Character 5							19	
0	Nickname Character 6						Right-most Character	20	

The source name message format is identical to the Kalypso/Zodiak protocol except for the range of source ID values.

Update

The Update messages inform the receiver of critical state changes. Initialize indicates the start of switcher tally processing (the beginning of life). Data consistent indicates that all tally packets are up to date and tally calculations should be performed.

Table 18. Kayenne Update

b7	b6	b5	b4	b3	b2	b1	b0	Notes	Off set
Command Code 1 (0xF)				Update (0xD)				Message code	0
Command Code 2 (0xF)				Update ID 1 = Initialize Tally data 2 = Data Consistent				Message code	1

The use of this message is the same as the Kalypso/Zodiak protocol.

Message Parsing and Processing

Except for the difference in detecting the start of a message, the parsing and processing of messages is quite similar to that appropriate for the Kalypso/Zodiak protocol.

Generally messages are sent in the following order:

Table 19. Message Order

1	Update(3) "beginning of life" ->clearAllData
2	ME contribution (1)
3	ME contribution (2)
4	ME contribution (3)
5	ME contribution (4)
6	ME contribution (5)
7	DPM contribution (1)
8	SS contribution (1)
9	Output status (1)
10	Output status (2)
11	Output status (3)
12	Output status (4)
13	Output status (5)
14	Output status (6)
15	Source Name (1-146)
16	Update (2) "Data constant" -> processTally

Subsequent tally updates send only the messages that involve changes followed by Update (2).

Message code recognition and parsing can use the following algorithm:

```
thisByte = 0;
LOOP (forever)
{
    lastByte = thisByte;
    thisByte = read();
    if ((thisByte & lastByte & 0xF0) == 0xF0)
    {
        commandCode = lastByte & 0xF;
        instanceID = thisByte & 0xF;

        // check for update/initialize
        if (commandCode == 0xD) && (instanceID == 0x1)
            clear all data.

        // check for update/data consistent
        if (commandCode == 0xD) && (instanceID == 0x2)
            processTally()

        // read in message according to commandCode
        else
            read message data bytes for specified instance.
    }
}
```

Editor Protocol

A video editing system uses the switcher editor port to exercise real-time control over the switcher's functions. The editor is really an external computer that can input and output functions. Such functions select cross-points, perform auto transitions, or perform E-MEM (Effects Memory) register recalls, to name just a few possibilities. Refer to the appropriate switcher *User Manual* for definitions and explanations of switcher functions and features.

Kalypso/Zodiak Editor Protocol

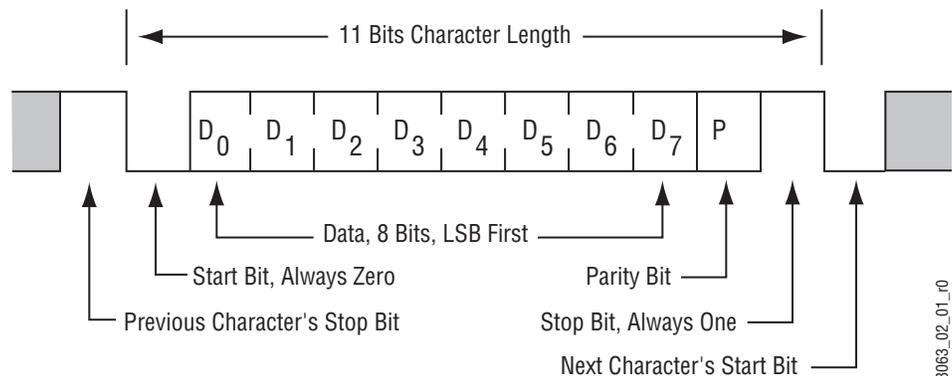
Introduction

This section describes the Editor protocol and its error handling as used by Kalypso and Zodiak systems. This Kalypso/Zodiak Editor protocol is based on the Grass Valley Group Model 4000 switcher protocol, used by the Model 4000, 3000, and 2200 Grass Valley Group switchers, but commands have been extended to support Kalypso and Zodiak features.

Serial Data Word Description

The serial data word contains eleven bits, as shown in [Figure 1](#):

Figure 1. Serial Data Word

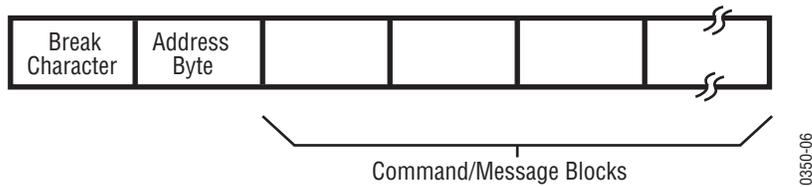


Editor Protocol Access

This Editor protocol is based upon, but not identical to, the proposed SMPTE (Society of Motion Picture and Television Engineers) Digital Control Interface, which establishes a means for the orderly flow of data over a serial bus.

The major components of the protocol are shown in Figure 2 and discussed in the following paragraphs.

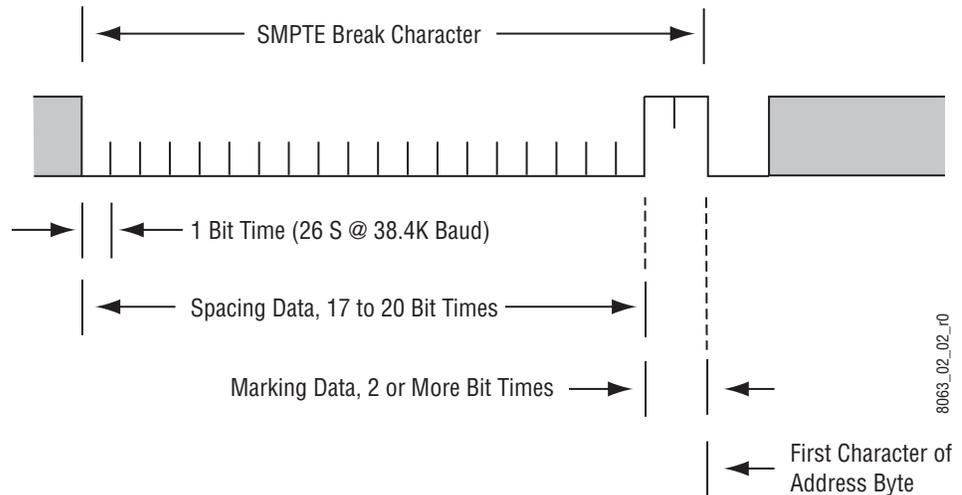
Figure 2. Components of the Editor Interface Protocol



Break Character

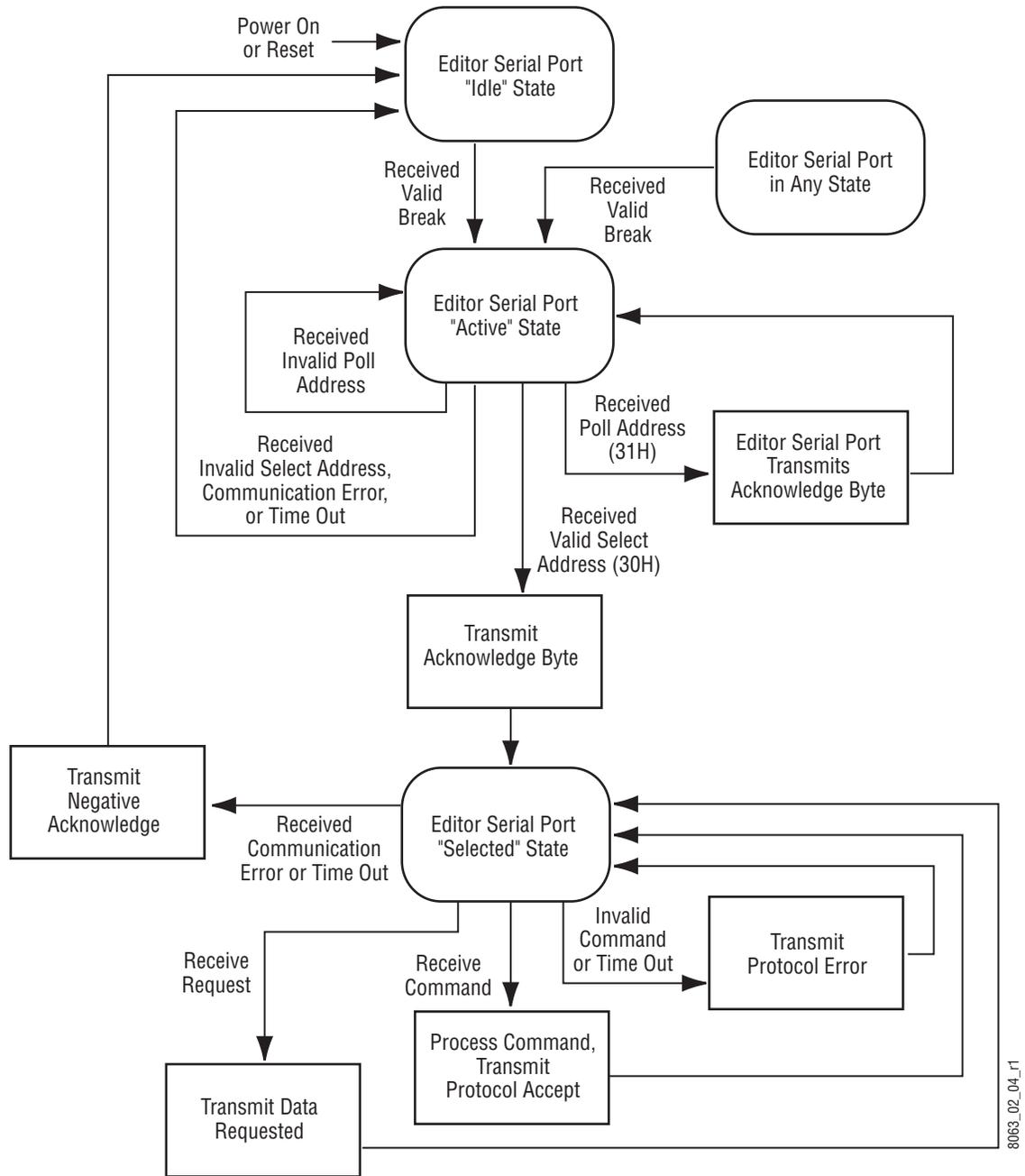
The main component of the Editor protocol is a 1-1/2 character-length "break." A break consists of a spacing or logical low (0) condition on the serial bus (see Figure 2-2) followed by a minimum of two bit-times of marking (logic 1) condition. The Bus Controller (editor) transmits a break to the switcher, forewarning of an impending message (Figure 3).

Figure 3. Break Character



The break forces all *tributaries*, or listeners (such as the switcher's editor port), from the "Idle" state into the "Active" state. A break can be sent only by the Edit Controller. When the switcher receives the break, it ceases all bus communications, resets any communication error condition, and prepares to receive an address (Figure 4).

Figure 4. Editor Protocol State Diagram



8063_02_04_r1

Address Byte

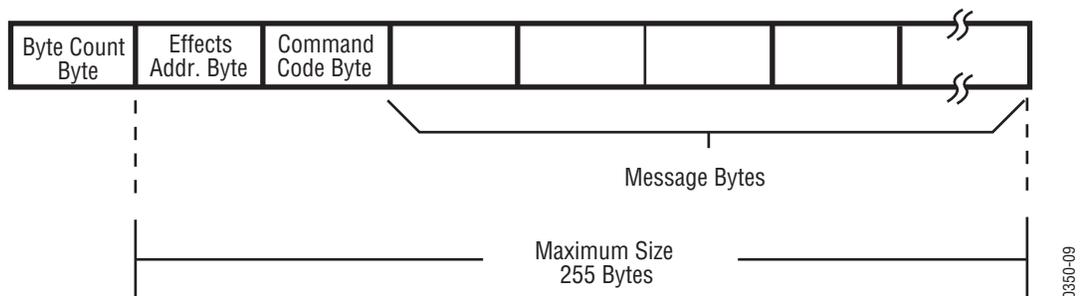
Once the break has been received, the Editor protocol advances to the active state and begins expecting a one-byte address. The switcher will respond to the switcher address 30H (Hexadecimal), when transmitted by the Edit controller. After receiving the address, the Editor protocol accepts command/message block transmissions containing commands, data requests, and other data directed toward the switcher. As long as communication errors do not occur, the Editor protocol can skip the break/address sequence during the remainder of the transmissions. Command/message blocks can be sent repeatedly, one after another, without break interruptions.

The address is one byte long and directed to a specific tributary. If the received address matches the tributary's assigned address, the tributary advances to the Selected state and awaits instructions. The address of the switcher is fixed as 30H. In response to the address, the 3000 transmits an acknowledgment byte of 84H to the controller, indicating the change in status to the "Selected" State. If an address of 31H is received, the switcher transmits an acknowledgment byte of 84H to the controller, but stays in the "Active" State. If the address does not match, the tributary drops off the bus (goes Idle) and awaits another break.

Command/Message Block Structure

In the Selected state, the Editor protocol receives command/message blocks that direct switcher operations. The basic message structure is shown in [Figure 5](#).

Figure 5. Message Structure



The maximum length of a message is 255 bytes plus the byte count byte. At a baud rate of 38,400, the maximum length command/message takes more than 4 fields to complete.

The Editor protocol can handle data continuously, receiving a contiguous stream of command/message blocks. Such an ability enables multiple switcher operations to be performed without constant cycling through the

break/address sequence. Responses (handshakes or read data) are returned in the same order as the command/message blocks are received.

Byte Count

The command/message block can range in size from two to 256 bytes. The first byte of the block contains the byte count. A byte count consists of the total number of subsequent bytes in the block; valid byte count values range from 01H to FFH (1 to 255). After receiving a valid byte count (01H to FFH) and the proper number of data bytes (1 to 255), the switcher executes the command.

Effects Address Byte

The second byte of the command/message block is the effects address byte, typically referred to as EX. This byte identifies the desired “effect bank” within the switcher where the associated command will go. Valid Effects Addresses for each command are specified later in this manual.

Command Code Byte

The command code is the third byte of the command/message block. Command codes fall into two broad categories: requests and commands. Only one command is allowed in any command/message block. Likewise, only one response is embedded in any command/message block.

Requests

Requests interrogate the status of the specified operational parameter of the switcher. When the editor issues a read command/message block, the switcher responds by sending the current status of the requested operational parameter(s) back to the editor. This status information is returned in the format of the write command/message block. By echoing the write format, the editor can later send responses to a request to the switcher and cause execution of that function.

Commands

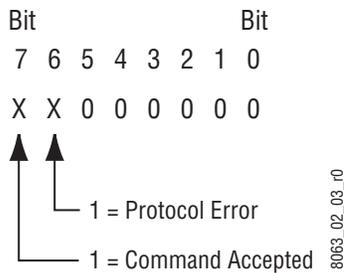
Commands are used by the editor to change operational parameter(s) or initiate a function within the switcher.

Status Replies

The switcher responds to a command by sending a two-byte status message to the editor. The first byte is the byte count which is 01H, and the second byte is the switcher's response to the command.

The two most common responses are 80H (command accepted) and 40H (protocol error) as shown in (Figure 6).

Figure 6. Status Reply Message



The status bits are described below. Note that the Command Accept bit and Protocol Error bit cannot both be set.

- If Command Accept is true, the command is forwarded to the specified effects system. Only the Command Code Byte, Byte Count Byte, and the Effect Address Byte are checked for validity. It does not mean that the switcher executed the command correctly.
- If Protocol Error is true, an illegal function may have been requested.

Error Detection

Error detection is a normal part of the communication process. Communication or “handshaking” errors can take many forms, such as parity errors, framing errors, or data overrun errors. When the Editor protocol detects communication errors, it responds with a negative acknowledge byte of 85H and then aborts all communications. Any remaining un-executed commands are discarded after the error indication occurs.

In order to resume communication, the editor must transmit a break/address sequence to return the channel to the Select state. If an error is detected during the receipt of the tributary address, the Editor protocol drops back into the Idle state without transmitting the 85H Negative Acknowledge byte.

In general, it is preferable for the editor to wait until it receives the Command Accept response before sending a new command, although it is possible to send a continuous stream of commands without waiting for replies after each one.

Editor Commands

Introduction

This section describes the editor command set used by the Kalypso Video Production Center and the Zodiac Digital Production Switcher when using the Kalypso/Zodiac editor protocol. These interface commands follow the Model 4000 interface closely, and are designed to respond to existing Model 4000 commands from an editor (with a few exceptions - such as E-MEM transfer). Commands have been extended to provide access to Kalypso and Zodiac features, like Keyer 3 and 4, which do not exist on the Model 4000. This allows an existing Model 4000 editor interface to be useful without modification, or expanded to access new switcher functionality.

In this section the Kalypso and Zodiac systems are generically called "switcher" when the same commands are used. When differences in the commands for these models exist, the differences are indicated with notes or with separate labeled tables.

Note the following for both Kalypso and Zodiac switchers:

- The E-MEM data transfer commands have been replaced with Save and Load commands. Instead of transferring data over the RS-422 serial line the Load and Save commands transfers data to and from files located on the switcher's internal hard drive. These files may be transferred from the switcher's hard drive to the editor using FTP over an ethernet connection.
- The E3 (Mask) command is not implemented. Mask buses do not exist on the switcher, instead the Utility buses are used for masking.
- The E8 (Layered) command is not implemented. Split layered mode is not accessible.
- Access is provided to all 128 switcher sources (see Command C0, C1).

Note the following for Kalypso systems only:

- E-MEM Levels are used differently in Kalypso than in the Model 4000. The Effects Address in this protocol refer to Kalypso E-MEM enable groups as identified by the default names. Kalypso may be configured to assign any E-MEM level to any enable group. This protocol assumes that the enable group will be referenced even if the E-MEM levels assigned may change.
- Access is provided to new buses: K3, K4, U1, U, and Aux Bus 1 – 46 (see Command C0, C1).
- Access is provided to secondary wipe generator (see Command C8)
- Access is provided to Primary and Secondary M/E partitions when Kalypso is in split mode (via effect address).

Note the following for Zodiak systems only:

- Access is provided to new buses: K3, K4, U1, and Aux Bus 1 – 13 (see Command C0, C1).
- Access to DSK transition doesn't work with existing 4000 code.

Editor protocol commands are detailed in tables presented on the following pages. Addresses, byte counts, important names, and supporting details are listed for each command. References to pushbuttons are shown in capital letters, using the legends as they appear on a Control Panel. Control Panel section names are shown in initial capitals.

Note All values within tables are hexadecimal unless otherwise indicated by the notation “decimal.”

Command Usage

The appropriate command to use may be determined by referring to [Table 20](#).

Table 20. Command Use Summary

Purpose	Command(s)
Basic Source Control	Source Bus (C1 – C4) Commands – page 50
	Source Select (C0) Command – page 55
	Preview Bus (E2) Command – page 57
	Split Key (E4, E5, E6, E7, E9 and EA) Commands – page 58
To simulate the pressing of certain buttons	Pushbutton Select and Control (C6, C7 and FB) Commands – page 59
Selection of Wipe Patterns	Wipe Pattern (C8) Command – page 63
Control of Transition mode pushbuttons	Transition Mode (CA) Command – page 65
Set transition rates	Auto and Key Transition Rate (CC and CD) Commands – page 67
E-MEM learns and recalls	Learn and Recall E-MEM Register (DA and DB) Commands – page 71
E-MEM and Configuration Save and Load Data	Save and Load Data (5F and DF) Commands – page 78
	Save and Load Status (6D and ED) Commands – page 82
Timeline control for multiple keyframe effects	Timeline Control (4E, 4F, CE, and CF) Commands – page 83
Commands for general use by an editor	All Stop (F2) Command – page 88
	Software Version (6C and EC) Commands – page 90

Application Examples

Several examples of serial communications using the switcher Editor Interface are provided on the following pages in the sections appropriate to the commands. In general, “Transmit” refers to communication from the editor to the switcher and “Receive” refers to communication from the switcher to the editor. Refer to specific command tables for details on these examples.

Example 1: Bringing the Editor Interface into a Ready State

Table 21. Transmit

Break	Address
BRK	30

Table 22. Receive

Status
84

Example 2: How to Set Crosspoint #6, A Row, Mix/Effects 2

Table 23. Transmit

Byte Count	Ex Address	Command	Data
03	02	C1	06

Table 24. Receive

Byte Count	Status
01	80

Example 3: How to Perform a DSK Auto Transition (Zodiak Only)

Table 25. Set DSK Elements (DSK1, DSK2, DSK3)

Function	Byte Count	Ex Address	Command	Data	
DSK1	04	04	C6	02	0E
DSK2	04	04	C6	02	10
DSK2	04	04	C6	02	2B

Table 26. Set DSL Link On for Active M/E

Function	Byte Count	Ex Address	Command	Data	
DSK LINK	04	EX	C6	02	30

Table 27. Set Mix Transition Type on M/E

Function	Byte Count	Ex Address	Command	Data	
MIX	04	EX	C6	02	06

Table 28. Trigger Auto Transition

Function	Byte Count	Ex Address	Command	Data	
AUTO TRANS	04	EX	C6	02	06

Table 29. Perform Single DSK Transition – Cut or Mix

Function	Byte Count	Ex Address	Command	Data	
DSK1 CUT	04	04	FB	02	23
DSK1 MIX	04	04	C6	02	24

Source Bus (C1 – C4) Commands

The Source Bus commands enable the editor to select any source on a bus specified by the effects address, and the command code.

Table 30. Source Bus Commands

Function	Byte Count	Effects Address	Command Code	Message
BKGD A/AUX	03	EX	C1	Source #
BKGD B	03	EX	C2	Source #
KEY1/DSK 1	03	EX	C3	Source #
KEY2/DSK 2	03	EX	C4	Source #

Kalypso Effects Addresses

Valid effects addresses for the Source Bus commands used by the Kalypso system are as indicated in [Table 31](#).

Table 31. Kalypso Effects Addresses - Source Bus Commands

	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment
Accessible Model 4000 Code	00	PGM-PST	07	Mix/Effects 3 Secondary Partition ^d	12	Aux Bus 4A	19	Aux Bus 7B
	01	Mix/Effects 1	0C	Aux Bus 1A	13	Aux Bus 4B	1A	Aux Bus 8A
	02	Mix/Effects 2	0D	Aux Bus 1B	14	Aux Bus 5A	1B	Aux Bus 8B
	03	Mix/Effects 3	0E	Aux Bus 2A	15	Aux Bus 5B	1C	Aux Bus 9A
	04	PGM-PST Secondary Partition ^d	0F	Aux Bus 2B	16	Aux Bus 6A	1D	Aux Bus 9B
	05	Mix/Effects 1 Secondary Partition ^d	10	Aux Bus 3A	17	Aux Bus 6B	--	--
	06	Mix/Effects 2 Secondary Partition ^d	11	Aux Bus 3B	18	Aux Bus 7A	--	--
Kalypso Code	40	AuxBus1A	57	AuxBus12B	6E	AuxBus24A	85	AuxBus35B
	41	AuxBus1B	58	AuxBus13A	6F	AuxBus24B	86	AuxBus36A
	42	AuxBus2A	59	AuxBus13B	70	AuxBus25A	87	AuxBus36B
	43	AuxBus2B	5A	AuxBus14A	71	AuxBus25B	88	AuxBus37A
	44	AuxBus3A	5B	AuxBus14B	72	AuxBus26A	89	AuxBus37B
	45	AuxBus3B	5C	AuxBus15A	73	AuxBus26B	8A	AuxBus38A
	46	AuxBus4A	5D	AuxBus15B	74	AuxBus27A	8B	AuxBus38B
	47	AuxBus4B	5E	AuxBus16A	75	AuxBus27B	8C	AuxBus39A
	48	AuxBus5A	5F	AuxBus16B	76	AuxBus28A	8D	AuxBus39B
	49	AuxBus5B	60	AuxBus17A	77	AuxBus28B	8E	AuxBus40A
	4A	AuxBus6A	61	AuxBus17B	78	AuxBus29A	8F	AuxBus40B
	4B	AuxBus6B	62	AuxBus18A	79	AuxBus29B	90	AuxBus41A
	4C	AuxBus7A	63	AuxBus18B	7A	AuxBus30A	91	AuxBus41B
	4D	AuxBus7B	64	AuxBus19A	7B	AuxBus30B	92	AuxBus42A
	4E	AuxBus8A	65	AuxBus19B	7C	AuxBus31A	93	AuxBus42B
	4F	AuxBus8B	66	AuxBus20A	7D	AuxBus31B	94	AuxBus43A
	50	AuxBus9A	67	AuxBus20B	7E	AuxBus32A	95	AuxBus43B
	51	AuxBus9B	68	AuxBus21A	7F	AuxBus32B	96	AuxBus44A
	52	AuxBus10A	69	AuxBus21B	80	AuxBus33A	97	AuxBus44B
	53	AuxBus10B	6A	AuxBus22A	81	AuxBus33B	98	AuxBus45A
	54	AuxBus11A	6B	AuxBus22B	82	AuxBus34A	99	AuxBus45B
	55	AuxBus11B	6C	AuxBus23A	83	AuxBus34B	9A	AuxBus46A
	56	AuxBus12A	6D	AuxBus23B	84	AuxBus35A	9B	AuxBus46B

^a New effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses for the C1 – C4 Source Bus commands used by the Zodiak system are as indicated in [Table 32](#).

Table 32. Zodiak Effects Addresses - Source Bus Commands

	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment
Accessible 4000 Code	00	PGM-PST	0D	Aux Bus 1B	13	Aux Bus 4B	19	Aux Bus 7B
	01	Mix/Effects 1	0E	Aux Bus 2A	14	Aux Bus 5A	1A	Aux Bus 8A
	02	Mix/Effects 2	0F	Aux Bus 2B	15	Aux Bus 5B	1B	Aux Bus 8B
	03	Mix/Effects 3	10	Aux Bus 3A	16	Aux Bus 6A	1C	Aux Bus 9A
	04	DSK ^a	11	Aux Bus 3B	17	Aux Bus 6B	1D	Aux Bus 9B
	0C	Aux Bus 1A	12	Aux Bus 4A	18	Aux Bus 7A	--	--
Zodiak Code	40	AuxBus1A	47	AuxBus4B	4E	AuxBus8A	54	AuxBus11A
	41	AuxBus1B	48	AuxBus5A	4F	AuxBus8B	55	AuxBus11B
	42	AuxBus2A	49	AuxBus5B	50	AuxBus9A	56	AuxBus12A
	43	AuxBus2B	4A	AuxBus6A	51	AuxBus9B	57	AuxBus12B
	44	AuxBus3A	4B	AuxBus6B	52	AuxBus10A	58	AuxBus13A
	45	AuxBus3B	4C	AuxBus7A	53	AuxBus10B	59	AuxBus13B
	46	AuxBus4A	4D	AuxBus7B				

^aFunction associated only with Zodiak.

Kalypso Source Numbers

Valid source numbers for all buses used by Kalypso systems are provided in [Table 33](#).

Table 33. Kalypso Common Source Numbers for all buses

	Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal
Accessible Model 4000 Code	Source 1	01	1	Source 15	0F	15	Source 29	1D	29	Source 43	2B	43
	Source 2	02	2	Source 16	10	16	Source 30	1E	30	Source 44	2C	44
	Source 3	03	3	Source 17	11	17	Source 31	1F	31	Source 45	2D	45
	Source 4	04	4	Source 18	12	18	Source 32	20	32	Source 46	2E	46
	Source 5	05	5	Source 19	13	19	Source 33	21	33	Source 47	2F	47
	Source 6	06	6	Source 20	14	20	Source 34	22	34	Source 48	30	48
	Source 7	07	7	Source 21	15	21	Source 35	23	35	ME1 PGM A	31	49
	Source 8	08	8	Source 22	16	22	Source 36	24	36	ME2 PGM A	32	50
	Source 9	09	9	Source 23	17	23	Source 37	25	37	ME3 PGM A	33	51
	Source 10	0A	10	Source 24	18	24	Source 38	26	38	ME4 PGM A	34	52
	Source 11	0B	11	Source 25	19	25	Source 39	27	39	ME1 PVW A	35	53
	Source 12	0C	12	Source 26	1A	26	Source 40	28	40	ME2 PVW A	36	54
	Source 13	0D	13	Source 27	1B	27	Source 41	29	41	ME3 PVW A	37	55
	Source 14	0E	14	Source 28	1C	28	Source 42	2A	42	ME4 PVW A	38	56
Kalypso Code	ME1 PGM B	39	57	Bkgnd1	4B	75	Source 57	5D	93	Source 75	6F	111
	ME2 PGM B	3A	58	Bkgnd2	4C	76	Source 58	5E	94	Source 76	70	112
	ME3 PGM B	3B	59	StillStore 1	4D	77	Source 59	5F	95	Source 77	71	113
	ME4 PGM B	3C	60	StillStore 2	4E	78	Source 60	60	96	Source 78	72	114
	ME1 PGM C	3D	61	StillStore 3	4F	79	Source 61	61	97	Source 79	73	115
	ME2 PGM C	3E	62	StillStore 4	50	80	Source 62	62	98	Source 80	74	116
	ME3 PGM C	3F	63	StillStore 5	51	81	Source 63	63	99	Source 81	75	117
	ME4 PGM C	40	64	StillStore 6	52	82	Source 64	64	100	Source 82	76	118
	ME1 PGM D	41	65	StillStore 7	53	83	Source 65	65	101	Source 83	77	119
	ME2 PGM D	42	66	StillStore 8	54	84	Source 66	66	102	Source 84	78	120
	ME3 PGM D	43	67	Source 49	55	85	Source 67	67	103	Source 85	79	121
	ME4 PGM D	44	68	Source 50	56	86	Source 68	68	104	Source 86	7A	122
	ME1 PVW 2	45	69	Source 51	57	87	Source 69	69	105	Source 87	7B	123
	ME2 PVW 2	46	70	Source 52	58	88	Source 70	6A	106	Source 88	7C	124
	ME3 PVW 2	47	71	Source 53	59	89	Source 71	6B	107	Source 89	7D	125
	ME4 PVW 2	48	72	Source 54	5A	90	Source 72	6C	108	Source 90	7E	126
	Test	49	73	Source 55	5B	91	Source 73	6D	109	Source 91	7F	127
	Black	4A	74	Source 56	5C	92	Source 74	6E	110	Source 92	80	128

Zodiak Source Numbers

Valid source numbers for all buses used by Zodiak systems are provided in [Table 34](#).

Table 34. Zodiak Common Source Numbers for all Buses

		Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal	Source Name	Hexadecimal	Decimal
Accessible Model 4000 Code		Source 1	01	1	Source 15	0F	15	Source 29	1D	29	Source 43	2B	43
		Source 2	02	2	Source 16	10	16	Source 30	1E	30	Source 44	2C	44
		Source 3	03	3	Source 17	11	17	Source 31	1F	31	Source 45	2D	45
		Source 4	04	4	Source 18	12	18	Source 32	20	32	Source 46	2E	46
		Source 5	05	5	Source 19	13	19	Source 33	21	33	Source 47	2F	47
		Source 6	06	6	Source 20	14	20	Source 34	22	34	Source 48	30	48
		Source 7	07	7	Source 21	15	21	Source 35	23	35	ME1 PGM A	31	49
		Source 8	08	8	Source 22	16	22	Source 36	24	36	ME2 PGM A	32	50
		Source 9	09	9	Source 23	17	23	Source 37	25	37	ME3 PGM A	33	51
		Source 10	0A	10	Source 24	18	24	Source 38	26	38	DSK PGM A	34	52
		Source 11	0B	11	Source 25	19	25	Source 39	27	39	ME1 PVW A	35	53
		Source 12	0C	12	Source 26	1A	26	Source 40	28	40	ME2 PVW A	36	54
		Source 13	0D	13	Source 27	1B	27	Source 41	29	41	ME3 PVW A	37	55
		Source 14	0E	14	Source 28	1C	28	Source 42	2A	42	DSK PVW A	38	56
Zodiak Code		DSK PGM B	39	57	Source 54	5A	90	Source 67	67	103	Source 80	74	116
		DSK PVW B	3A	58	Source 55	5B	91	Source 68	68	104	Source 81	75	117
		Black	4A	74	Source 56	5C	92	Source 69	69	105	Source 82	76	118
		Bkgnd1	4B	75	Source 57	5D	93	Source 70	6A	106	Source 83	77	119
		Bkgnd2	4C	76	Source 58	5E	94	Source 71	6B	107	Source 84	78	120
		StillStore 1	4D	77	Source 59	5F	95	Source 72	6C	108	Source 85	79	121
		StillStore 2	4E	78	Source 60	60	96	Source 73	6D	109	Source 86	7A	122
		StillStore 3	4F	79	Source 61	61	97	Source 74	6E	110	Source 87	7B	123
		StillStore 4	50	80	Source 62	62	98	Source 75	6F	111	Source 88	7C	124
		Source 49	55	85	Source 63	63	99	Source 76	70	112	Source 89	7D	125
		Source 50	56	86	Source 64	64	100	Source 77	71	113	Source 90	7E	126
		Source 51	57	87	Source 65	65	101	Source 78	72	114	Source 91	7F	127
		Source 52	58	88	Source 66	66	102	Source 79	73	115	Source 92	80	128
		Source 53	59	89									

Source Select (C0) Command

The Source Bus command enables the editor to select any Source on a bus specified by the effects address, and the command code. This command includes compatibility of commands C1, C2, C3 and C4.

Table 35. Source Bus Command

Function	Byte Count	Effects Address	Command Code	Message
Source Bus	03	EX	C0	Source #

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the CO Source Bus command are defined in [Table 36 on page 56](#).

Table 36. Kalypso Effects Addresses- Source Select Command

Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment
00	PGM-PST A	2B	Mix/Effects 3 Key 2	56	AuxBus12A	79	AuxBus29B
01	Mix/Effects 1 A	2C	PGM-PST Key 2 Secondary Partition	57	AuxBus12B	7A	AuxBus30A
02	Mix/Effects 2 A	2D	Mix/Effects 1 Key 2 Secondary Partition	58	AuxBus13A	7B	AuxBus30B
03	Mix/Effects 3 A	2E	Mix/Effects 2 Key 2 Secondary Partition	59	AuxBus13B	7C	AuxBus31A
04	PGM-PST A Secondary Partition	2F	Mix/Effects 3 Key 2 Secondary Partition	5A	AuxBus14A	7D	AuxBus31B
05	Mix/Effects 1 A Secondary Partition	30	PGM-PST Key 3	5B	AuxBus14B	7E	AuxBus32A
06	Mix/Effects 2 A Secondary Partition	31	Mix/Effects 1 Key 3	5C	AuxBus15A	7F	AuxBus32B
07	Mix/Effects 3 A Secondary Partition	32	Mix/Effects 2 Key 3	5D	AuxBus15B	80	AuxBus33A
08	PGM-PST B	33	Mix/Effects 3 Key 3	5E	AuxBus16A	81	AuxBus33B
09	Mix/Effects 1 B	38	PGM-PST Key 4	5F	AuxBus16B	82	AuxBus34A
0A	Mix/Effects 2 B	39	Mix/Effects 1 Key 4	60	AuxBus17A	83	AuxBus34B
0B	Mix/Effects 3 B	3A	Mix/Effects 2 Key 4	61	AuxBus17B	84	AuxBus35A
0C	PGM-PST B Secondary Partition	3B	Mix/Effects 3 Key 4	62	AuxBus18A	85	AuxBus35B
0D	Mix/Effects 1 B Secondary Partition	40	AuxBus1A	63	AuxBus18B	86	AuxBus36A
0E	Mix/Effects 2 B Secondary Partition	41	AuxBus1B	64	AuxBus19A	87	AuxBus36B
0F	Mix/Effects 3 B Secondary Partition	42	AuxBus2A	65	AuxBus19B	88	AuxBus37A
10	PGM-PST U1	43	AuxBus2B	66	AuxBus20A	89	AuxBus37B
11	Mix/Effects 1 U1	44	AuxBus3A	67	AuxBus20B	8A	AuxBus38A
12	Mix/Effects 2 U1	45	AuxBus3B	68	AuxBus21A	8B	AuxBus38B
13	Mix/Effects 3 U1	46	AuxBus4A	69	AuxBus21B	8C	AuxBus39A
18	PGM-PST U2	47	AuxBus4B	6A	AuxBus22A	8D	AuxBus39B
19	Mix/Effects 1 U2	48	AuxBus5A	6B	AuxBus22B	8E	AuxBus40A
1A	Mix/Effects 2 U2	49	AuxBus5B	6C	AuxBus23A	8F	AuxBus40B
1B	Mix/Effects 3 U2	4A	AuxBus6A	6D	AuxBus23B	90	AuxBus41A
20	PGM-PST Key 1	4B	AuxBus6B	6E	AuxBus24A	91	AuxBus41B
21	Mix/Effects 1 Key 1	4C	AuxBus7A	6F	AuxBus24B	92	AuxBus42A
22	Mix/Effects 2 Key 1	4D	AuxBus7B	70	AuxBus25A	93	AuxBus42B
23	Mix/Effects 3 Key 1	4E	AuxBus8A	71	AuxBus25B	94	AuxBus43A
24	PGM-PST Key 1 Secondary Partition	4F	AuxBus8B	72	AuxBus26A	95	AuxBus43B
25	Mix/Effects 1 Key 1 Secondary Partition	50	AuxBus9A	73	AuxBus26B	96	AuxBus44A
26	Mix/Effects 2 Key 1 Secondary Partition	51	AuxBus9B	74	AuxBus27A	97	AuxBus44B
27	Mix/Effects 3 Key 1 Secondary Partition	52	AuxBus10A	75	AuxBus27B	98	AuxBus45A
28	PGM-PST Key 2	53	AuxBus10B	76	AuxBus28A	99	AuxBus45B
29	Mix/Effects 1 Key 2	54	AuxBus11A	77	AuxBus28B	9A	AuxBus46A
2A	Mix/Effects 2 Key 2	55	AuxBus11B	78	AuxBus29A	9B	AuxBus46B

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the C0 Source Bus command are defined in [Table 37](#).

Table 37. Zodiak Effects Addresses- Source Select Command

Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment	Effects Address	Assignment
00	PGM-PST A	22	Mix/Effects 2 Key 1	40	AuxBus1A	4D	AuxBus7B
01	Mix/Effects 1	23	Mix/Effects 3 Key 1	41	AuxBus1B	4E	AuxBus8A
02	Mix/Effects 2	28	PGM-PST Key 2	42	AuxBus2A	4F	AuxBus8B
03	Mix/Effects 3	29	Mix/Effects 1 Key 2	43	AuxBus2B	50	AuxBus9A
04	DSK 1	2A	Mix/Effects 2 Key 2	44	AuxBus3A	51	AuxBus9B
05	DSK 2	2B	Mix/Effects 3 Key 2	45	AuxBus3B	52	AuxBus10A
06	DSK 3	30	PGM-PST Key 3	46	AuxBus4A	53	AuxBus10B
08	PGM-PST B	31	Mix/Effects 1 Key 3	47	AuxBus4B	54	AuxBus11A
10	PGM-PST U1	32	Mix/Effects 2 Key 3	48	AuxBus5A	55	AuxBus11B
11	Mix/Effects 1 U1	33	Mix/Effects 3 Key 3	49	AuxBus5B	56	AuxBus12A
12	Mix/Effects 2 U1	38	PGM-PST Key 4	4A	AuxBus6A	57	AuxBus12B
13	Mix/Effects 3 U1	39	Mix/Effects 1 Key 4	4B	AuxBus6B	58	AuxBus13A
20	PGM-PST Key 1	3A	Mix/Effects 2 Key 4	4C	AuxBus7A	59	AuxBus13B
21	Mix/Effects 1 Key 1	3B	Mix/Effects 3 Key 4				

Preview Bus (E2) Command

The Preview Bus command enables the editor to select any Source on the preview primary bus.

Table 38. Source Bus Command

Function	Byte Count	Effects Address	Command Code	Message
Preview Bus	03	06	E2	Source #

Source Numbers

Valid Source numbers are shown in [Table 33 on page 53](#) and [Table 34 on page 54](#).

Split Key (E4, E5, E6, E7, E9 and EA) Commands

These commands allow an editor to use the Split Key feature of the switcher. There are two modes for selecting the key source: Auto Select mode uses the key source assigned to the selected fill source. Video Select mode uses the selected fill video as the key source.

Table 39. Split Key Commands

Function	Byte Count	Effects Address	Command Code	Message
Key 1	04	EX	E6	Fill Source #, Key Source #
Key 2	04	EX	E7	Fill Source #, Key Source #
Key 3	04	EX	E9	Fill Source #, Key Source #
Key 4	04	EX	EA	Fill Source #, Key Source #
DSK 1 ^a	04	04	E4	Fill Source #, Key Source #
DSK 2 ^a	04	04	E5	Fill Source #, Key Source #
DSK 3 ^a	04	04	E9	Fill Source #, Key Source #

^a Used on Zodiac systems only

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Split Key command are defined in [Table 40](#).

Table 40. Kalypso Effects Addresses - Split Key Commands

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 1 Secondary Partition ^a

^a New assignment associated with Kalypso.

Zodiac Effects Addresses

Valid effects addresses used by Zodiac systems for the Split Key command are defined in [Table 41](#).

Table 41. Zodiac Effects Addresses - Split Key Commands

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	DSK ^a

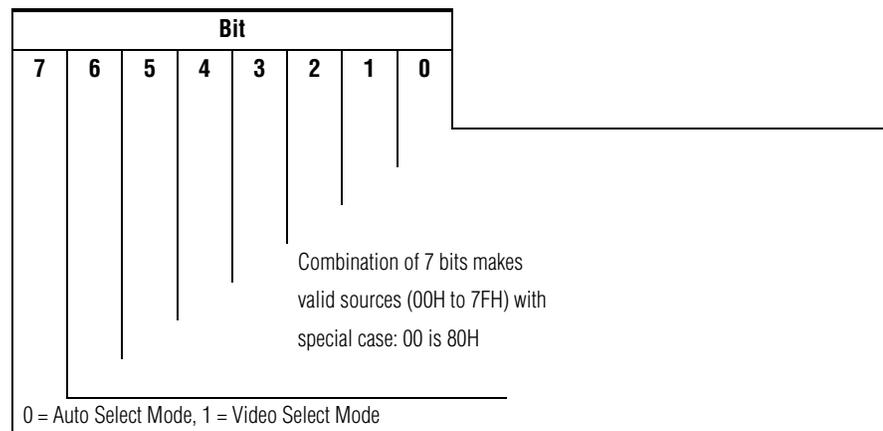
^a New assignment associated with Zodiac.

Source Numbers

Valid Source numbers are shown in [Table 33 on page 53](#) and [Table 34 on page 54](#).

The fill source is a valid source in hex. The format of the Key Source byte is provided in [Table 42](#).

Table 42. Key Source Byte Format



Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Pushbutton Control commands are defined in [Table 44](#).

Table 44. Kalypso Effects Addresses - Pushbutton Select and Control

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
0B	Master E-MEM
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 3 Secondary Partition ^a

^aNew effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the Pushbutton Control commands are defined in [Table 45](#).

Table 45. Zodiak Effects Addresses - Pushbutton Select and Control

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	DSK ^a
0B	Master E-MEM

^aFunction associated only with Zodiak.

Kalypso Pushbutton Numbers

Kalypso system pushbutton numbers supported by these commands are specified in [Table 46 on page 61](#).

Note Invalid pushbutton numbers will be ignored. To insure that at least one transition element is on, the editor should turn on the desired transition element (e.g., Bkgd, Key 1, etc.) before turning off unwanted ones. Normally, Transition Wipe 1 will use Complex 1 and Transition Wipe 2 will use Complex 2, but, this is not a requirement. It may be impossible for the edit controller to select the wipe pattern if the operator has changed this association from the UI.

Table 46. Kalypso Pushbutton Numbers

M/E Pushbuttons				Master E-MEM/Effects Edit/Effects Run Pushbuttons				
Number	Function	Press	On / Off	Number	Function	Press	On / Off	M/E
0201H	Cut	X		0502H	ENABLE M/E 1 ^a		X	
0202H	Auto Tran	X		0503H	ENABLE M/E 2 ^a		X	
0203H	Preset Black	X		0504H	ENABLE M/E 3 ^a		X	
0206H	Mix	X		0501H	ENABLE PGM-PST ^a		X	
0207H	A/B Bkgd		X	0507H	ENABLE MISC 1 ^a		X	
0208H	Pri Wipe	X		0510H	ENABLE MISC 2 ^a		X	
0204H	Sec Wipe ^a	X		0511H	ENABLE MISC 3 ^a		X	
022DH	User Trans 1 ^a	X		0512H	ENABLE MISC 4 ^a		X	
022EH	User Trans 2 ^a	X		0513H	ENABLE MISC 5 ^a		X	
022FH	User Trans 3 ^a	X		0514H	ENABLE MISC 6 ^a		X	
020EH	Key 1	X	X	0515H	ENABLE MISC 7 ^a		X	
0210H	Key 2	X	X	0508H	ENABLE DPM ^a		X	
022BH	Key 3 ^a	X	X	050CH	ENABLE ALL		X	
022CH	Key 4 ^a	X	X	0516H	Still Store Grp A ^a		X	
0212H	Key Priority	X	X	0517H	Still Store Grp B ^a		X	
0213H	Trigger ^a	X	X	0518H	Still Store Grp C ^a		X	
0223H	Key 1 Cut	X		0544H	Prev KF ^a	X ^b		
0224H	Key 1 Mix	X		0545H	Next KF ^a	X ^b		
0225H	Key 2 Cut	X		0560H	Rew ^a	X ^b		
0026H	Key 2 Mix	X		0561H	Hold Inputs ^a	X ^b		
0227H	Key 3 Cut ^a	X		0562H	Rev ^a	X	X	
0228H	Key 3 Mix ^a	X		0563H	Auto Run ^a	X	X	X
0229H	Key 4 Cut ^a	X		0564H	Stop Next ^a	X	X	
022AH	Key 4 Mix ^a	X		0565H	Run ^a	X		X
1607H	Wipe 1 Direction - Normal	X		0580H	Auto Recall ^a	X	X	
1608H	Wipe 1 Direction - Reverse	X						
1687H	Wipe 2 Direction - Normal	X						
1688H	Wipe 2 Direction - Reverse	X						

^a New pushbutton function associated with Kalypso.

^b Effects addresses are irrelevant and will be ignored. All effects addresses will be treated as 0X0B.

Zodiak Pushbutton Numbers

Zodiak system pushbutton numbers supported by these commands are specified in [Table 47 on page 62](#).

Note Invalid pushbutton numbers will be ignored. To insure that at least one transition element is on, the editor should turn on the desired transition element (e.g., Bkgd, Key 1, etc.) before turning off unwanted ones. Normally, Transition Wipe 1 will use Complex 1 and Transition Wipe 2 will use Complex 2, but, this is not a requirement. It may be impossible for the edit controller to select the wipe pattern if the operator has changed this association from the UI.

Table 47. Zodiak Pushbutton Numbers

M/E Pushbuttons				Master E-MEM/Effects Edit/Effects Run Pushbuttons				
Number	Function	Press	On / Off	Number	Function	Press	On / Off	M/E
0201H	Cut	X		022AH	Key 4 Mix ^a	X		
0202H	Auto Tran	X		0230H	DSK LINK		X	
0203H	Preset Black	X		1607H	Wipe 1 Direction - Normal	X		
0206H	Mix	X		1608H	Wipe 1 Direction - Reverse	X		
0207H	A/B Bkgd		X	1687H	Wipe 2 Direction - Normal	X		
0208H	Pri Wipe	X		1688H	Wipe 2 Direction - Reverse	X		
0209H	DSK1	X		0502H	ENABLE M/E 1 ^a		X	
020BH	DSK2	X		0503H	ENABLE M/E 2 ^a		X	
0211H	DSK3	X		0504H	ENABLE M/E 3 ^a		X	
0204H	Sec Wipe ^a	X		0501H	ENABLE PGM-PST ^a		X	
022DH	User Trans 1 ^a	X		0505H	ENABLE DSK		X	
022EH	User Trans 2 ^a	X		0507H	ENABLE MISC 1 ^a		X	
022FH	User Trans 3 ^a	X		0510H	ENABLE MISC 2 ^a		X	
020EH	Key 1	X	X	0511H	ENABLE MISC 3 ^a		X	
0210H	Key 2	X	X	0508H	ENABLE DPM ^a		X	
022BH	Key 3 ^a	X	X	050CH	ENABLE ALL		X	
022CH	Key 4 ^a	X	X	0544H	Prev KF ^a	X ^b		
0212H	Key Priority	X	X	0545H	Next KF ^a	X ^b		
0213H	Trigger ^a	X	X	0560H	Rew ^a	X ^b		
0223H	Key 1 Cut/DSK1 Cut ^a	X		0561H	Hold Inputs ^a	X ^b		
0224H	Key 1 Mix /DSK1 Mix ^a	X		0562H	Rev ^a	X	X	
0225H	Key 2 Cut /DSK2 Cut ^a	X		0563H	Auto Run ^a	X	X	X
0026H	Key 2 Mix /DSK2 Mix ^a	X		0564H	Stop Next ^a	X	X	
0227H	Key 3 Cut /DSK3 Cut ^a	X		0565H	Run ^a	X		X
0228H	Key 3 Mix /DSK3 Mix ^a	X		0580H	Auto Recall ^a	X	X	
0229H	Key 4 Cut ^a	X						

^a New pushbutton function associated with Zodiak.

^b Effects addresses are irrelevant and will be ignored. All effects addresses will be treated as 0X0B.

Wipe Pattern (C8) Command

The Wipe Pattern command selects the specified Transition Wipe 1 or Transition Wipe 2 pattern for an M/E. Only one Transition Wipe 1 and one Transition Wipe 2 pattern can be active at a time for each M/E.

Table 48. Wipe Pattern Command

Function	Byte Count	Effects Address	Command Code	Message
Wipe Pattern	03	EX	C8	Wipe #

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Wipe Pattern command are defined in [Table 49](#).

Table 49. Kalypso Effects Addresses - Wipe Pattern Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 3 Secondary Partition ^a

^a New effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the Wipe Pattern command defined in [Table 50](#).

Table 50. Zodiak Effects Addresses - Wipe Pattern Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3

Wipe Numbers

Wipe numbers range from 00H to 41H, corresponding to decimal numbers 0 through 65. The format of the Wipe Number byte is provided in [Table 51](#).

Table 51. Wipe Number Byte Format

Bit							
7	6	5	4	3	2	1	0
P/S	W	W	W	W	W	W	W

The seven least significant bits (W W W W W W W) represent the wipe number in hexadecimal. When the P/S bit is a 0, the editor selects a Transition Wipe 1 Pattern; however, when the P/S bit is a 1, the editor selects a Transition Wipe 2 Pattern. The Keyer Wipe generators are not accessible.

Table 52. Wipe Pattern Names and Hexadecimal Numbers

Wipe Pattern Name	Hexadecimal Number	Wipe Pattern Name	Hexadecimal Number	Wipe Pattern Name	Hexadecimal Number
0 Vertical	00H	22 POINT LT	16H	44 8 PT STAR – Large	2CH
1 T/L CORNER	01H	23 SQUARE	17H	45 8 PT STAR – Small	2DH
2 POINT DN	02H	24 CROSS	18H	46 SLITSCAN	2EH
3 CIRCLE	03H	25 V SAWTOOTH	19H	47 V ZIG ZAG	2FH
4 DIAG LT	04H	26 H SAWTOOTH	1AH	48 V SCAN	30H
5 TRIANGLE	05H	27 5 PT STAR	1BH	49 H X-CROSS	31H
6 CLOCK	06H	28 HEART	1CH	50 V X-CROSS	32H
7 ARROW	07H	29 H SCAN	1DH	51 V SNOWFALL	33H
8 HEXAGON	08H	30 H SPLIT	1EH	52 H SNOWFALL	34H
9 SPIRAL	09H	31 B/L CORNER	1FH	53 V QUAD SPIRAL	35H
10 HORIZONTAL	0AH	32 POINT UP	20H	54 DIAG ZIG ZAG	36H
11 T/R CORNER	0BH	33 DIAMOND	21H	55 H QUAD SPIRAL	37H
12 POINT RT	0CH	34 DUAL DIAGONAL	22H	56 T/B ZIG ZAG	38H
13 ELLIPSE	0DH	35 DOUBLE H SAWTOOTH	23H	57 L/R ZIG ZAG	39H
14 DIAG RT	0EH	36 DOUBLE V SAWTOOTH	24H	58 CHECKERBOARD	3AH
15 PENTAGON	0FH	37 4 X 3 RECTANGLE	25H	59 EXTERNAL	3BH
16 DUAL CLOCK	10H	38 SMALL RANDOM	26H	60 USER 1	3CH
17 DIAGONAL CROSS	11H	39 LARGER RANDOM	27H	61 USER 2	3DH
18 OCTAGON	12H	40 3 PT STAR	28H	62 USER 3	3EH
19 H ZIG ZAG	13H	41 4 PT STAR	29H	63 USER 4	3FH
20 V SPLIT	14H	42 6 PT STAR – Small	2AH	64 USER 5	40H
21 B/R CORNER	15H	43 6 PT STAR – Large	2BH	65 USER 6	41H

Transition Mode (CA) Command

The Transition Mode command sets up the mode for the “next transition” that will occur.

Table 53. Transition Mode Command

Function	Byte Count	Effects Address	Command Code	Message
Transition Mode	03	EX	CA	Mode

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Transition Mode command are defined in [Table 54](#).

Table 54. Kalypso Effects Addresses - Transition Mode Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 3 Secondary Partition ^a

^a New effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the Transition Mode command defined in [Table 55](#).

Table 55. Zodiak Effects Addresses - Transition Mode Command

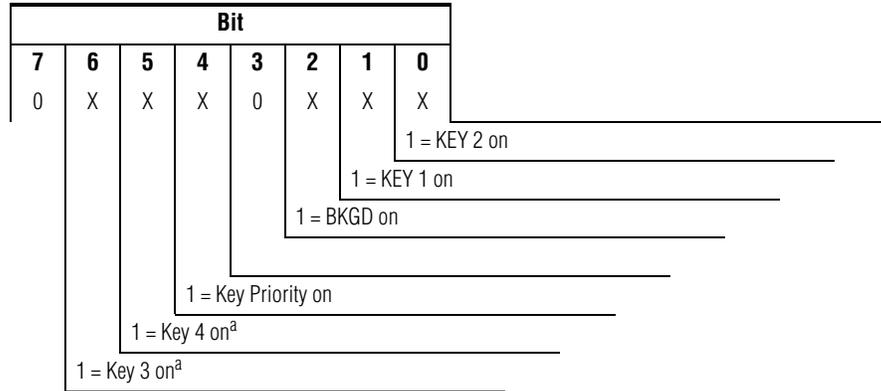
Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	DSK ^a

^a Function associated only with Zodiak.

Mode Byte — M/E

The format of the Transition Mode byte for an M/E as used by both Kalypso and Zodiac systems is provided in [Table 56](#).

Table 56. Transition Mode Byte Format – M/E

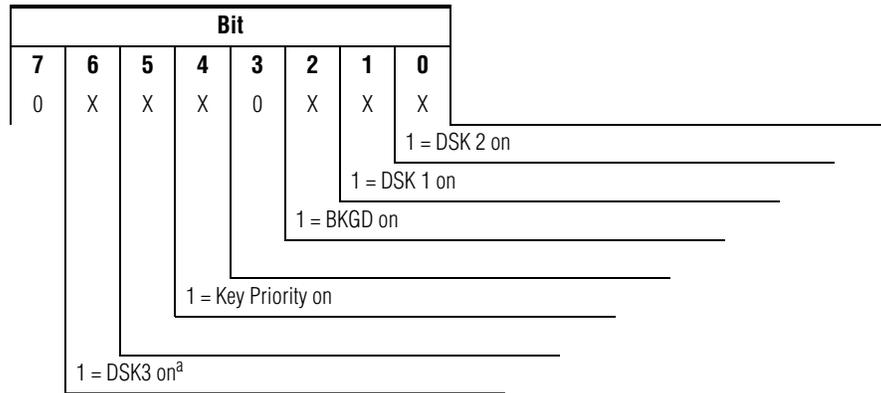


^a New bit assignment associated with Kalypso and Zodiac

Mode Byte — DSK (Zodiac Only)

The format of the Transition Mode byte for the DSK as used by Zodiac systems is provided in [Table 57](#).

Table 57. Transition Mode Byte Format – DSK



^a New bit assignment associated with Zodiac

Auto and Key Transition Rate (CC and CD) Commands

The Transition Rate commands set the rates used by the switcher to perform a transition. They can also cause a transition to take place by setting the “execute bit.” The “next transition” which will occur should have been previously selected, either manually at the panel or by sending a Transition Mode command.

Five transition rates may be set in the switcher:

- Auto transition rate,
- Key 1 transition rate,
- Key 2 transition rate,
- Key 3 transition rate, and
- Key 4 transition rate.

Which rate is used depends on:

- The “next transition” which has been selected, and
- The effects address.

The transition rates that are set depend on the effects address and the keyer bits in the units digit of the rate data. The effects address specifies which M/E. The keyer bits control which of the keyer transition rates are also set.

The Auto Transition Rate command is primarily intended to set the auto transition rate but can also set the Key 1, 2, 3 and Key 4 transition rates. If the execute bit is set, the auto transition will be performed. The Key transition rate command only sets Key transition rates.

Table 58. Auto and Key Transition Rate Commands

Function	Byte Count	Effects Address	Command Code	Message
Auto Transition Rate	05	EX	CC	HD TN UN
Key Transition Rate	05	EX	CD	HD TN UN

PGM–PST/DSK (Zodiak Only)

On Zodiak systems, the transition rate depends on the state of the “next transition”. If the next transition is set only to Background, then the auto transition rate is used. If the next transition is set to DSK 1, DSK 2, or DSK 3, or all three, then the key transition rate is used. If Background and DSK 1, DSK 2, or DSK3 is selected, then the Background (PGM–PST) transition has precedence over the DSK transition and the auto transition rate is used.

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Auto Transition Rate command are defined in [Table 59](#).

Table 59. Kalypso Effects Addresses - Auto Transition Rate Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 3 Secondary Partition ^a

^a New effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the Auto Transition Rate command are defined in [Table 60](#).

Table 60. Zodiak Effects Addresses - Auto Transition Rate Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	DSK ^a

^a Function associated only with Zodiak.

Transition Rate

Transition rates are specified in numbers of frames. To translate a transition rate in seconds to number of frames, use the following formulae:

NTSC

Frame Rate = (transition time in seconds) × (30 frames/second)

PAL

Frame Rate = (transition time in seconds) × (25 frames/second)

Example: A four second AUTO TRAN would have a frame rate of 120 frames in the NTSC standard (4 seconds × 30 frames/second = 120 frames). For the PAL standard, substitute 25 frames/second to arrive at frame rate of 100 frames.

The command specifies the auto transition rate in terms of frames ranging from 000 to 999. Specify the auto transition rate in the command by supplying three data bytes:

- HD — The hundreds digit in BCD. The MSB is the “execute” bit.
- TN — The tens digit in BCD.
- UN — The units digit in BCD. The four MSBs are control bits to select Key 1, 2, 3 and Key 4 rates in addition to the auto transition rate.

The format of the HD byte is provided in [Table 61](#).

Table 61. HD Byte Format

Bit							
7	6	5	4	3	2	1	0
X	0	0	0	X	X	X	X

Combination of 4 bits makes Hundreds digits
Rate value of 0 through 9 in BCD (0000 to 1001)

0 = Set rate only, do not perform the transition
1 = Set rate and perform the auto transition

The format of the TN byte is provided in [Table 62](#).

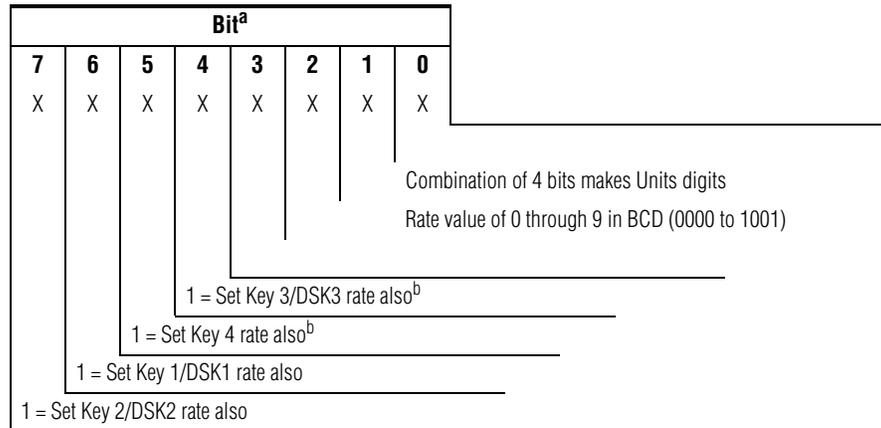
Table 62. TN Byte Format

Bit							
7	6	5	4	3	2	1	0
0	0	0	0	X	X	X	X

Combination of 4 bits makes Tens digits
Rate value of 0 through 9 in BCD (0000 to 1001)

The format of the UN byte is provided in [Table 63](#).

Table 63. UN Byte Format



^a If bits 4, 5, 6 and 7 are set to 1, or are all zero, then all Key rates will be set. If bits 4, 5, 6 and 7 are all zero, command CC will set no Key rates; command CD will set all Key rates.

^b New bit assignment associated with Kalypso and Zodiak.

Learn and Recall E-MEM Register (DA and DB) Commands

The Learn E-MEM Register command is a command that causes the Effects Address state data in the specified E-MEM register to be stored or “learned” into the E-MEM register specified in the Mode byte. This command has the same effect as manually pressing the LRN button and then selecting an E-MEM register number (0 through 99). The command also supports learning an Effects Dissolve into a register.

The Recall E-MEM command, when sent, causes the state data in the specified E-MEM (Effects Memory) register to be recalled, thus changing the current state of the switcher to the stored settings for the given effects addresses. This command performs the same action as manually recalling an E-MEM register number of 0 through 99.

Control bits may be set to force or inhibit the recall of the register as an effects dissolve. Additionally, control bits may be used to force or inhibit the running of an effect which contains multiple keyframes, and the recall of learned enables prior to the recall of the register.

Note Do not send this command more than once per video frame. Also note that a new recall command received by the switcher will override any recall that is still in progress for an effects address.

Table 64. Learn and Recall E-MEM Register Commands

Function	Byte Count	Effects Address	Command Code	Message
Learn E-MEM Register	04	EX	DA	Mode, Register Number
	06	FF	DA	Mode, Register Number, 2-byte Bit-Mask, 4000 Format
	06	FE	DA	Mode, Register Number, 2-byte Bit-Mask, Switcher Format
Recall E-MEM Register	04	EX	DB	Mode, Register Number
	06	FF	DB	Mode, Register Number, 2-byte Bit-Mask, 4000 Format
	06	FE	DB	Mode, Register Number, 2-byte Bit-Mask, Switcher Format

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Learn and Recall E-MEM Register commands are defined in [Table 65](#).

Table 65. Kalypso Effects Addresses – Learn and Recall E-MEM Register Commands

	Effects Address	Assignment		Effects Address	Assignment
Accessible Model 4000 Code	00	PGM-PST	Kalypso Code	E0	PGM-PSTSecondary Partition
	01	Mix/Effects 1		E1	Mix/Effects1 Secondary Partition
	02	Mix/Effects 2		E2	Mix/Effects2 Secondary Partition
	03	Mix/Effects 3		E3	Mix/Effects3 Secondary Partition
	06	Miscellaneous		E5	Miscellaneous1
	07	DPM 1		E6	Miscellaneous2
	0B	All enabled levels		E7	Miscellaneous3
	FF	All specified levels in the Bit-Mask ^a		E8	Miscellaneous4
	FE	Kalypso Bit-Mask ^b		E9	Miscellaneous5
		EA		Miscellaneous6	
		EB		Miscellaneous7	
		EC		DPM	
		ED		Still Store A	
		EE		Still Store B	
		EF		Still Store C	

^a The editor may learn/recall a specific Effects Address or may learn/recall all levels currently enabled on the control panel. If the Effects Address is FF then the levels to be learned/recalled are specified in a 4000 type 2-byte Bit-Mask.

^b New effects address assignment for Kalypso, for use with Kalypso type 2-byte Bit-Mask.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the Learn and Recall E-MEM Register commands are defined in [Table 66](#).

Table 66. Zodiak Effects Addresses – Learn and Recall E-MEM Register Commands

	Effects Address	Assignment		Effects Address	Assignment
Accessible Model 4000 Code	00	PGM-PST	Zodiak Code	E5	Miscellaneous1
	01	Mix/Effects 1		E6	Miscellaneous2
	02	Mix/Effects 2		E7	Miscellaneous3
	03	Mix/Effects 3		EC	DPM
	04	DSK ^a			
	06	Miscellaneous 1			
	07	DPM			
	0B	All enabled levels			
	FF	All specified levels in the Bit-Mask ^b			
	FE	Zodiak Bit-Mask ^c			

^a New effects address assignment associated with Zodiak.

^b The editor may learn/recall a specific Effects Address or may learn / recall all levels currently enabled on the control panel. If the Effects Address is FF then the levels to be learned / recalled are specified in a 4000 type 2-byte Bit-Mask.

^c New effects address assignment for Zodiak, for use with Zodiak type 2-byte Bit-Mask.

To enable levels, use C6 or C7 commands. For example, to enable the DSK level use the values given in [Table 67](#).

Table 67. Enable Levels

Function	Byte Count	Ex Address	Command	Data	
ENABLE DSK	04	0B	C6	05	05

Learn Mode Byte

The Mode byte allows an E-MEM learn with Effects Dissolve. The format of the Learn Mode byte is provided in [Table 68](#).

Table 68. Learn Mode Byte Format

Bit ^a							
7	6	5	4	3	2	1	0
0	X	0	0	0	0	0	0

1 = Learn with Effects Dissolve

^a Setting Bit 6 to one causes an Effects Dissolve to automatically occur when the specified E-MEM register is recalled.

Recall Mode Byte

The Mode byte allows an E-MEM recall with Effects Dissolve, Auto Recall, and Run, either separately or together. The Mode byte takes precedence over the Switcher button tallies. The mode byte looks at four operations:

- Force the Run/Auto Recall/Effects Dissolve/Sequence operation
- Inhibit the Run/Auto Recall/Effects Dissolve/Sequence operation
- Operate according to current Switcher panel settings

The format of the Recall Mode byte is provided in [Table 69](#).

Table 69. Recall Mode Byte Format

Bit							
7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X
							1 = Recall E-MEM Register and Inhibit Auto Recall
						1 = Recall E-MEM Register and Inhibit a Run	
					1 = Recall E-MEM Register and Inhibit Effects Dissolve		
				1 = Recall E-MEM Register and Inhibit Sequence			
			1 = Recall E-MEM Register and Force Auto Recall				
		1 = Recall E-MEM Register and Force Run					
	1 = Recall E-MEM Register and Force Effects Dissolve						
	1 = Recall E-MEM Register and Start Sequence						

Effects Dissolve, Sequence, Auto Recall, and Run each have a Force bit and an Inhibit bit. If the Force bit is on, the action is performed irrespective of the state of the switcher. If the Inhibit bit is on, the action is not performed, irrespective of the state of the switcher. If both bits are off, then the action is performed according to the switcher state, that is, what has been recalled from E-MEM, modified by the current state of the panel. (Refer to [Table 70](#).)

Table 70. Force/Inhibit Bits

Force	Inhibit	Action
1	0	Perform action
0	1	Inhibit action
0	0	Act according to panel state and register contents
1	1	Invalid

Setting Bit 5 to one recalls an E-MEM register and runs any keyframes stored in that register. This has the same effect as pushing the AUTO RUN pushbutton and then recalling an E-MEM register from the control panel. Setting Bit 1 to one recalls an E-MEM which does not run, irrespective of the state of the AUTO RUN button. If both Bits 5 and 1 are set to zero, then the E-MEM is recalled and will run only if the AUTO RUN button is on.

Setting Bit 6 to one recalls an Effects Dissolve into an E-MEM register. This has the same effect as pushing the EFF DIS pushbutton and then recalling an E-MEM register from the control panel. Setting Bit 2 prevents the Effects dissolve from taking place irrespective of whether the effect was learned with Effects Dissolve on, or the Effects Dissolve button was pressed prior to sending the Recall E-MEM command. If both Bits 6 and 2 are zero, then the Effects Dissolve will occur if the effect was learned with Effects Dissolve on, or if the Effects Dissolve button is on (has just been pressed).

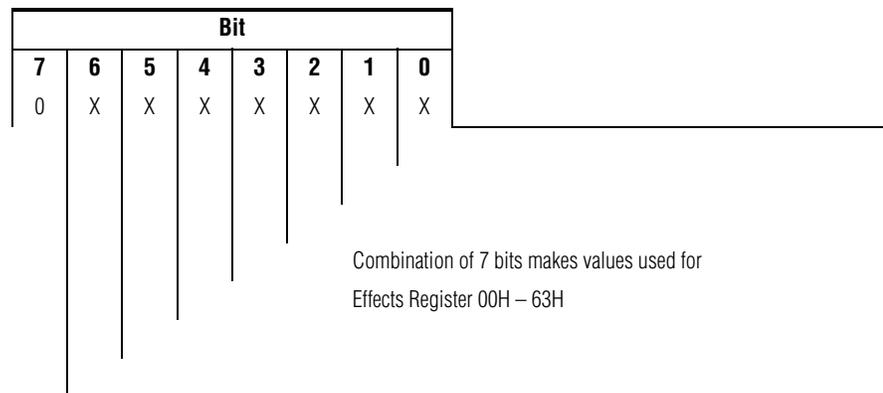
Auto Recall is controlled by the bits in the same manner as Auto Run, described above.

It is possible to combine Force bits to automatically run a register starting with an effects dissolve after the enabled levels have been recalled.

Registers

Valid E-MEM registers are 00 through 99 (00H–63H). The format of the Effects Register Number byte is provided in [Table 71](#).

Table 71. Effects Register Number Byte Format



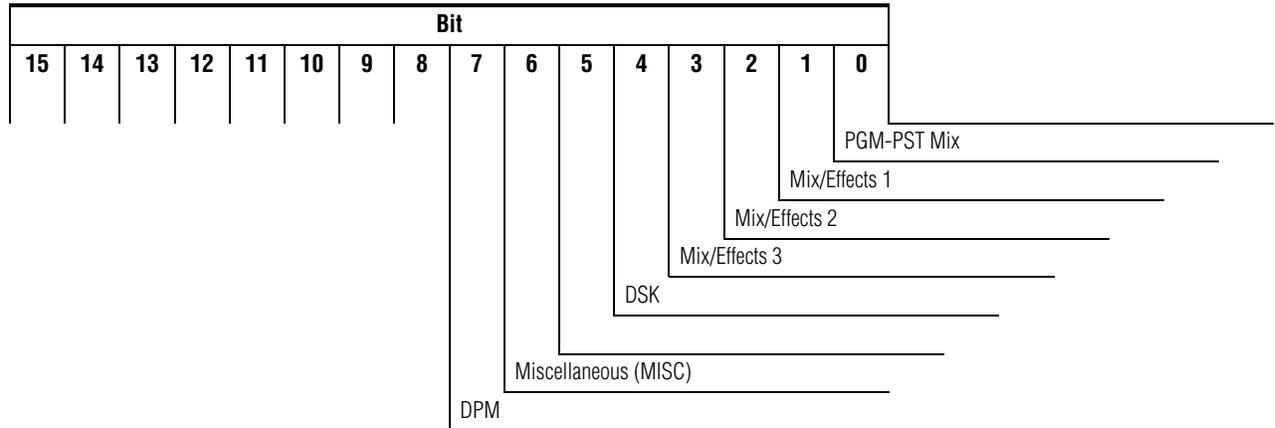
4000 Bit-Mask Format

The two-byte bit mask allows the editor to specify any combination of effects addresses 00 through 7FF. The LSB (bit 0) represents the first address, and bit 10 represents the DPM 4 address. The top bits are “don't care.” This is used with the FF Effects Address.

Zodiak Bit-Mask Format

The two-byte bit mask allows the editor to specify any combination of effects addresses 00 through 7FF. The LSB (bit 0) represents the first address, and bit 10 represents the DPM 4 address. The top bits are “don't care.” This format is used with the FE Effects Address.

Table 74. Zodiak Effects Addresses Byte Format



Save and Load Data (5F and DF) Commands

An editor may request the switcher to save an E-MEM register to a disk file on the switcher using a specified file name. At a later time the editor may request that the disk file be loaded to a specified E-MEM register. There is a separate message type for each of these operations. The message specifies the E-MEM register number and an option field followed by a null terminated text file name. The effects address specifies whether E-MEM data or configuration (either suite prefs or user prefs) are saved or loaded. An “Ack” is returned if a valid command is received. Use Status Request Command (6D) to determine if the file operation is successful.

Table 75. Save and Load Data Commands

Function	Byte Count	Effects Address	Command Code	Message
Save Data	06H to FFH	EX	5F	Reg Number, Opt(2bytes), Name (up to 250 bytes)
Load Data	06H to FFH	EX	DF	Reg Number, Opt(2bytes), Name (up to 250 bytes)

Effects Addresses

Valid effects addresses for the Save and Load Data commands defined in [Table 76](#) are new.

Table 76. Effects Addresses - Save and Load Data Commands

Effects Address	Assignment	Byte Format Reference	
		Save Option	Load Option
0B	All E-MEM Levels	Table 77	Table 83
1B	Source Mem	Table 78	Table 84
1C	RMem	Table 79	Table 85
1D	System Config	Table 80	Table 86
1E	Suite Prefs	Table 81	Table 87
1F	User Prefs	Table 82	Table 88

Reg Number

The value in this byte is a register number from 0 to 99 (decimal).

Name

The name is a null terminated string from 1 to 250 bytes long that specify a path name for the E-MEM file to be loaded or saved. The name specified is appended to the string “C:\EMEM\” to form the complete path name. If the editor supplies no name (a single zero byte) the Editor protocol uses a default name of “EDITOR”, so that the complete path name on the switcher is “C:\EMEM\EDITOR”.

The file name is automatically generated by the switcher and is of the form “REGnnn.EMM”, where nnn is the decimal register number.

Examples

09 0B 5F 00 00 00 41 42 43 00

Makes Kalypso save E-MEM register 0 to the following file:
"C"\EMEM\ABC\REG000.EMM"

09 0B DF 63 00 00 41 42 43 00

Makes Kalypso load the following file to E-MEM register 99:
"C"\EMEM\ABC\REG099.EMM"

06 0B 5F 01 00 00

Makes Kalypso save to E-MEM register 1 the following file:
"C"\EMEM\EDITOR\REG001.EMM"

Save Option Byte Formats

Table 77. E-MEM (0B)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

1 = Don't Overwrite; 0 = Force Overwrite
1 = All Levels Learned; 0 = All Levels Enabled

Table 78. SourceMem (1B)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

1 = Don't Overwrite; 0 = Force Overwrite

Table 79. RMem (1C)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

1 = Don't Overwrite; 0 = Force Overwrite

Table 80. System Config (1D)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

1 = Don't Overwrite; 0 = Force Overwrite

Table 81. Suite Prefs (1E)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

1 = Don't Overwrite; 0 = Force Overwrite

Table 82. User Prefs (1F)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

1 = Don't Overwrite; 0 = Force Overwrite

Load Option Byte Formats

Table 83. E-MEM (0B)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

0 = Load; 1 = Recall after Load
1 = All Levels Learned; 0 = All Levels Enabled

Table 84. SourceMem (1B)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 85. RMem (1C)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 86. System Config (1D)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 87. Suite Prefs (1E)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	0

No Options are currently defined.

Opt 1
Opt 2
Opt 3
Opt 4

Table 88. User Prefs (1F)

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	0

No Options are currently defined.

Opt 1
Opt 2
Opt 3
Opt 4

Save and Load Status (6D and ED) Commands

The switcher will respond with a Status Response message on receipt of a Status Request command. The status code reflects the current state of file operations.

Table 89. Save and Load Status Commands

Function	Byte Count	Effects Address	Command Code	Message
Status Request	02	EX	6D	None
Status Response	04	EX	ED	Status (2bytes)

Effects Addresses

Valid effects addresses for the Save and Load Status command are defined in [Table 90](#). (This is new code generated for Kalypso and Zodiak systems.)

Table 90. Effects Addresses - Save and Load Status Command

Effects Address	Assignment
0B	All E-MEM Levels
1B	Source Mem
1C	RMem
1D	System Config
1E	Suite Prefs
1F	User Prefs

Status Message

Valid Status code for the Save and Load commands are defined in [Table 91](#). (This is new code generated for Kalypso and Zodiak systems.)

Table 91. Status Codes - Save and Load

Status	Message
0	Idle
1	Error
2	Loading
3	Saving

Timeline Control (4E, 4F, CE, and CF) Commands

The Timeline Control commands allow the editor to manipulate a specified effect's timeline. The controlling device may read the current position, write a new position, or trim the current position by a specified amount. All positions or trims are measured as a field count. The switcher's response to the Read Timeline command is the Set Timeline command.

Table 92. Timeline Control Commands

Function	Byte Count	Effects Address	Command Code	Message
Read Timeline Time (1)	02	EX	4E	None
Set Timeline Time (2)	06	EX	CE	4 byte value
	08	FF	CE	Value + 2 byte bit mask
	08	FE	CE	Value + 2 byte bit mask Kalypso format
Trim Timeline Time (2)	06	EX	CF	4 byte value
	08	FF	CF	Trim value + 2 byte bit mask
	08	FE	CF	Trim value + 2 byte bit mask Kalypso format
Run Timeline (2)	06	EX	4F	4 byte value
	08	FF	4F	Speed value + 2 byte bit mask
	08	FE	4F	Speed value + 2 byte bit mask Kalypso format

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the Timeline Control commands are defined in [Table 65](#).

Table 93. Kalypso Effects Addresses – Timeline Control commands

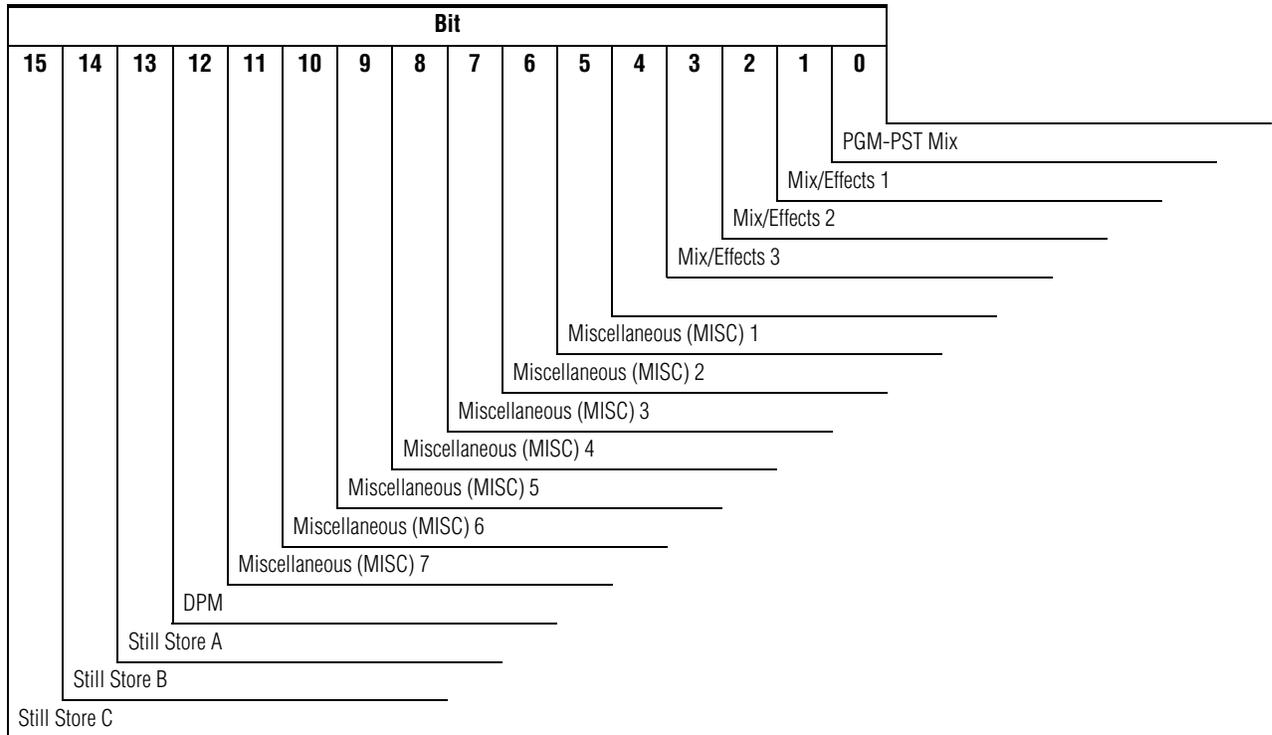
Effects Address		Assignment	Effects Address		Assignment
Accessible Model 4000 Code	00	PGM-PST	Kalypso Code	E0	PGM-PST Secondary Partition
	01	Mix/Effects 1		E1	Mix/Effects1 Secondary Partition
	02	Mix/Effects 2		E2	Mix/Effects2 Secondary Partition
	03	Mix/Effects 3		E3	Mix/Effects3 Secondary Partition
	06	Miscellaneous		E5	Miscellaneous1
	07	DPM 1		E6	Miscellaneous2
	0B	Master Timeline (all enabled levels)		E7	Miscellaneous3
	FF	All specified levels in the Bit-Mask ^a		E8	Miscellaneous4
	FE	Kalypso Bit-Mask ^b		E9	Miscellaneous5
		EA		Miscellaneous6	
		EB		Miscellaneous7	
		EC		DPM	
		ED		Still Store A	
		EE		Still Store B	
		EF		Still Store C	

^a Use the 4000 style 2-byte Bit-Mask,

^b New effects address assignment associated with Kalypso type 2-byte Bit-Mask.

resents the first address, and bit 14 represents the Still Store C address. The 4 bit is "don't care." This format is used with the FE Effects Address.

Table 95. Kalypso Effects Addresses - Byte Format

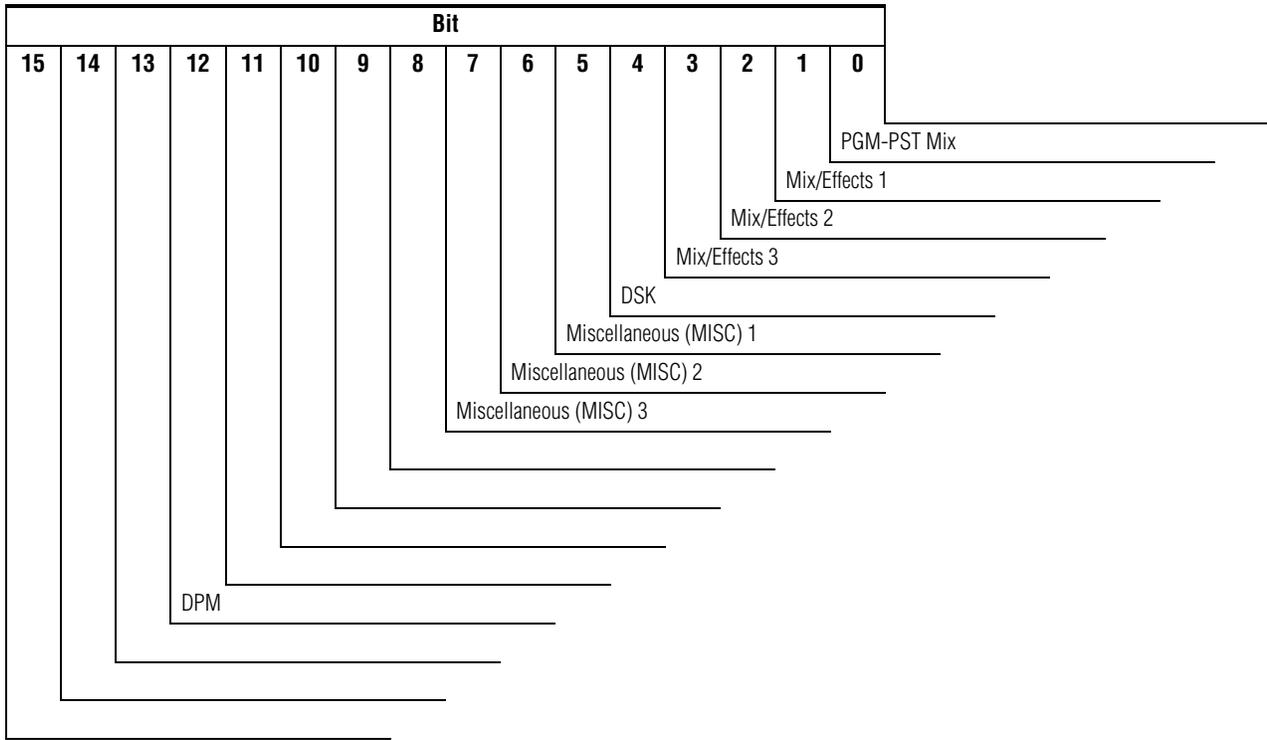


Zodiak Bit-Mask Format

The two-byte bit mask used by Zodiak systems allows the editor to specify any combination of effects addresses 00 through 7FFF. The LSB (bit 0) rep-

represents the first address, and bit 12 represents the DPM address. This format is used with the FE Effects Address.

Table 96. Zodiac Effects Addresses Byte Format



Data Field Format

All time data (field count) and speed data is specified to be binary data in S15.16 format. The lower two bytes specify a fractional field value. The topmost bit is the sign bit and the next 15 bits represent whole field values. Note the following examples.

- +1 field : 00 01 00 00
- 1 field : FF FF 00 00
- +1 (unity) speed : 00 01 00 00
- 1/2 speed : FF FF 80 00

Setting the Run Time for a speed of zero is the same as sending a STOP command.

The editor is allowed to write a negative field count or to trim in a negative direction which results in a negative field count. In either case the result is to display a negative timecode in the menu and position the video at the 1st

keyframe. The video will remain at that position until one of the following occurs:

- The effect runs long enough for the field count to pass zero
- The editor successively trims the field count in a positive direction until zero is passed
- The editor writes a positive, non-zero field count to the switcher

Note Kalypso and Zodiac currently ignore the speed parameter in the Timeline Run command. Run always causes the effect to run at +100% speed.

Examples

08 0C CF FF FF 00 00 07 80 ==> Trim all DPM levels by -1 field.

08 0C CE 00 1E 00 00 00 0E ==> Set the 3 M/Es to 30 fields.

08 0C 4F 00 01 00 00 00 0E ==> Run the effects for all 3 M/Es.

All Stop (F2) Command

The ALL STOP command is issued to set all or a specific part of the switcher to a known state. The command causes the following functions to occur:

- Any ongoing KEY MIX is stopped,
- An ongoing AUTO TRAN (Transition) is stopped,
- An ongoing EFF DIS (Effects Dissolve) is stopped,
- An ongoing SEQ (Sequence) is stopped,
- The lever arms are set to a limit,
- The transition type is set to MIX,
- The transition mode is set to BKGD,
- All Keys are removed,
- Stop Effect Run, and
- The Source selections remain unchanged.

Table 97. All Stop Command

Function	Byte Count	Effects Address	Command Code	Message
Write	03	EX	F2	FF

Kalypso Effects Addresses

Valid effects addresses used by Kalypso systems for the All Stop command are defined in [Table 98](#).

Table 98. Kalypso Effects Addresses - All Stop Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	PGM-PST Secondary Partition ^a
05	Mix/Effects 1 Secondary Partition ^a
06	Mix/Effects 2 Secondary Partition ^a
07	Mix/Effects 3 Secondary Partition ^a
0B	All Levels

^a New effects address assignment associated with Kalypso.

Zodiak Effects Addresses

Valid effects addresses used by Zodiak systems for the All Stop command are defined in [Table 99](#).

Table 99. Zodiak Effects Addresses - All Stop Command

Effects Address	Assignment
00	PGM-PST
01	Mix/Effects 1
02	Mix/Effects 2
03	Mix/Effects 3
04	DSK
0B	All Levels

Software Version (6C and EC) Commands

The switcher will respond with a Software Version Response message on receipt of a Request Software Version command. The response will be returned two fields after the request is received. The Version string is ASCII Text of the format:

<major>.<minor>.<patch><A or B><release>, e.g., 2.0.0B19

Table 100. Software Version Commands

Function	Byte Count	Effects Address	Command Code	Message
Request Software Version	02	EX	6C	None
Software Version Response	04H to FFh	EX	EC	Model number, version string

Effects Address

The valid Effects Address for the Software Version command is defined in [Table 101](#).

Table 101. Effects Address - Software Version Command

Effects Address	Assignment
00	PGM-PST

Model Number

Model numbers are as defined in [Table 102](#).

Table 102. Model Numbers

Model Number	Meaning
01	3000-2
02	3000-3
03	4000-2A
04	4000-2B
05	4000-3
06	2200-2
07	2200-2i
08	Kalypso 3-M/E + PP
09	Kalypso 2-M/E + PP
0A	Kalypso 1-M/E + PP
0B	Kalypso 0-M/E + PP
0C	Zodiak 2.5 -M/E
0D	Zodiak 3-M/E

Version Number

The software Version number comprises a Major Version number, a Minor Version number, and two fix-level parameters. The Major Version and

Minor Version are each represented as a two-byte ASCII character, and the two fix-level parameters are each represented as a one-byte ASCII character.

Switcher Model Features

The following is a brief summary of the features associated with the various switcher models.

Table 103. Switcher Model Summary

Model	Select Buttons	Inputs/Sources	M/Es	PGM-PST/DSK	Other
3000-2	16	64 inputs	2	Yes	Composite
3000-3	24		3		
4000-2B	16		2		Component
4000-2A	24		2		
4000-3	24		3	Component, layered mode on DSK	
2200-2	16		2		
2200-2i	16	2	No	Component	
Kalypso 3-M/E + PP (Kalypso 4-M/E)	32	128 sources	3	Yes/No	Component
Kalypso 2-M/E + PP	TBD		2		
Kalypso 1-M/E + PP (Kalypso 2-M/E)	24		1		
Kalypso 0-M/E + PP	TBD		0		
Zodiak 2.5-M/E	24		2.5	Yes/Yes	
Zodiak 3-M/E	24		3		

Peripheral Bus II Protocol

Introduction

This section describes the Peripheral Bus II (PBus II) serial protocol and related commands, which can be used to allow a master (controlling) device to control external (controlled) devices that support the protocol.

Controlling devices able to use the Peripheral Bus II protocol include the Kalypso Video Production Center and the Zodiac Digital Production Switcher. In this section these devices are referred to by the generic term “switcher”. Earlier Grass Valley Group switchers (Model 200, 250, 3000, 4000, and 2200) also use the Peripheral Bus II protocol, acting as controlling devices. However, some very early Grass Valley Group switchers (Model 300, 1600/1680) use a different protocol (called the Peripheral protocol), which is incompatible with the Peripheral Bus II protocol described here.

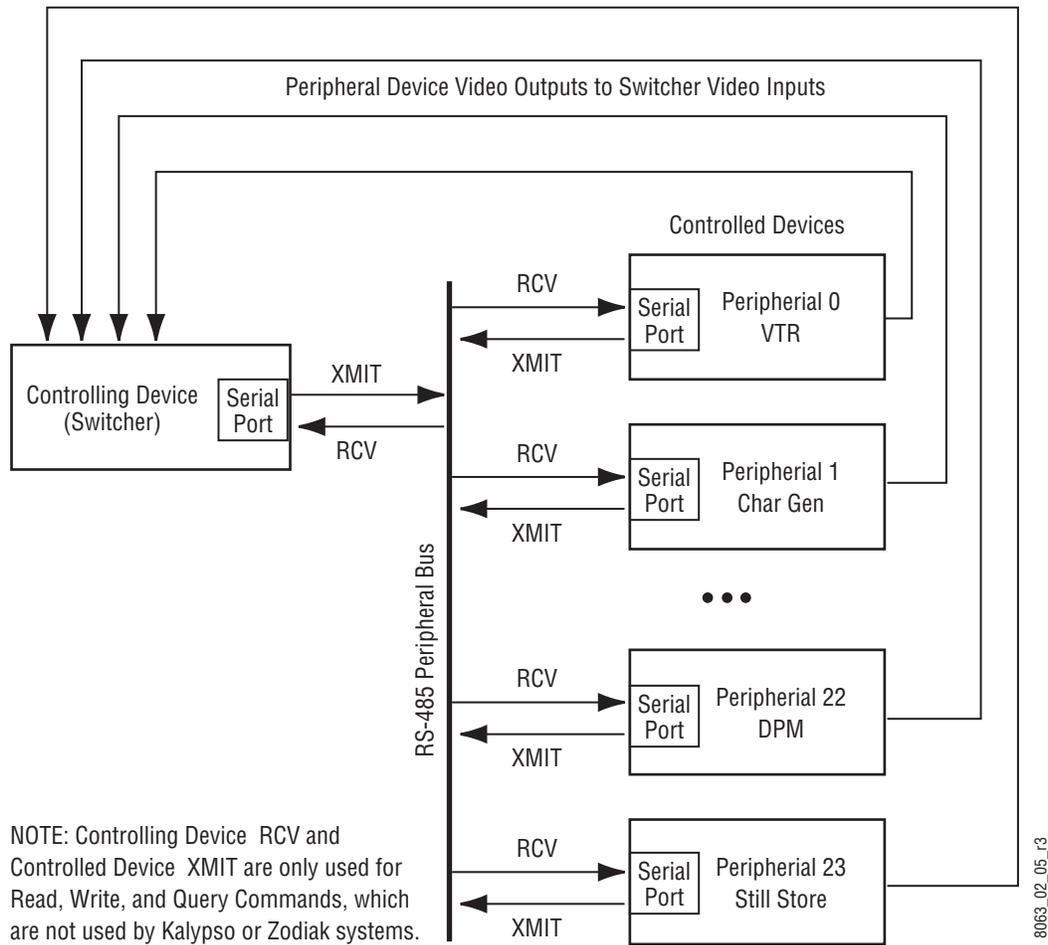
Programming a controlled device to communicate properly with the switcher requires a thorough understanding of the protocol and the available commands. Controlled devices may be programmed to support different sets of the available PBus II commands.

Overview

System Description

The Peripheral Bus II system consists of a single controlling device (the switcher) and up to 24 peripheral controlled devices connected to a serial bus ([Figure 7](#)). The controlling device communicates with controlled devices via a serial connection. All devices on the bus must operate at the same baud rate, parity, and communication standard (RS-485).

Figure 7. Peripheral Bus System Block Diagram Example



E-MEM System Interaction (Learns, Recalls)

The switcher can be set to address a particular controlled device, such as a character generator, during E-MEM learns and recalls. When a switcher effect is learned into an E-MEM register, the switcher sends a command over the peripheral bus to the controlled device telling it to learn its current status into a memory register of its own. Later, when the same switcher E-MEM register is recalled, the switcher will send a command over the peripheral bus to instruct the controlled device to recall the effect that it previously had stored in its memory.

The controlled device should store the learned information in nonvolatile memory. The switcher does not keep track of what has been learned on the controlled device, and does not ask for controlled device learn or recall information.

For key framed effects in a controlled device (like a DPM), learning the current status may not be sufficient. In this case, the controlled device should associate its current effect number with the register number sent in

the Learn command. When a Recall command is received, the controlled device should look up the effect that was active at the time and recall that effect. For example, if the controlling device sends a learn 5 command and the controlled device's current effect is 37, then when the controlling device sends a recall 5 command, the controlled device should recall its effect 37. The cross reference between PBus II register numbers and the controlled device's effect number should be retained in nonvolatile memory and some provision for saving this information to disk for transport to another system should be provided.

Grass Valley Group switchers use PBus II effect numbers 0 through 99, corresponding to the switcher's E-MEM register numbers 0 through 99. The controlled device must provide storage for all 100 PBus II effect numbers when interfacing to a Grass Valley Group switcher. Grass Valley Group switchers never send learn or recall commands for register numbers above 99 (0x63).

Some controlled devices (e.g., Kaleidoscope) run effect timelines as part of the Recall command. We do not recommend this practice, or at least suggests a user preference be available on the controlled device to control running the effect on Recall. In general, the controlled device should first recall an effect and prepare to run it when a Recall is received. A Trigger should then be employed to cause the effect to run.

Trigger Interactions

The switcher can also be set to send triggers to controlled devices over the peripheral bus. Triggers act like GPIs. They can trigger any controlled device function that has been programmed for control via the peripheral bus. The switcher can send a trigger to a controlled device when it recalls an effect (the trigger is on the first keyframe), or the trigger can be placed anywhere on the switcher effect's timeline to send triggers during the running of an effect.

It is important to understand the purpose of triggers relative to recalls. A peripheral bus Recall can be sent when the controlling device recalls one of its effects. The Recall instructs the controlled device to recall an appropriate effect or return to a specific state. Triggers are intended to cause some activity to occur within the controlled device.

An example of using recalls and triggers is having a controlled device that feeds video to the switcher select an output signal when a specific switcher effect is recalled (a Recall command is sent out the peripheral bus), and then change its output at specific times when the switcher runs the effect (send Trigger commands at specific keyframes).

The Peripheral Bus II protocol does not define or recommend which function is associated with a specific trigger number. It is recommended that a Digital Video Effects system implement Run, Rewind, and Stop triggers at

minimum. A Digital Disk Recorder should implement Play, Stop and Cue at minimum, and use Recall commands to load clips.

Command Timing and Frame Accuracy

It is possible to use the Peripheral Bus II protocol for frame accurate control, but this is not guaranteed under all conditions. PBus II uses a serial interconnect and the commands are long. It is possible to saturate the serial link with commands especially at lower baud rates, making the link non-deterministic. For example, a recall command is 11 characters. At 9600 baud this supports only 16 characters per field, sufficient for only one command per field. At 38.4 Kb, there is link bandwidth to support 64 characters per field, enough to send three recall and three trigger commands.

It is the responsibility of the controlling device to output commands at an appropriate time so the controlled devices can respond in sync with other activities of the system. The controlling device should provide and specify deterministic responses to triggers and recalls.

Protocol Description

The Peripheral Bus II protocol provides a means for orderly communication by a controlling device to up to 24 peripheral controlled devices on a serial data bus.

Hardware Interface

Communications follow the RS-485 standard at 9.6, 19.2, 38.4 k, 56.7, 76.8 Kb, or 115.2 KBaud with even, odd, or no parity. RS-485 is a derivative of RS-422 that specifies controlled devices must disable their transmit drivers when not sending information. This is required because all controlled devices share a common connection ([Figure 7 on page 94](#)).

RS-422 may be used instead of RS-485 only if a single line is connected from the controlling device's transmitter to all the controlled devices' receivers and the controlled devices' transmitter lines are completely disconnected. This configuration prevents the controlling device from receiving information from the controlled devices. This disrupts Read, Write, and Query functions, but these commands are not sent by Kalypso or Zodiac systems (of Grass Valley Group switchers, only the Model 200/250 uses these commands).

All devices connected to the peripheral bus must be set to exactly the same communications standard. The recommended settings are 38.K baud and no parity.

Command Structure

The Peripheral Bus II protocol includes six types of messages: Three are essential for a fully functional PBus II interface, and three are optional.

- Learn,
- Recall,
- Trigger,
- Query (optional)
- Read (optional), and
- Write (optional).

The peripheral interface command set has been constructed from ASCII characters. This permits the commands to be displayed on a conventional computer terminal. Additionally, the ASCII characters used for command codes are different from those used for data in order to simplify the controlled device software and help prevent coding errors.

Each message sent out on the peripheral bus has a three part format. A unique ASCII character serving as a keyword is sent first, additional data follows, and finally a carriage return terminates the message. In order to eliminate problems caused by interleaving spaces and line feeds that may have been inserted for readability, those characters are ignored in message parsing by both the controlling and controlled devices.

Twenty-three ASCII characters comprise the complete set of characters available. Seven ASCII characters represent keywords, and 16 characters represent data (Table 104).

Table 104. Peripheral Bus II ASCII Characters

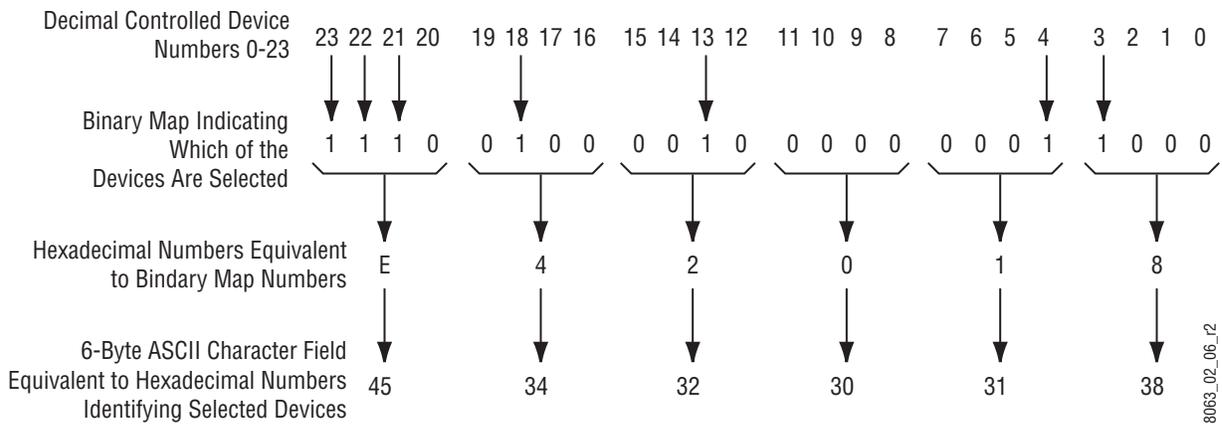
Category	Character	Hex	Description
Keyword	L	4C	Learn
	R	52	Recall
	T	54	Trigger
	Q	51	Query
	S	53	Read
	W	57	Write and Read Response
	<carriage return>	0D	End of Message
Data	1-9	30-39	
	A-F	41-46	

Controlled Device Identification

Because the controlling device communicates with multiple controlled devices, a mechanism is required to specify which device a particular command is intended for. It is also desirable to be able to send the same command to multiple devices simultaneously.

PBus II commands employ a six-byte field of ASCII characters (dddddd) representing six hex numbers to specify which controlled devices are to receive the command. The six hex numbers represent a binary 24 bit map. A set bit (1) in the map means the controlled device assigned to that bit number is included, and a (0) bit means the controlled device is not included (see [Figure 8](#)).

Figure 8. Controlled Device Selection Data Interpretation



Commands

Note The Kalypso and Zodiac systems use the Learn, Recall, and Trigger commands. The Query, Read, and Write commands and responses are not used by these systems, but command descriptions for them are included in this document for completeness.

Learn Command

The controlling device sends a Learn command to a selected controlled device to cause the controlled device to “learn” its current status or multiple keyframe effect. The controlled device’s response to the Learn command should be to store in its own memory whatever data is needed in order for it to recall its current setup at some later time. No further response from the controlled device is required.

The format of the Learn command is:

Lddddddrrr<cr>

- The first byte is the ASCII keyword for Learn, L (4CH).
- Next a six-byte field of ASCII characters (dddddd) acts as a bit map that identifies which controlled devices are included in the Learn (see [Figure 8 on page 98](#)).
- Next the controlling device sends a three-byte field of ASCII characters (rrr) representing a hex number that identifies the hex register number (000H-FFFH) from which the controlling device will recall the learned effect.
- A carriage return ends the message.

Recall Command

The controlling device sends a Recall command to a selected controlled device to cause the controlled device to “recall” a previously stored effect. The controlling device would send the Recall command some time in advance of the vertical interval in which the controlling device will actually perform the Recall. The controlled device’s response to the Recall command should be to recall from its own memory the specified effect. No further response from the controlled device is required.

The format for a Recall command is:

Rddddddrrr<cr>

- The first byte is the ASCII keyword for Recall, R (52H).
- Next a six-byte field of ASCII characters (dddddd) acts as a bit map that identifies which controlled devices are included in the Recall (see [Figure 8 on page 98](#)).
- Next the controlling device sends a three-byte field of ASCII characters (rrr) representing a hex number that identifies the hex register number from which the controlling device will recall the learned effect (000H-FFFH).
- A carriage return ends the message.

Trigger Command

The controlling device sends a Trigger command to a selected controlled device to cause the controlled device to launch or run an event which has previously been readied. An example of this is an interface to a character generator where a disk read, whose execution time is unpredictable, recalls the text of a credit roll sequence. This event is then followed by a precisely timed Trigger command which causes the credit roll to begin.

The Trigger command would be sent either as part of a register recall, when a trigger keyframe is encountered on an effect timeline, or as the result of some manual operation on the panel. When invoked as part of a Recall command, the Trigger command is sent to the selected controlled device prior to the vertical interval in which the controlling device will actually perform the Recall.

No response to the message is required of the controlled device other than to launch the desired event.

The format for a Trigger command is:

Tddddddg<cr>

- The first byte is the ASCII keyword for Trigger, T (54H).
- Next a six-byte field of ASCII characters (dddddd) acts as a bit map that identifies which controlled devices are included in the Trigger (see [Figure 8 on page 98](#)).
- Next the controlling device sends the ASCII character (g) representing a hex number that corresponds to the controlled device function which the controlling device wants the controlled device to launch (0H-FH).
- A carriage return ends the message.

Query Command

The controlling device sends a Query command to the selected controlled device to determine whether it is connected to the bus and operating. The selected controlled device should return a Query response within four (4) television fields. If no response is received within that time, the controlling device will assume that the controlled device is not connected or is unable to answer due to minimum implementation of the protocol.

The format of the Query command is:

```
Qdd<cr>
```

- The first byte is the ASCII keyword for Query, Q (51H).
- Next a two-byte field of ASCII characters (dd) representing a hex number is sent that corresponds to the controlled device number (00H-17H).
- A carriage return ends the message.

Query Response

A controlled device sends a Query response to the controlling device in response to a Query. The Query response must be returned within four (4) television fields, otherwise the controlling device will assume that the controlled device is not there or is not capable of responding.

In order to provide the Query response, the controlled device must enable its RS-485 drivers, transmit the response, and then disable its drivers. It is essential that the controlled device do this cleanly, with the driver enabled only long enough to transmit the message. It is also very important that the controlled device leave its drivers disabled at all other times.

The format for the Query Response is:

```
Q[device type]<cr>
```

- The first byte is the ASCII keyword for Query, Q (51H), the same as the Query command.
- Next an ASCII alphanumeric field of up to 16 characters is sent. These characters, which are treated as raw data, identify the device. For example, "GVG HOR BUS 35" indicates a Grass Valley Group Horizon routing switcher bus number 35.
- A carriage return ends the message.

Read Command

The controlling device sends a Read command to a selected controlled device to request the contents of the controlled device's register. The controlled device responds with a Write response (see [page 104](#)). If there is no reply from the controlled device within four (4) television fields, the controlling device assumes that the device is not capable of executing the command.

The format of the Read command is:

Rddrrr<cr>

- The first byte is the ASCII keyword for Read, R (52H).
- Next a two-byte field of ASCII characters (dd) representing a hex number is sent that corresponds to the controlled device number (00H-17H).
- Next three ASCII characters (rrr) are transmitted, representing the hex number of the register which the controlling device wants to read (000H-FFFH)
- A carriage return ends the message.

Write Command

The controlling device sends a Write command to the selected controlled device in order to transmit data to the controlled device.

The format of the Write command is:

`Wddrrrr [reg data] <cr>`

- The first byte is the ASCII keyword for Write, W (57H).
- Next a two-byte field of ASCII characters (dd) representing a hex number is sent that corresponds to the controlled device number (00H-17H).
- Next three ASCII characters are transmitted which represent the hex number of the register which the controlling device wants to write to (000H-FFFH).
- Next the register number, up to 16 ASCII characters, which are treated as raw data, may be sent.
- A carriage return ends the message.

Write Response

A controlled device sends a Write response to the controlling device in response to a Read command. The Write response must be returned within four (4) television fields, otherwise the controlling device will assume that the controlled device is not there or is not capable of responding.

In order to provide the Write response, the controlled device must enable its RS-485 drivers, transmit the response, and then disable its drivers. It is essential that the controlled device do this cleanly, with the driver enabled only long enough to transmit the message. It is also very important that the controlled device leave its drivers disabled at all other times.

The format of the Write response is:

`wddrrrr [reg data] <cr>`

- The first byte is the ASCII keyword for Write, W (57H), the same as the Write command.
- Next a two-byte field of ASCII characters (dd) representing a hex number is sent that corresponds to the controlled device number (00H-17H).
- Next three ASCII characters (rrr) are transmitted representing the hex number of the register whose contents are to be read (000H-FFFH).
- Next up to 16 ASCII characters may be sent. The controlling device will treat these characters as raw data.
- A carriage return ends the message.

DPM CPL Protocol

Introduction

This section describes the Control Point Language (CPL) protocol that can be used between a Kalypso or Zodiac Digital Production switcher (“switcher”) and an external digital picture manipulator (“DPM”).

The interface supports effect recall and run (including the switcher’s run lever arm), control of switcher aux buses by the DPM when used as input selectors, and on air tally involving the DPM.

General Background about Switcher/DPM Integration

The switcher/DPM integration involves the following functions:

- Effects on both the switcher and the DPM can be triggered and run under user control from the switcher panel. The DPM effect number can be any number in the range of DPM effects, and the number (not the effect itself) is stored in the switcher’s E-MEM register when a LEARN operation is done at the switcher. When a register is recalled at the switcher, this effect number will be sent to the DPM for effect recall.
- DPM source selection can be provided by the switcher's aux buses. One aux bus pair can be assigned to each DPM channel in a multi-channel configuration. Switcher aux buses support field accurate front/back source switching for the DPM.

Note Sources changes are synchronized to vertical interval by the switcher. Therefore this interface does not support front/back switching for nonplanar effects such as page turns.

Note The switcher relies on the key half of an aux bus pair to be connected to the same transforming channel as its video half. This interface makes no provision for use of the key side of a switcher's aux bus to feed a second video channel. While basic functionality can be achieved in this mode, functions such as effects send, front/back switching and tally will not function properly. Grass Valley Group accepts no responsibility for interfaces which do not adhere to this constraint.

- The term “Tally,” as used here, means that the switcher can receive a message from the DPM indicating whether the DPM is on-screen or not, and the switcher can use this information to tally sources feeding the DPM, providing proper panel and camera tally.

Note The DPM informs the switcher whether or not its image is on screen without regard to whether the DPM itself is on air. The switcher computes the on-air tally based on the on/off screen information and whether or not the DPM output is on air.

- Distinguishing between the terms “source” and “input”: “Source mapping” provides a level of indirection between the switcher's physical inputs and a source selection panel button. Input 20 could be mapped to button 1. “Input” would then refer to the physical entity, and “source” to the button and to what is stored as part of the switcher's effect.

Source information exchanged with the DPM is the logical “source” form of the information that is used—not the physical input. The switcher translates this internally to a physical input.

Note The term “crosspoint” can be misleading, meaning both source and input at different times. In this documentation, the term crosspoint refers to the actual switching matrix used in routing signals.

Switcher Configuration

In order for this interface to function as expected, the switcher must be configured properly. Configuration details are covered in the separate *Installation and Service Manual* of the switcher model involved.

General Protocol Notes

The protocol used for this interface is designed around setting values and receiving status (values). A subscription mechanism provides for status updates when changes occur providing more timely delivery of information without constant and frequent polling.

The interface is not designed as a pure controlling device/controlled device relationship; both the switcher and DPM have responsibility for part of the communications and function more as peers: The switcher forwards source selection button presses to the DPM and orchestrates running effects; the DPM sends video, key, and front/back information to the switcher to control aux buses.

This protocol is derived from one used for internal communications between the Grass Valley Group Krystal™ Digital Picture Manipulator control panel and effect manager via an ethernet connection. Much of the message structure is to support communications between multiple panels and multiple effect managers but was retained for consistency. The Krystal panel to effect manager communications is not described in this document.

The format of messages is described in *Message Format on page 113*.

Message specifics are described in *Parameters on page 121* and *Other Command Specifics on page 122*.

Data communications specifics are described in *Transmission Media and Protocols on page 123*.

Communications examples are contained in *Examples on page 124*.

Message Timing

The timing of some messages is critical in order for the switcher and DPM to perform in lock step. These are considered real time messages.

The run control messages sent by the switcher to the DPM must be acted on in a fixed number of video fields from when they are sent by the switcher. A delay adjustment is provided to compensate for variations in devices.

Source changes sent by the DPM to the switcher will be processed so that the video will change at the beginning of the next field.

All other messages do not have a deterministic requirement and are processed differently within the switcher.

Transmission of real time messages should be given priority over the non-deterministic ones. The maximum packet size was chosen to allow for interleaving high priority messages when several low priority messages are awaiting transmission.

Note The last packet containing real time messages sent by the DPM in a field must complete transmission 3 milliseconds before the end of the field. (13.6 mS from the start of the field for 525/60 or 22 mS for 625/50.) No real time messages should be sent to the switcher during this end-of-field period. Transmissions during this time can disturb critical processes within the switcher and cause the program video output of the switcher to glitch randomly.

Init/Online

This section describes the sequence used on start up communications. A DPM must respond and perform the indicated steps in order to achieve a reliable connection with the switcher.

If the **Switcher** is reset, or its DPM port configuration is changed:

1. The switcher sends out the message:
SEND_EVENT, EID_ONLINE (15)
This tells the DPM that the switcher just came on-line.
2. The DPM acknowledges this message with the RETURN_CODE message echoing the return ID from the switcher's SEND_EVENT message:
val = OK (0). (The switcher ignores the return value.)
3. The switcher then requests the DPM system name (by subscribing to it):
SUBSCRIBE, IMMEDIATE, PID_SYSTEM_NAME
4. The DPM sends its name:
ISSUE, PARAMETER_ARTICLE, PID_SYSTEM_NAME
5. Once the switcher receives the system name message (can be any name text) the switcher sends subscription messages for all of the parameters listed below Appendix B.

Note Other subscriptions may be solicited as well, but may be safely ignored. Their status is used only for Krystal and Kaleidoscope interfaces.

6. The DPM returns ISSUE messages containing the values for all of the subscribed parameters. From then on, any changes to any of these parameters will result in new issue messages being sent to the switcher.

Note The switcher does not poll for changes. It relies on the subscriptions registered during initialization to provide it with timely updates to parametric changes.

If the **DPM** is reset or its switcher interface is reconfigured, the sequence is:

1. The DPM sends out the message:
SEND_EVENT, event EID_ONLINE
This tells the switcher that the DPM just came on-line.
2. The switcher acknowledges receiving the on-line message by sending the RETURN_CODE message:
(the message header's return ID is set to 0)
val = EID_ONLINE (15).
then sends out a message cancelling all subscriptions:
UNSUBSCRIBE, CLEAR_SUBSCRIPTIONS
3. The switcher then takes over the initialization process with step 3 above.

Source Selection

There are multiple parts to the source selection part of the interface.

When the operator makes a selection on the aux row of the control panel (or via a remote aux panel), the switcher sends a message to the DPM with the video *or* key source, never both. The switcher relies on the DPM's use of source memory to select the appropriate key source. Essentially, the switcher is simply telling the DPM that an operator pressed a button. It is up to the DPM to process this information in an expected manner.

The switcher will send a selection for the front source, the back source or both depending on the near/far selection made from the switcher's AUX BUS menu.

The following SET commands communicate this information to the DPM. These messages include front/back information in the device ID field. The channel bit is set in the upper 16 bits. The lower 16 bits indicate if the front or back side is being set (1 = back). These fields are defined as bit maps so that multiple channels may be set at once. However in practice, a source set command applies to only one channel at a time.

PID_VIDEO_SOURCE

PID_KEY_SOURCE

Note The switcher does not make aux bus selections as a direct result of an aux row button press. If the DPM does not respond to the source message, the switcher's aux bus source will not change.

As an independent mechanism, the DPM sends messages with the video and key sources, and the front/back status per channel. They are separate SET messages, with the content of these parameters:

PID_FRONT_VIDEO_SOURCE_LIST (for 8 channels)

PID_BACK_VIDEO_SOURCE_LIST

PID_FRONT_KEY_SOURCE_LIST

PID_BACK_KEY_SOURCE_LIST

PID_FRONT_BACK_SELECT_LIST

The DPM must send source information on changes. Sending at regular intervals (e.g., every-field) is acceptable and can improve recovery should the link be interrupted.

Near/Far and Front/Back

The switcher differentiates between sides of a planar transform in two different, but related ways:

“Front” is the side of the picture plane which would be visible if no rotation is applied to the transformation; “back” is the opposite side.

“Near” is the side which is visible; “far” is the side which is not visible.

When no rotation is applied, Near = Front and Far = Back.

Use of near and far allow the operator to change sources and insure the change is either seen (near) or not seen (far) regardless of the picture’s rotation amount.

In order for the switcher to sort out near and far, it must be provided with front/back information from the DPM. Front/Back information tells the switcher which side is visible, that is, which is the near side.

Selections in the switcher’s AUX BUS menu allow delegation of the aux selector row to either near or far or both. The row will change and display the selected side. If BOTH is selected, the near side is displayed and a selection will cause a change in both the front and back sides.

The AUX BUS menu displays both the DPM’s front and back sources, the current aux selector row delegation (near/far) and which side the DPM has reported as the near side (front/back).

If the DPM does not fully implement the front and back source selection and front/back flag, the switcher will not be able to properly resolve near/far. While the system will still function, the information displayed to the user may be incorrect.

Source Hold

On Kalypso systems, holds can be imposed to prevent source selections from changing as effects are recalled or run. Two ways to do this involve selecting the **Hold** button to the right of the source buttons, or holding down a source button on a bus. Aux buses assigned to a DPM are actually controlled by the DPM, not the Kalypso system. In order to provide the same level of hold control for these aux buses, the Kalypso system takes control of an aux bus back from the DPM when that bus has a hold imposed upon it. When the bus is being held by either mechanism, the Kalypso system makes the source selection and ignores any source changes for that bus sent by the DPM. The DPM is still informed of source selections made from the Kalypso system via the normal mechanisms.

Tally

The switcher subscribes for PID_VIDEO_ON_AIR and PID_KEY_ON_AIR information from the DPM. The DPM must compute whether a channel’s image is on screen or not and update these subscriptions when necessary.

Note The parameter ID used for on screen information is the same as the one used to communicate on air information from the switcher to the DPM. This can easily lead to confusion since the same PID is used for different meanings.

In determining on/off screen, the following information should take into consideration:

- location relative to the output raster of the DPM (e.g., off screen)
- image size (e.g., reduced to zero size)
- rotation (e.g., exactly on edge)
- transparency
- any other processing or transform which may affect visibility

Additional factors include actual use of the incoming signal.

If the video signal is replaced by a matte fill, its incoming video is not on screen.

If the key input is not used it is not on screen.

If instead, the video signal is used in a self key, then the video is on screen even if the video is a matte.

Other factors may exist depending on the functionality of the DPM.

In addition, the DPM may subscribe to the DPM's on air status. This information tells the DPM whether or not the switcher has the DPM video or key on air. The DPM should *not* incorporate this information in the on screen status it sends to the switcher.

Activity Check

The activity check provides a mechanism for the switcher to detect a link failure (disconnected cable, etc.) or a patching/routing a different DPM to the switcher.

The switcher periodically tests for link activity by sending out a Subscribe Immediate command for the DPM's ethernet address. If no response is received, the switcher notes the time-out. When a response is received after a timeout, the switcher assumes that subscription issues or other updates could have been missed and reestablishes communications as if the switcher had been reset.

If link activity is present, the switcher looks at the value returned from the DPM. If it is different than the last response, the switcher assumes a different DPM is now connected and reestablishes communications as if it had been reset.

The value which the DPM returns is unimportant. However, if multiple DPMs exist in a facility it is a good idea to report different values from each one to allow the switcher to detect a change in DPM connections.

The activity check is optional. The interface behaves normally even if the DPM never responds to the switcher's query. However, the switcher will not be able to recover from link failures, patching or routing without a reset from one of the two devices.

The switcher queries the DPM approximately every 20 seconds.

Running Effects

During initialization, the switcher subscribes for the DPM's current effect number. When a switcher effect is learned, the DPM's current effect number is stored as part of the effect.

When that effect is recalled at the switcher (DPM 1 level enabled), the switcher will send out a command to recall the DPM effect. (SET, PID_CURRENT_EFFECT).

At the time of the recall, the switcher also issues a SET, PID_EFFECT_POSITION. The time value sent is zero if REVERSE on the switcher E-MEM effect memory system is turned off; or to the ending time of the switcher's effect if REVERSE is on.

As an effect is running, the switcher sends out commands every field to the DPM to SET the effect position (PID_EFFECT_POSITION). This same mechanism supports runs via the RUN button as well as the switcher's run lever arm. Other time positioning controls on the switcher panel (PREV KF, NEXT KF, REWIND, etc. also result in the sending of a new effect position to the DPM. In this way, the DPM exactly follows the timeline of the switcher's effect.

Subscription

Status information is gathered indirectly as a result of a "subscription" request. The request tells the other device where to send changes to the specified parameter. The changes are delivered to the requestor in an "issue" message which may contain several parameter "articles". Many articles may be contained in one subscription. Consider a magazine subscription: The reader subscribes to the magazine and whenever new information is available, a new issue is delivered to the readers home. The reader does not have to specifically request each new issue of the magazine. The "subscribe immediate" mechanism is like requesting a trial magazine issue - you only get one. Subscriptions may be canceled at any time. The requestor can also modify the terms of delivery at will.

The switcher subscribes to information using several request IDs. It is important for the DPM to include the proper request ID in the status update issue messages. Different parameter articles may be sent in the same issue as long as the subscriptions for all the data had the same request ID. Articles destined for different request IDs must be placed in separate issue messages.

It is also important to limit issue messages to less than 256 bytes or the message will be discarded by the switcher.

Message Format

A binary protocol with variable-length messages is used. Multiple messages may be delivered in a single packet. To accommodate a packet that may contain multiple variable-length messages, each message is preceded by a message header. As shown in [Figure 9](#), the message header is comprised of five fields.

The Version field is used to identify the protocol version and allows changes in the protocol while providing compatibility with older implementations. Currently, only version 1 has been specified.

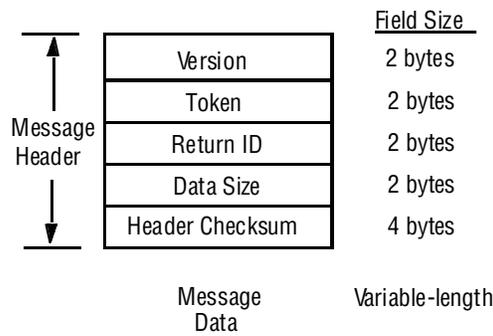
The Token field identifies the message.

The Return ID is a unique command identifier, that is used in conjunction with the Return Code message. If a control point desires to know the result of issuing a command, it places a non-zero value in the Return ID field. The Return ID and the command return code are sent back to the control point, in the form of a Return Code status message. The sender should manage the return ID's value to avoid receiving multiple return code messages with the same return ID.

Note The switcher violates this utilization of the return ID.
 The switcher does not respond with Return Code messages even if the return ID is nonzero.
 The switcher always responds to a Send Event On-line messages with a Return Code message even if the return ID is zero. It inserts a zero return ID into that Return Code message and sets the data value to EID_ONLINE (= 15) instead of OK or FAILURE.
 The switcher does specify a nonzero return ID in its Send Event On-line message and expects the DPM to echo that return ID in a Return Code message, however the switcher ignores the data value.

Only commands may request a Return Code message via the Return ID field. Status messages must insert a null return ID.

Figure 9. Message Format



8063_07_09_r0

The Data Size field specifies the size of the Message Data in bytes. It is this field which allows variable-length messages to be passed. The Header Checksum provides the mechanism by which message integrity and alignment can be verified. This is extremely important when parsing packets with multiple messages. The algorithm used with version 1 of the protocol is a simple summation of the message header information. This algorithm is shown below.

$$\text{Header Checksum} = \text{Version} + \text{Token} + \text{Return ID} + \text{Data Size}$$

All messages are network-byte ordered (i.e. most significant byte transmitted first) with a maximum packet size of 256 bytes. All messages will fit within this packet size, although care must be taken to insure subscription response issues are packed based on this maximum.

When constructing packets (HDLC frames) which contain multiple messages, each message must be padded to guarantee 32-bit alignment.

Message Tokens

The values assigned to each message token are enumerated in [Table 105](#). Other message values are treated as an error by the switcher.

Table 105. Message Tokens

Token	Value
Send Event	5
Set	6
Subscribe	8
Unsubscribe	10
Issue	16
Return Code	17

The following sections identify the messages used for the switcher/DPM interface. It provides a brief overview of each message, as well as details the contents of each message.

Set

The Set command is used to set a parameter to a specific value. As shown on the following page, the Set command data consists of four fields.

The parameter ID (“PID”) is used to uniquely identify the parameter which is to be set.

The flags field specifies how a parameter is to be set. A zero in this field indicates that an absolute set is to be performed (i.e. the parameter is set to the specified value) and is the only valid value for this parameter.

The device field contains a four byte identifier that further qualifies the parameter to be set. Typically, the device field identifies which channel is to be modified.

Table 106. Set Command Data

Field	Size	Value
Parameter ID	2 bytes	refer to Appendix B
Flags	2 bytes	0 = Absolute
Device	4 bytes	refer to Appendix B
Value	variable	refer to Appendix B

Refer to *Parameters on page 121* for a complete parameter list that includes each parameter ID and parameter type (short, long, float, string, etc.), as well as the devices associated with each parameter.

Subscribe

The subscribe command is used to solicit information from a device. This information is delivered in the form of a subscription issue¹. A subscription issue is sent after receiving a subscription request and thereafter when the requested information changes. As shown below, the Subscribe command data consists of a request address, a request ID, a serial number, a service ID, subscribe flags, and optional service specific data.

Table 107. Subscribe Command Data

Field	Size	Value
Request Address	16 bytes	(not used)
Request ID	2 bytes	any (requestor's choice) echoed in Issue message
Serial Number	2 bytes	any (requestor's choice)
Service ID	2 bytes	3 = Parameter Service
Flags	2 bytes	0 = normal service 1 = Subscribe Immediate 2 = Replace Service 3 = combination of 1 & 2
Service Data	variable	refer to Parameter Service below

¹ Refer to *Issue on page 117* for a complete description of a subscription issue.

The request address field specifies where the requested information is to be delivered. This field is only used when subscribing to information via Krystal's ethernet interface and is unused for HDLC communications.

The request ID is a user-defined subscription request identifier. It is used in conjunction with the request address to uniquely identify a subscription. Because the request ID is sent back as part of the subscription issue, it is commonly used for subscription issue routing within a control point. The serial number is also sent back to a control point as part of a subscription issue. A control point can use this field to qualify a subscription issue. For example, when a control point changes a subscription, it should modify the serial number from its previous state. If a subscription issue is received that contains the old serial number, then that issue should be discarded.

The service ID identifies the desired subscription service. Only parameter service is supported for the switcher/DPM interface. The parameter service is used to obtain and/or monitor the present value of one or more parameters.

The flags field modifies the behavior of the Subscribe command. When the subscribe immediate flag is set, the requested information is returned immediately. In essence, the information is returned once upon request without further updates. The replace service flag is used to replace a current service subscription (identified by request address, request ID, and service ID) with a new request. For example, one could replace an existing parameter service subscription with a new group of parameters.

Note Subscribe immediate results in only one issue with the current value(s). A subscription for changes is not activated.

Parameter Service

Table 108. Parameter Service Data

Field	Size	Value
Parameter ID	2 bytes	refer to Appendix B
alignment pad	2 bytes	any
Device	4 bytes	refer to Appendix B

The parameter ID field identifies the parameter of interest¹. This is followed by a pad field which 32-bit aligns the Device field. The Device field further qualifies the parameter or group of parameters being requested. In some instances, the device field contains a mask which may be used to request the parameter value of each device specified in the mask. For other parameters, a separate parameter service request is necessary².

Unsubscribe

The unsubscribe command cancels all or part of a subscription. As one might expect, the same command data necessary to subscribe to a service,

¹ Refer to *Parameters on page 121* for a complete list of parameters used for this interface.

² Refer to *Parameters on page 121* for the device types associated with each parameter.

is also needed to unsubscribe from it. The primary difference between the Subscribe and Unsubscribe command data is the value that the flags field may assume. As shown below, the Unsubscribe command flags may be set to one of four values.

Table 109. Unsubscribe Command Data:

Field	Size	Value
Request Address	16 bytes	(not used)
Request ID	2 bytes	any (requestor's choice) echoed in Issue message
Serial Number	2 bytes	any (requestor's choice)
Service ID	2 bytes	3 = Parameter Service
Flags	2 bytes	1 = Cancel Service 2 = Cancel Subscription 4 = Clear Subscriptions 8 = Clear All Subscriptions

With no flags set, the Unsubscribe command cancels the specified service information. For example, when unsubscribing from a select group of parameters, only those parameters identified in the parameter service data will be cancelled. When the cancel service flag is set, the subscription associated with the specified request address, request ID, and service ID is cancelled. It should be noted that a control point can have more than one subscription to the same service. This can be accomplished by using a different request ID with each service request. The cancel service flag will not cancel all subscriptions to the specified service, but only the one associated with the given request address and request ID. The cancel subscription flag is used to cancel all subscription services associated with the specified request address and request ID. The clear subscriptions flag is used to clear all subscriptions associated with a control point. Finally, the clear all subscriptions flag is used to clear out all control point subscriptions, including those for other control points. If any one of the Unsubscribe flags is set, the service specific data is ignored.

Issue

An Issue status message is sent to a control point as the result of a subscription request (a subscribe command) or a change in the value of a parameter with an active subscription. When a request is received, an issue is delivered to the subscribing control point. Another issue is not delivered until

the requested information changes. As shown on the following page, there are five fields of data associated with the Issue status message.

Table 110. Issue Message Data:

Field	Size	Value
Request ID	2 bytes	matches Request ID of Subscribe message
Serial Number	2 bytes	any (requestor's choice)
Service ID	2 bytes	3 = Parameter Article
alignment pad	2 bytes	any
Service Data	variable	refer to Parameter Article section below

The request ID is a user defined request identifier. It is passed in the data portion of the Subscribe command and is sent back to a control point as part of the Issue. The request ID is typically used by a control point to route issue data to the correct handler.

The serial number field provides the mechanism by which unwanted issue data can be ignored¹. This field is followed by the service ID field. The service ID specifies the type of issue. Only parameter issues are supported for the switcher/DPM interface.

The service ID is followed by a pad field which 32-bit aligns the service data. The service data itself contains service specific information.

Parameter Article

The parameter subscription service data consists of one or more parameter articles. Each article contains the value of one parameter. Multiple articles may be included in the same issue as long as the request address and request ID for all articles are the same. Remember that the request address and ID are provided by the requesting control point and must be returned with the issue.

The data associated with each parameter article is shown below.

Table 111. Parameter Article Data

Field	Size	Value
Parameter ID	2 bytes	refer to Appendix B
alignment pad	2 bytes	any
Device	4 bytes	refer to Appendix B
Parameter value	variable	refer to Appendix B

The parameter ID field identifies which parameter is contained in the article. The pad field 32-bit aligns the device field. The device field itself identifies the device associated with the specified parameter. The last field contains the parameter value. In order to accommodate parameters of varying size, the value field is variable length. When parameter service

¹ Refer to the Subscribe command for more information.

issue contains more than one parameter article, each article is 32-bit aligned within the packet.¹

To maximize link bandwidth, articles should be combined into a single issue whenever possible. However, this practice should not delay delivery of an article. That is, if no more articles require delivery, send a partially filled issue.

When a large number of subscriptions are active and many parameters are simultaneously changing value, it is easy to exceed the maximum packet size. When adding an article to an existing issue would exceed the maximum packet size, the article must be added to a new issue message.

Send Event

This command sends an event to the other device. An event is an input, without lasting state, which is used to trigger some action. As shown below, there are three Send Event command data fields.

The event ID (“EID”) field uniquely identifies the event.

The pad field 32-bit aligns the device field.

The last field contains a four-byte device identifier which further qualifies the event to be triggered.

The only Send Event used for the switcher/DPM interface is to instruct the other device to activate communications.

Table 112. Send Event Command Data

Field	Size	Value
Event ID	2 bytes	15 = On Line
alignment pad	2 bytes	any
Device	4 bytes	any
Value	variable	any

¹ Refer to [Parameters on page 121](#) for more information concerning parameters.

Note See the points listed below:

- The switcher violates this utilization of the return ID.
- The switcher does not respond with Return Code messages even if the return Id in the command sent by the DPM is nonzero.
- The switcher always responds to a Send Event On-line messages with a Return Code message even if the return ID is zero. It inserts a zero return ID into that Return Code message and sets the data value to EID_ONLINE (= 15) instead of SUCCESS or FAILURE.
- The switcher does specify a nonzero return ID in its Send Event On-line message and expects the DPM to echo that return ID in a Return Code message, however the switcher ignores the data value.

Return Code

The Return Code message returns the result of a command. All commands return an OK or ERROR status. A return code message is sent to a control point only if the return ID field, in the message header, contains a non-zero value.

As shown below, there is one field of data associated with the Return Code message. This field contains the return code itself.

Also, the return ID sent with the command is returned in the message header of the Return Code message. This is particularly important if a control point is sending multiple commands and needs the return ID to match a Return Code message to its command.

Table 113. Return Code Message Data

Field	Size	Value
Return code value	4 bytes	0 = OK / Success 0xFFFFFFFF =Error / Failure

Note Refer to the note under Send Event above.

Parameters

This appendix outlines those parameters supported for the switcher/DPM interface. A parameter is a value that may be modified and has lasting state. Parameters are sent via the SET command. When building a Set command, the parameter ID, its associated device, and the parameter value must be specified.

In addition, a parameter's value may be obtained via the subscription/issue mechanism.

[Table 114 on page 122](#) is a complete list of the parameters in tabular form. This list includes the name of each parameter used in this document, the PID's numeric value, its associated devices, and the type of each parameter with its acceptable range. Bytes and chars are 8-bit values, shorts are 16-bit values, longs are 32-bit values. "s15.16" is a signed 32-bit fixed point value where 1.0 = 0x00010000 (0.0 = 0x00000000, -1.0 = 0xFFFF0000, 0.5 = 0x00008000, etc.). Boolean is a 16-bit value where true or on = 1 and false or off = 0. Values are in decimal unless preceded by 0x indicating hexadecimal. The last two columns indicate who sends and receives which commands associated with each PID. "Tx" means the device transmits the message; "Rx" means that device receives the message. "SUB" indicates subscribe append or replace messages. "SUB-I" indicates subscribe immediate messages.

Table 114. Switcher/DPM Parameters

Parameter Name	ID	Device	Type/Range	Switcher	DPM
PID_FRONT_VIDEO_SOURCE_LIST	810	Channel bit map: 0x00000001 = Chan 1 0x00000002 = Chan 2 0x00000004 = Chan 3 0x00000008 = Chan 4 0x00000010 = Chan 5 0x00000020 = Chan 6 0x00000040 = Chan 7 0x00000080 = Chan 8 (typically ORed to address several channels at once)	8 bytes Chan 1 first 1 - 48 all 8 bytes must be sent even if bit not set in device map	Rx SET	Tx SET
PID_VIDEO_BACK_SOURCE_LIST	813				
PID_FRONT_KEY_SOURCE_LIST	811				
PID_BACK_KEY_SOURCE_LIST	814				
PID_FRONT_BACK_SELECT_LIST	816		8 bytes (as above) Chan 1 first 0 = front 1 = back		
PID_VIDEO_SOURCE	2642	Channel bit map: 0x00010000 = Chan 1 0x00020000 = Chan 2 0x00040000 = Chan 3 0x00080000 = Chan 4 0x00100000 = Chan 5 0x00200000 = Chan 6 0x00400000 = Chan 7 0x00800000 = Chan 8 ORed with Back side bit map: 0x00000001 = Chan 1 0x00000002 = Chan 2 0x00000004 = Chan 3 0x00000008 = Chan 4 0x00000010 = Chan 5 0x00000020 = Chan 6 0x00000040 = Chan 7 0x00000080 = Chan 8 (refer to text)	short 1 - 48	Tx SET	Rx SET
PID_KEY_SOURCE	2643				
PID_VIDEO_ON_SCREEN	819	Channel bit map (refer to source list parameters above)	1 boolean (applies to all channels specified in device field)	Tx SUB	Rx SUB
PID_KEY_ON_SCREEN	820			Rx ISSUE	Tx ISSUE
PID_VIDEO_ON_AIR (optional)	819			Rx SUB	Tx SUB
PID_KEY_ON_AIR (optional)	820			Tx ISSUE	Rx ISSUE
PID_CURRENT_EFFECT	886	0 = Effect Manager	short any value	Tx SUB Rx ISSUE Tx SET	Rx SUB Tx ISSUE Rx SET
PID_EFFECT_POSITION	875	0 = Effect Manager	s15.16 in frames	Tx SET	Rx SET
PID_SYSTEM_NAME	917	0 = Effect Manager	exactly 16 chars no terminating null pad with spaces	Tx SUB Rx ISSUE Rx SUB-I Tx ISSUE	Rx SUB Tx ISSUE optional: Tx SUB-I Rx ISSUE
PID_ETHERNET_ADDRESS (optional - refer to Activity Check description)	3047	0 = Effect Manager	long (refer to text)	Tx SUB-I Rx ISSUE	Rx SUB-I Tx ISSUE

Other Command Specifics

The bulk of the communications used for the switcher/DPM interface is via parametric interchange, as indicated in Appendix B. This section tabulates other commands used by the switcher/DPM interface.

Currently, the only other communications is the exchange of an on line event while communications are initiated. The DPM must transmit an on-

line event and respond to one sent by the switcher with an appropriate return code message.

Transmission Media and Protocols

The switcher/DPM interface is designed as a point-to-point link. Messages are delivered in an HDLC-like frame consisting of an 8-bit opening flag, a variable-length data field, a 16-bit CRC, and an 8-bit closing flag. Within the transmission frame, messages are delivered via the data field.

A 307.2K baud FM0 synchronous serial link with imbedded clock is used for deterministic communications. Data rate and frame format are not adjustable. HDLC is a synchronous protocol with built-in mechanisms for error detection and packet detection.

Not all UARTs support this protocol. Those that do handle the packetization, zero stuffing and CRC insertion/detection. A Motorola 68302 along with a 68195 are used in the switcher for this interface. A Motorola 68360 or Zilog 8030 or 8530 also support this protocol.

Unlike asynchronous protocols, communications is frame (packet) based. A transmit underrun is not permitted in a synchronous protocol and will abort the transmission of the frame and the receiver will discard it.

The physical connection between the switcher and the digital effect device (DPM) is point to point with two RS-422 pairs, one for each direction.

The connector provided for this interface is a female DB-9. It is located on the rear of the switcher frame and labeled "A1". The pinout is shown below.

Table 115. Return Code Message Data

Function	Pin
Tx + (switcher to DPM)	3
Tx -	8
Rx + (DPM to switcher)	7
Rx -	2
Ground	1, 4, 6, 9

Examples

The following are examples of switcher messages to and from the DPM. All values are in hex. The semi-colon after 2 or more bytes is just used as a separator to give the name of that field and any other explanatory information. (Values are in hexadecimal)

Example 1

DPM -> switcher, RETURN_CODE:

```
00 01; Version
00 11; Msg token, RETURN_CODE
00 00; Return Id
00 04; Data Size
00 00 00 16; Header checksum
00 00 00 0F; Return code
```

Example 2

Switcher -> DPM, UNSUBSCRIBE

```
00 01; Version
00 0a; Msg token, UNSUBSCRIBE
00 00; Return Id
00 18; Data Size
00 00 00 23; Header checksum
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 16 bytes rqstAddr
00 00; Rqst Id
00 00; Serial number
00 03; Service Id, PARAMETER_SERVICE
00 04; flags, CLEAR_SUBSCRIPTIONS
```

Example 3

Switcher -> DPM, SUBSCRIBE

```
00 01; Version
00 08; Msg token, SUBSCRIBE
00 00; Return Id
00 20; Data Size
00 00 00 29; Header checksum
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 16 bytes rqstAddr
00 00; Rqst Id
00 00; Serial number
00 03; Service Id, PARAMETER_SERVICE
00 02; flags, REPLACE_SERVICE
03 95; Param Id, PID_SYSTEM_NAME
00 00; padding
00 00 00 00; Device, EFFECT_MANAGER
```

Example 4

DPM -> Switcher, SET front video source

```

00 01; Version
00 06; Msg token, SET
00 00; Return Id
00 10; Data Size
00 00 00 17; Header checksum
03 2A; Param Id, PID_FRONT_VIDEO_SOURCE_LIST
00 00; flags, ABSOLUTE
00 00 00 05; Device, logical channel 1 and logical channel 3
07 00 09 00 00 00 00 00; 8 channels worth of source information;
; only channel 1 and channel 3 are relevant.

```

Example 5

DPM -> Switcher, SET front key source

```

00 01; Version
00 06; Msg token, SET
00 00; Return Id
00 10; Data Size
00 00 00 17;Header checksum
03 2B; Param Id, PID_FRONT_KEY_SOURCE_LIST
00 00; flags, ABSOLUTE
00 00 00 05; Device, logical channel 1 and logical channel 3
0F 00 09 00 00 00 00 00; 8 channels worth of source information;
; only channel 1 and channel 3 are relevant.

```

Example 6

DPM -> Switcher, SET front/back select

```

00 01; Version
00 06; Msg token, SET
00 00; Return Id
00 10; Data Size
00 00 00 17; Header checksum
03 30; Param Id, PID_FRONT_BACK_SELECT_LIST
00 00; flags, ABSOLUTE
00 00 00 05; Device, logical channel 1 and logical channel 3
00 00 01 00 00 00 00 00; 8 channels worth of source information;
; only channel 1 and channel 3 are relevant.
; 00 = front visible
: 01 - back visible

```

Example 7

DPM -> Switcher, ISSUE

```
00 01; Version
00 10; Msg token, ISSUE
00 00; Return Id
00 20; Data Size
00 00 00 31; Header checksum
00 00; Rqst Id
00 00; Serial Number
00 03; Service Id, PARAMETER_SERVICE
00 00; padding
03 95; Param Id, PID_SYSTEM_NAME
00 00; padding
00 00 00 00; Device, EFFECT_MANAGER
    16-byte system name in ASCII
```

Example 8

Switcher -> DPM, SUBSCRIBE to more than one Param Id in one message

```
00 01; Version
00 08; Msg token, SUBSCRIBE
00 00; Return Id
00 28; Data Size
00 00 00 31; Header checksum
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 16 bytes rqstAddr
00 01; Rqst Id
00 00; Serial number
00 03; Service Id, PARAMETER_SERVICE
00 00; flags, APPEND_SERVICE
03 33; Param Id, PID_VIDEO_ON_AIR
00 00; padding
00 00 00 FF; Device, 8 channels
03 34; Param Id, PID_KEY_ON_AIR
00 00; padding
00 00 00 FF; Device, 8 channels
```

Example 9

Switcher -> DPM, SET a new video source

```
00 01; Version
00 06; Msg token, SET
00 00; Return Id
00 0C; Data Size
00 00 00 13; Header checksum
0A 52; Param Id, PID_VIDEO_SOURCE
00 00; flags, ABSOLUTE
00 01 00 01; Device, logical channel 1, visible side (source memory
    device uses mask to indicate side)
00 0B; Source 11
00 00; padding
```


Router Protocol

Introduction

The Kalypso Video Production Center and the Zodiac Digital Production Switcher can be connected with a Grass Valley Group SMS 7000 routing system for control of the router. This section gives an overview of this interface. Commands used by this protocol for this purpose will be identified but not described in detail. A complete description of the protocol and all the available commands is available in the separate *Routing Products Protocol Manual*.

A Kalypso system can interface with a router to accomplish two different and completely separate functions:

- Router Control, consisting of user initiated Panel, Menu, and R-MEM control of router source selections for specific router destinations.
- Emergency Bypass option operation which involves source selection for specific router destinations.

A Zodiac system supports Router Control, as listed above, but does not support Emergency Bypass.

Native Protocol

Both router features interface to a Grass Valley Group SMS-7000 routing system using Native Protocol. Native protocol supports Ethernet and serial physical connections. The Router Control feature employs Ethernet, and the Emergency Bypass option employs RS-422 serial connections.

Commands

The following Native Protocol commands are used for Router Control and the Emergency Bypass option. See the separate *Routing Products Protocol Manual* for details about using these and other commands.

Table 116. Native Protocol Commands

Name	Function
BK I	set background refresh interval
QN S	query source name
QN D	query destination name
QN L	query level
QD name	query destination status
TD name name	take
PR name level	protect
UP name level	unprotect
BK E On	turn level 4 acknowledge on

Still Store Image File Format

Introduction

This document describes the file formats used by the Kalypso Classic and Zodiac system still stores, and by the Kalypso HD/Duo system still stores. Experienced developers should be able to utilize this information to write applications that convert images and animations to and from the Kalypso and Zodiac production switcher systems.

Still Store Hardware

The Kalypso Classic and Zodiac systems support SD formats (525 and 625 lines). The still store hardware supplied with these systems supports only those two formats.

The Kalypso HD system since software version 10.0 supports either HD formats (1080i and 720p) or SD formats (525 and 625 lines). The Kalypso Duo system uses the same hardware, but is restricted to operating in only the SD modes. The Kalypso HD and Duo systems store and retrieve SD still store images of 525 and 625 lines in the same format as documented for the Kalypso Classic and Zodiac still store.

Image Storage Mechanism

Kalypso Classic and Zodiac system still store video (and key) data conforms to CCIR 601 specifications, but is packed in a particular order that matches the still store hardware memory organization. While many graphics applications can create CCIR 601 video data, none pack data in the order expected by the still store.

The Kalypso HD system still store video (and key) data conforms to SMPTE 274M (1920x1080, including 1080I@ 30Hz, 29.9Hz, 25Hz) and SMPTE 296M (1280x720, including 720P @ 59.9Hz and 60Hz). The above considerations regarding data packing order also applies to HD images.

Kalypso and Zodiak still store systems use a pair of files to describe an image rather than embedding information about the image into a single file. Since FTP is the primary interface to the still store, it is not possible to transfer part of a file. A separate “metadata” file provides rapid access to information about the image without transferring the image itself.

Still store “images” refer to one of the following:

- Video only still
- Video plus key still
- Video only animation
- Video plus key animation

In many cases, the still store supplies small, animated icons or bugs. To increase efficiency, images do not have to be a full television frame. During the capture process, the image can be “fenced” to a $1/2 \times 1/2$, $1/3 \times 1/3$, or $1/5 \times 1/5$ of the full size image. Only the fenced region is written to the file.

On Zodiak systems, the animation and fencing features were implemented in Zodiak Release 3 software. Earlier Zodiak software versions supported only full size still images.

Each image has an identifier handle. This handle must be unique within a directory from which the still store loads images. A DOS standard 8.3 file name and extension express the image handle and bind the data and metadata files together. A separate image name (stored in the metadata file) is used for display purposes.

Image files names have a specific format:

- Metadata files are of the form I000nnnn.gva
- Data files are of the form I000nnnn.gvi

The first four characters are the letter I followed by three zeros.

The last four characters represent the image handle: a number between 0001 and 2000 (0001 – 0050 for Zodiak version 2). Leading zeros must be included. The still store matches image and metadata files according to their file names. If the information contained in the file pair disagree, the image will not load properly. The metadata file has space for the image handle, but this field is unused. The image handle contained in the file name is sufficient to identify the image.

In order to create a still store image, the application must be able to convert the graphics data into CCIR 601, SMPTE 274M, or SMPTE 296M format, pack it in the proper order and write it to the data file. It also must generate a metadata file that accurately describes the contents of the matching data file. The metadata file of an existing image also holds the key to unpack the matching image data should the application wish to ingest an image created by the still store.

The values described in this document are shown as normally (big endian) ordered bytes, but are written to the image files in little endian order.

Metadata File Format Versions and Kalypso Software Releases

The image metadata file format has evolved since the still store was first released. This format is now at version 6, and only this metadata version is described in this API.

Kalypso Release 7 and Zodiac Release 2 are the earliest software releases that support the version 6 metadata format. Images created using this API will only be usable on Kalypso or Zodiac systems running these or higher releases of software. The Zodiac Release 2 software has not implemented the animation or fencing features, but these features are supported in Zodiac Release 3.

Images that use older format metadata will need to be resaved using a Kalypso or Zodiac system running software that supports the version 6 metadata format before an application based on this API will be able to access the images.

Metadata contents must reflect the content of the matching data file or the still store will have problems reading the image. In severe cases, such parsing errors could result in a still store or frame main processor crash.

Video Conversion

Most graphics applications produce “video” as red, green and blue components. An alpha channel is sometimes included to control transparency of the graphic in a final composite. Graphics systems do not necessarily generate images of the proper size for direct use by video systems. In addition, pixels in the video domain are not square. The still store also expects interlaced video. It may be necessary to crop, aspect ratio convert, interlace or otherwise manipulate the image spatially to conform to the video raster. The alpha channel should be treated as a black and white video signal and scaled accordingly.

The still store accepts video as Y, Cr and Cb. An optional “key” signal (a black to white signal) may be included. The key correlates to the alpha channel.

SD and HD images use different colorimetry. SD uses ITU-R BT 601 while HD uses BT 709.

For SD formats CCIR 601 defines the basic video signal levels used by the still store and the conversion between RGB and Y, Cr, Cb as:

- $Y = 0.299R + 0.587G + 0.114B$
- $Cr = 0.713(R - Y) = 0.500R - 0.419G - 0.081B$
- $Cb = 0.564(B - Y) = 0.500B - 0.169R - 0.331G$

For HD formats SMPTE 274M (1920 x 1080) and 296M (1280 x 720) specify:

- $Y = 0.2126R + 0.7152G + 0.0722B$
- $Cr = (0.5/(1 - 0.2126)) * (R - Y)$
- $Cb = (0.5/(1 - 0.0722)) * (B - Y)$

In addition, headroom is specified for overshoot and undershoot.

- Black is a Y code word of 64
- White is a Y code word of 940
- Cr and Cb sit on a pedestal of 512 to avoid negative numbers and has a range of ± 448 about this central value.

These factors need to be taken into account when scaling values.

Cr and Cb are half the bandwidth of Y. Proper filtering should be applied during RGB to Y, Cr, Cb conversion to avoid aliasing. Half the Cr and Cb samples are then discarded. Cr and Cb are both derived from the same sample (first, third, fifth, etc.).

In addition, code words below 4 and above 1019 are reserved for sync and control words. The Y, Cr, and Cb values must be limited accordingly.

While CCIR 601, SMPTE 274M and SMPTE 296M specifies pixels per line and the number of lines per field, SMPTE RP-187 does a better job of defining things. RP-187 specifies images in terms of a “clean aperture” (what the end image will be cropped to) and a “production aperture.” The still store stores the entire production aperture. However, cropping and positioning is done in “screen units” (also defined in RP-187) that are based on clean aperture. Note that screen units are different for 4:3 and 16:9 aspect ratios.

Shaped Video

Modern video production equipment usually supports two mechanisms of transferring video between equipment.

The most common approach used in all digital environments is known as “shaped video” because the video produced by one device has already been modified by its accompanying key. The receiving device “cuts a hole” in the background image and adds the incoming video to it without further processing.

When distribution paths were predominately analog, this approach did not work well. Black level errors and noise could corrupt the background portions of the final composite, so the video output by one device was divided by its own key signal (“unshaped”) so that the receiving device could reapply that key signal, eliminating the opportunity for signal errors to corrupt the background.

The still store accepts either “shaped” or “unshaped” video for video + key pairs. When a still store output is selected on a keyer, the still store notifies the keyer which mode to operate in. If the mode is set incorrectly, the composite will not look right.

Part of the metadata identifies if the video is shaped or if it needs to be shaped prior to compositing (unshaped). The application generating the image file is responsible for setting this flag correctly.

In general, the output of a graphics system that produces red, green, blue and alpha channels will be producing shaped video. A shaped video signal is black everywhere outside the alpha area. Inside the alpha area, it should reduce in amplitude as the image becomes more transparent.

SD Image Data File Format

The Kalypso and Zodiac SD still store image data format reflects the organization of the still store memory used to rasterize images. This is a raw data file without any headers or embedded information – just the pixels (and also in some cases “pad” bytes of “0” to fill the remaining portion of a line or a field). The still store expects a specific organization of information in this file based on the contents of the metadata file.

Six 10-bit video (or key) pixels (twelve values: Y plus half bandwidth Cb and Cr) are packed into four 32-bit values as shown in [Table 117](#):

Table 117. SD Still Store Image Data

DWORD 3	Y 6	Y 5	Y 4	00
DWORD 2	Y 3	Y 2	Y 1	00
DWORD 1	Cr 5	Cb 5	Cr 3	00
DWORD 0	Cb 3	Cr 1	Cb 1	00
	31	22 21	12 11	2 1 0

8063_04_08_10

Note that the sample numbers for Cb and Cr show only odd numbered samples. This is based on discarding half of the full bandwidth chrominance samples (the even numbered samples). If Cb and Cr have already been down sampled, do not discard samples.

All images must begin in field 1 of the frame and include both fields. Incomplete lines (half line blanking at the top or bottom of the screen) must be extended with active pixels (black is OK).

Cb and Cr words should be set to 512 (no color) for black and white video or key signals.

Where a key signal exists, a frame (two fields) of video is followed by a frame of key.

When the image is a video only animation, frames are stacked back to back. When the image is a video + key animation, video and key are interleaved. That is video field 1, video field 2, key field 1, key field 2, video field 3, etc.

[Table 118 on page 137](#) shows the amount of information written to the data file per frame:

Table 118. SD Format Data Amounts

Frame Rate	Fence Size	Pixels per Line	Bytes per Line	Lines		Pixel Groups per Line	Total Bytes per Frame
				Field 1	Field 2		
525/60	Full	720	1920	244	243	30	1,048,576
	1/4	360	960	122	121	15	234,240
	1/9	240	640	81	81	10	103,680
	1/25	144	384	48	48	6	36,864
625/50	Full	720	1920	288	288	30	1,179,648
	1/4	360	960	144	144	15	276,480
	1/9	240	640	96	96	10	122,880
	1/25	144	384	57	57	6	43,776

Note Values for Pixels per Line and Lines in Field1/Field2 are active video pixels only. Total bytes per frame includes pad bytes.

SD Full Size File Format / Size Calculations

Vertical and horizontal image start points are defined in CCIR 601. A full screen image in the Kalypso still store is equivalent to the production aperture defined in SMPTE RP-187.

Each line of a full screen image, consists of 720 pixels packed into 1920 bytes as described above, followed by 128 pad bytes so that each line occupies 0x800 bytes.

A 525/60 field 1 contains 244 lines. A 525/60 field 2 contains 243 lines. Both fields are padded such that each occupies 0x80000 bytes and a frame occupies 0x100000 bytes (1,048,576).

Both fields for a 625/50 image contain 288 lines. 625/50 fields are not padded with additional lines. Each field in a full raster 625/50 image is 0x90000 bytes so a frame is 0x120000 bytes (1,179,648).

Table 119 and Table 120 show file layout for 525/60 and 625/50 images.

Table 119. 525/60 Format

Field	Line	Start Address	End of Active Video Address	End of Line Address	
Field 1	1	0x0000	0x077f	0x07ff	
	2	0x0800	0x0f7f	0x0fff	
	3	0x1000	0x177f	0x17ff	
	4	0x1800	0x1f7f	0x1fff	
	...				
	242	0x78800	0x78f7f	0x78fff	
	243	0x79000	0x7977f	0x797ff	
	244 (last active line of field 1)	0x79800	0x79f7f	0x79fff	
unused		0x7a000	unused	0x7ffff	
Field 2	1	0x80000	0x8077f	0x807ff	
	2	0x80800	0x80f7f	0x80fff	
	3	0x81000	0x8177f	0x817ff	
	4	0x81800	0x81f7f	0x81fff	
	...				
	242	0xf8800	0xf8f7f	0xf8fff	
	243 (last active line of field 2)	0xf9000	0xf977f	0xf97ff	
	unused		0xf9800	unused	0xfffff

Table 120. 625/50 Format

Field	Line	Start Address	End of Active Video Address	End of Line Address	
Field 1	1	0x0000	0x077f	0x07ff	
	2	0x0800	0x0f7f	0x0fff	
	3	0x1000	0x177f	0x17ff	
	4	0x1800	0x1f7f	0x1fff	
	...				
	287	0x8f000	0x8f77f	0x8f7ff	
	288	0x8f800	0x8ff7f	0x8ffff	
Field 2	1	0x90000	0x9077f	0x907ff	
	2	0x90800	0x90f7f	0x90fff	
	3	0x91000	0x9177f	0x917ff	
	4	0x91800	0x91f7f	0x91fff	
	...				
	287	0x11f000	0x11f77f	0x11f7ff	
	288	0x11f800	0x11ff7f	0x11ffff	

SD Fenced Image File Size Calculations

A fenced region must start with a pixel containing co-sited color difference samples.

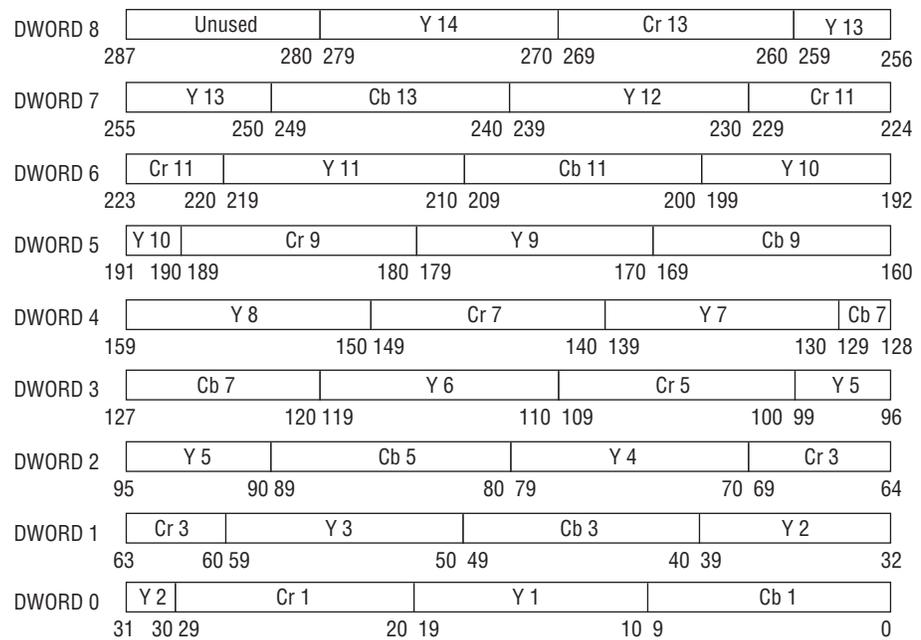
No padding occurs at the end of the line. The only padding occurs at the end of field 2 for a 525/60 1/4 size image. That field is padded to the same length as field 1 (122 lines) by adding one line.

Refer to [Table 118 on page 137](#) for specific fenced region sizes.

HD Image Data File Format

Fourteen 10-bit video (or key) pixels (28 values, 14 Y plus 14 half bandwidth Cb and Cr) are packed into a pixel group consisting of nine 32-bit values as shown in [Table 121](#).

Table 121. HD Still Store Image Data Pixel Group



8063_04_07_10

Table [Table 122](#) shows the amount of information written to the data file per frame.

Table 122. HD Format Data Amounts

Image Format	Fence Size	Pixels per Line	Bytes per Line	Lines per Frame (Field 1)	Lines per Frame (Field 2)	Pixel Groups Per Line	Total Bytes per Frame
1080i (Interlaced)	Full	1920	4968	540	540	138	5,365,440
	1 / 4	1008	2592	270	270	72	1,399,680
	1 / 9	672	1728	180	180	48	622,080
	1 / 25	392	1008	108	108	28	217,728
720p (progressive)	Full	1280	3312	720		92	2,384,640
	1 / 4	672	1728	360		48	622,080
	1 / 9	448	1152	240		32	276,480
	1 / 25	266	684	144		19	98,496

HD Full Size File Format / Size Calculations

Vertical and horizontal image start points are defined in SMPTE 274M and SMPTE 296M. A full screen image in the Kalypso HD still store is equivalent to the production aperture defined in SMPTE RP-187.

Each line of a full-screen 1080i image consists of 1920 active and 12 inactive pixels packed into 4968 bytes as described in tables [Table 121](#) and [Table 122](#). There are 137 2/14 active pixel groups (1920 pixels) and 12/14 of the last pixel group (12 pixels, which contain inactive video) per line (a pixel group contains 14 pixels). There are no padding lines between fields.

Each line of a full-screen 720p image consists of 1280 active and 8 inactive pixels packed into 3312 bytes as described in the tables. There are 91 6/14 active pixel groups (1280 pixels) and 8/14 of the last pixel group (8 pixels, which contains inactive video) per line. There are no padding lines at the end of the frame.

The pixel groups of all the fenced image sizes are fully packed.

Refer to [Table 123](#) and [Table 124](#) for layouts of 1080i and 720p formats.

Table 123. 1080i Format

Field	Line	Start Address	Address of Last Byte of Line
Field 1	1	0x0	0x1367
	2	0x1368	0x26cf
	3	0x26d0	0x3a37
	4	0x3a38	0x4d9f
	...		
	538	0x28b528	0x28c88f
	539	0x28c890	0x28dbf7
	540	0x28dbf8	0x28ef5f
Field 2	1	0x28ef60	0x2902c7
	...		
	540	0x51cb58	0x51debf

Table 124. 720p Format

Line	Start Address	Address of Last Byte of Line
1	0x0	0xcef
2	0xcf0	0x19df
3	0x19e0	0x26ef
4	0x26d0	0x33bf
...		
719	0x244920	0x24560f
720	0x245610	0x2462ff

HD Fenced Image File Size Calculations

A fenced region must start with a pixel containing co-sited color difference samples.

[Table 122 on page 139](#) shows the sizes of fenced images. Each group (of 14 pixels in 36 bytes) in each fenced size is completely filled, which is why a line at 1/4 fence size has more than 1/2 the number of pixels in a full-size image line.

Image Metadata File Format

The metadata file provides instructions necessary for the still store to read the image data file. It also contains instructions on how the image should be output from the still store (and how it should appear in the image browser). The metadata file format has evolved significantly since the first still store implementation in Kalypso. This format is now at version 6, and only this version is described here. See *Metadata File Format Versions and Kalypso Software Releases* [on page 133](#) for more information.

[Table 125](#) indicates the size offset, and contents of each data field.

- Size is in bits, and also indicates whether the parameter is signed or unsigned. E.g., Uint16 is a 16-bit unsigned value, Int32 is a signed 32-bit value. Float values are 32-bit values in IEEE 754 format (ANSI/IEE 754-1985).
- Offset is in bytes from the beginning of the file in hexadecimal.

Table 125. Image Metadata

Offset	Size	Parameter	Value /Comment
0x00	UInt16	magicNumber	0x5ADF
0x02	UInt16	metatataVersion	6 (don't continue processing if not version 6)
0x04	UInt32	imageHandle	0 (not used) See discussion above in Introduction.
0x08	81 characters	fileName[81]	0's (not used)
0x59	81 characters	imageName[81]	Null terminated character string. Characters after first null are ignored. See discussion above in Introduction.
0xAA	UInt16	imageType	3 for single frame video only 4 for single frame key only (not used) 5 for single frame video + key 6 for multiple frame video only 7 for multiple frame key only (not used) 8 for multiple frame video + key fill image data file with appropriate information
0xAC	UInt16	imageLineRate	1 for 525/60 2 for 625/50 $3 + 1 * 2^6 + [(line\ rate) * 2^8]$ for 720p $4 + [(line\ rate) * 2^8]$ for 1080i (line rate optional, see Line Rate discussion for_details)
0xAE	UInt16	pad1	0 (compiler generated structure alignment filler)
0xB0	Int32	aspectRatio16x9	0 for 4:3 (see Output Crop and Position discussion) 1 for 16:9 for supported HD line rates 720p and 1080i

Table 125. Image Metadata - (continued)

Offset	Size	Parameter	Value /Comment
0xB4	Int32	fenceSize	0 for full screen 1 for 1/2 x 1/2 2 for 1/3 x 1/3 3 for 1/5 x 1/5 Fill image data file with appropriate information. See discussion above in Image Data File Format section.
0xB8	float	outputCropRight	Controls cropping of image when loaded to the output of the from still store. See discussion below in Crops and Position section.
0xBC	float	outputCropLeft	
0xC0	float	outputCropTop	
0xC4	float	outputCropBottom	
0xC8	float	outputVertPos	Controls position of image when loaded to the output of the stillstore. See discussion below in Crop and Position section.
0xCC	float	outputHorizPos	
0xD0	Int32	outputFreezeMode	Controls how frozen image is displayed by the still store. 0 to display complete frame 1 to display field 1 with interpolated field 1 for field 2 2 to display field 2 with interpolated field 2 for field 1 3 to display field 1 in both fields (replicate) 4 to display field 2 in both fields (replicate)
0xD4	Int8	shapedVideo	0 for unshaped or video only 1 for shaped video See discussion above in Shaped Video section.
0xD5	Int8	pad2	0 (compiler generated structure alignment filler)
0xD6	Int16	clipLength	Number of frames stored in data file. Stills have a length of 1
0xD8	Int16	markInFrame	Playback starting frame number (0 is first frame in file)
0xDA	Int16	markOutFrame	Playback ending frame (number of the frame AFTER last one to be played)
0xDC	Int16	vidKeyOffset	Temporal offset for key in frames (not used). Fill with 0s.
0xDE	Int16	loopFromFrame	Number of last frame in a looped region.
0xE0	Int16	loopToFrame	Number of first frame included in looped region.
0xE2	Int16	loopCount	Number of times looped region is repeated. 0 - no repeat (play looped region once) 1 - 32767 repeat loop region this number of times -1 loop forever A value of 1 says repeat the looped region once, i.e., play it twice.
0xE4	Int32	dwLowDateTime	Date and time of image modification in Win32 (MFC) format (not used). Insert Win32 formatted date and time or fill with 0s.
0xE8	Int32	dwHighDateTime	
0xEC	276 characters	pad3[276]	0s (for consistent 512 byte file size regardless of version)

Pad parameters pad1 and pad2 are fillers generated by the compiler to align the next parameter to a 16 or 32 bit boundary. These pads must be included in the file but their value is unimportant.

Metadata files are written as 512 bytes regardless of the size of the active information. Pad3 fills out the message to 512 bytes. Values are not important but dummy values must be included in the file.

Line Rate

For offset 0xac: imageLineRate, the 2⁶ bit, bit 6, is the “progressive scan” bit. For rates other than 525 and 625, the upper eight bits of the 16-bit imageLineRate field can optionally be used to express the original frame rate. The

still store does not use this line rate itself as it is unable to interpolate line rates if this rate differs from the rate that the system is currently set to. However this information is preserved for future use or for information purposes.

23.97 Hz or 24Hz would be represented by 24.

25 Hz would be represented by 25.

29.97 or 30.0 Hz would be represented by 30.

47.95 or 48 Hz would be represented by 48.

50 Hz would be represented by 50

59.94 or 60 Hz would be represented by 60.

Example: 720p at 60 Hz frame rate would be:

60 * 2⁸ 60 (decimal) in upper 8 bits

1 * 2⁶ Progressive Scan bit in bit 6

3 1280 x 720 image format

Example: 1080i at 29.97 Hz frame rate would be:

30 * 2⁸ 30 (decimal) in upper 8 bits

4 1920 x 1080 image format

Refer to [Table 126](#) for a representation of the bit arrangement.

Table 126. 0xAC Image Line Rate Detail

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Frame Rate								Reserved	Prog. Bit	Image Format					
24 = 24 or 23.97 Hz 30 = 30 or 29.97 Hz 48 = 48 or 47.95 Hz 50 = 50 Hz 60 = 60 or 59.94 Hz								0	0 = interlace 1 = progressive	1 = 525/60 2 = 625/50 3 = 1280 x 720 4 = 1920 x 1080					

Still Store Output Control

When images are created (especially when they are fenced or when an animation is looped), the author makes decisions about how the image should appear when output from the still store. A set of “output” parameters in the image metadata file convey these instructions to the still store. The switcher operator has the opportunity to change these settings if they don’t work for a specific situation, but the metadata values provide a starting point for those adjustments.

Output Crop and Position

Cropping determines what portion of the image is output by the still store. Positioning relocates the image relative to screen center. These parameters do not affect the contents of the image data file. They only affect how the still store presents the image when loaded to an output.

Values are in screen units relative to screen center. Screen units are based on the “clean aperture” as defined in SMPTE RP-187.

The still store captures and outputs a production aperture by default. The values shown in [Table 127](#) place the crop edges at the points defined by SMPTE as the production aperture, essentially disabling the crop.

Table 127.

Image Format	Aspect Ratio	Left Edge	Right Edge	Top Edge	Bottom Edge
525/60	4:3	-4.07	+4.07	+3.08	-3.08
	16:9	--	--	--	--
625/50	4:3	-4.17	+4.17	+3.05	-3.05
	16:9	--	--	--	--
1280 x 720 or 1920 x 1080	16:9	-16.272	16.272	9.1125	-9.1125

The specified values will set the crop exactly at the production aperture, but this is the middle of the shaped edge. For an uncropped image, crops need to be set outside these values.

Note 16:9 mode is not currently supported for SD formats. All SD images, regardless of their actual aspect ratio should use 4:3 screen unit numbers and set the aspectRatio16x9 parameter to 0.

Output Freeze Mode

When the still store is displaying a still or a stopped animation, the user can set the still store to display a complete frame or restrict output to one of the two fields. Frame output is ideal for stills since full vertical bandwidth is used. However, if the image contains motion (as would be the case for an animation or a still grabbed from moving video), frame output may have too much jitter. Selecting one of the field modes will eliminate the jitter at the expense of vertical bandwidth.

Field interpolate modes present the selected field during its display period and interpolate the other field by averaging adjacent lines from the selected field. Field replicate modes simply output the selected field during both field times.

Mark In/Out

When an animation is captured, extraneous material before or after the desired segment is often included. The mark in and out points instruct the still store to start and end playback at specific points. The frames prior to the in point and after the out point are not played.

Note that the still store actually plays up to the frame *before* the mark out frame number. The mark out frame is not played.

Loops

Animations often are repetitive in nature. That is, the entire animation (or a portion of it) repeats. In order to save storage space, the still store permits these repeated regions to be included only once in an animation file. Metadata instructions control where a loop repeat occurs and how many times that segment is repeated.

The loopFrom point is the last frame in a looped region. The loopTo point is the first frame in a looped region. The loopCount is the number of times the segment is repeated. When the animation begins playback, it starts at the mark in point and plays to the loop from point. If a repeat is enabled, playback then jumps to the loopTo frame number and continues playback. The number of times loopFrom is encountered is monitored. When this count matches loopCount, the still store does not jump back to the loopTo point and plays to the mark out point. A special case of loopCount (-1) allows the loop region to repeat forever.

Note that loopFrom is normally after loopTo. If loopTo is set after loopFrom, and loopCount is not zero, the still store will jump from the loopFrom point to the loopTo point then play to the mark out point. No repeats occur, but the material between loopFrom and loopTo is skipped.

Key Offset

Video + key images are often captured in two passes, one for the video and one for the key. For animations, temporal alignment of the video and key components may be difficult in a two-pass capture. The keyOffset parameter allows the timing of the key signal to be slipped relative to the video so that the two play back in sync even if they are captured out of sync.

Animations that are generated rather than captured will probably not have video to key sync problems and will not need to adjust key offset. But the key offset may need to be considered when importing still store animations that were captured on the still store.

This parameter is not currently implemented, and should be set to 0. This means at this time the video and key portions of a video plus key animation must be temporally aligned.

Index

Symbols

- .gvi file
 - Still Store format [132](#)

A

- Activity check, Digital Effects I/F [111](#)
- Address Byte [44](#)
- All Stop Command [88](#)
 - Effects Addresses [88, 89](#)
- Auto and Key Transition Rate Commands [67](#)
 - Effects Addresses [68](#)
 - Transition Rate [68](#)

B

- Baud Rate
 - changing tally port setting [15](#)
- baud rates
 - serial tally [15](#)
- Break Character [42](#)
- Bus Controller [42](#)
- Byte Count [45](#)
- Byte Format
 - Effects Addresses
 - Timeline Control Commands [84, 85](#)
 - Effects Addresses, Learn and Recall E-MEM Register Commands [76, 77, 86](#)
 - Effects Register Number, Learn and Recall E-MEM Register Commands [75](#)
 - Key Source, Split Key Commands [59](#)
 - Learn Mode, Learn and Recall E-MEM Register Commands [73](#)
 - Load Option, Save and Load Data Commands
 - E-MEM [81](#)
 - RMem [81](#)
 - SourceMem [81](#)
 - Suite Prefs [81](#)
 - System Config [81](#)
 - User Prefs [81](#)
 - Recall Mode, Learn and Recall E-MEM Register Commands [74](#)

- Save Option, Save and Load Data Commands
 - E-MEM [80](#)
 - RMem [80](#)
 - SourceMem [80](#)
 - Suite Prefs [80](#)
 - System Config [80](#)
 - User Prefs [80](#)
- Transition Mode Command [66](#)
- Transition Rate HD byte, Auto and Key Transition Rate Commands [69](#)
- Transition Rate TN byte, Auto and Key Transition Rate Commands [69](#)
- Transition Rate UN byte, Auto and Key Transition Rate Commands [70](#)

C

- CCIR 601
 - Still Store format [131](#)
- colorimetry
 - Still Store [133](#)
- Command Accept bit [46](#)
- Command Code Byte [45](#)
- command/message block [44](#)
- Command/Message Block Structure [44](#)
- Commands [45](#)
 - All Stop [88](#)
 - Auto and Key Transition Rate [67](#)
 - Learn (PBus II) [99](#)
 - Learn and Recall E-MEM Register [71](#)
 - Preview Bus [57](#)
 - Pushbutton Select and Control [59](#)
 - Query (PBus II) [102](#)
 - Read (PBus II) [103](#)
 - Recall (PBusII) [100](#)
 - Save and Load Data [78](#)
 - Save and Load Status [82](#)
 - Software Version [90](#)
 - Source Bus [50](#)
 - Source Select [55](#)
 - Split Key [58](#)
 - Timeline Control [83](#)
 - Transition Mode [65](#)

- Trigger (PBus II) [101](#)
- Wipe Pattern [63](#)
- Write (PBus II) [104](#)
- communication errors [46](#)
- Contribution Tally
 - Communications [26](#)
 - External Processing [19](#)
 - Kalypso [16](#)
 - Kayenne [30](#)
 - Zodiak [16](#)
- Contribution Tally Command Codes
 - Kalypso [25](#)
 - Zodiak [25](#)
- Contribution Tally Message Parsing
 - Kalypso [28](#)
 - Kayenne [39](#)
 - Zodiak [28](#)
- Contribution Tally Message Structure
 - Kalypso [26](#)
 - Kayenne [30](#)
 - Zodiak [26](#)
- Contribution Tally Output Status
 - Kayenne [36](#)
- crop
 - Still Store format [144](#)

D

- Data Field Format, Timeline Control Commands [86](#)
- data file
 - Still Store format [132](#)
- Digital Control Interface [42](#)
- Digital Effects Interface
 - Activity Check [111](#)
 - Examples [124–127](#)
 - General Background [105](#)
 - General Protocol Notes [106](#)
 - Message Format [113, 116](#)
 - Issue [117](#)
 - Parameter Article [118](#)
 - Return Code [120](#)
 - Send Event [119](#)
 - Set [115](#)
 - Subscribe [115](#)
 - Unsubscribe [116](#)
 - Message Timing [107](#)
 - Other Command Specifics [122](#)

- Parameters [121](#)
- Running Effects [112](#)
- Source Selection [109](#)
 - Near/Far Front/Back [109](#)
 - Source Hold [110](#)
- Subscription [112](#)
- Tally [106, 110](#)
- Transmission Media and Protocols [123](#)

documentation online [4](#)

E

- eDPM Contribution Tally
 - Kayenne [35](#)
- Effects Address Byte [45](#)
- Effects Addresses
 - All Stop Command [88, 89](#)
 - Auto and Key Transition Rate Commands [68](#)
 - Learn and Recall E-MEM Register Commands [72, 73](#)
 - Learn and Recall E-MEM Register Commands
 - Byte Format [76, 77, 86](#)
 - Pushbutton Select and Control Commands [60](#)
 - Save and Load Data Commands [78](#)
 - Save and Load Status Commands [82](#)
 - Source Bus Commands [51, 52](#)
 - Source Select Command [56, 57](#)
 - Split Key Commands [58](#)
 - Timeline Control Commands [83](#)
 - Transition Mode Command [65](#)
 - Wipe Pattern Command [63](#)
- E-MEM
 - and PBus II [94](#)
- Error detection [46](#)
- Examples, Digital Effects Interface [124–127](#)
- External Processing,
 - Contribution Tally [19](#)

F

- FAQ database [4](#)
- fenced image
 - Still Store format [132](#)
- frame accurate control
 - PBus II [96](#)
- frequently asked questions [4](#)
- FTP

Still Store format [132](#)

G

Grass Valley web site [4](#)

gva file

Still Store format [132](#)

H

handshaking [46](#)

HD file size

Still Store format [139](#)

HD image data

Still Store format [139](#)

I

image handle

Still Store format [132](#)

Image Store Contribution Tally

Kayenne [36](#)

Iso Tally [12](#)

K

Kaleidoscope

PBus II [95](#)

Kayenne

Contribution Tally [30](#)

key offset

Still Store format [145](#)

Keyer Modifications [12](#)

L

Learn

PBusII [99](#)

Learn and Recall E-MEM Register Commands

[71](#)

Effects Addresses [72, 73](#)

Registers [75](#)

line rate

HD Still Store [142](#)

Look-ahead Tally [11](#)

loops

Still Store format [145](#)

M

M/E Contribution Tally

Kalypso [18](#)

Kayenne [32](#)

Zodiak [18](#)

mark in/out

Still Store format [145](#)

Message Format

Digital Effects I/F [113](#)

Issue [117](#)

Parameter Article [118](#)

Return Code [120](#)

Send Event [119](#)

Set [115](#)

Subscribe [115](#)

Unsubscribe [116](#)

metadata

Still Store format [141](#)

metadata file

Still Store format [132](#)

Mode Byte, Transition Mode Command [66](#)

Mode Byte-DSK, Transition Mode Command [66](#)

Model Numbers, Software Version Commands
[90](#)

N

Names [14](#)

NTSC Transition Rate [68](#)

O

On-air Tally [11](#)

online documentation [4](#)

Other Command Specifics, Digital Effects I/F
[122](#)

Output Contribution Tally

Kalypso [21](#)

Zodiak [22](#)

output control

Still Store format [143](#)

output freeze

Still Store format [144](#)

Output Tally [11](#)

P

- packed data
 - Still Store format [131](#)
- paired files
 - Still Store format [132](#)
- PAL Transition Rate [68](#)
- Parameters, Digital Effects I/F [121](#)
- PBus II
 - command structure [97](#)
 - controlled device identification [98](#)
 - effect numbers [95](#)
 - E-MEM [94](#)
 - frame accurate control [96](#)
 - hardware interface [96](#)
 - Kaleidoscope [95](#)
 - Learn [99](#)
 - messages [97](#)
 - protocol [93](#)
 - Query [102](#)
 - Query Response [102](#)
 - Read [103](#)
 - Recall [100](#)
 - system description [93](#)
 - Trigger [95](#), [101](#)
 - Write [104](#)
 - Write Response [104](#)
- Peripheral Bus II
 - see also* PBus II
- Peripheral protocol [93](#)
- pixel group
 - HD Still Store [139](#)
- position
 - Still Store format [144](#)
- Preview Bus Command [57](#)
 - Source Numbers [57](#)
- Processing Loops [13](#)
- progressive scan bit
 - HD Still Store [142](#)
- Protocol Error bit [46](#)
- Pushbutton Select and Control Commands [59](#)
 - Effects Addresses [60](#)
 - Pushbutton Numbers [61](#), [62](#)

Q

- Query
 - PBusII [102](#)

- Query response
 - PBusII [102](#)

R

- Read
 - PBusII [103](#)
- Ready State [49](#)
- Recall
 - PBusII [100](#)
- Requests [45](#)
- Running Effects, Digital Effects I/F [112](#)

S

- Save and Load Data Commands [78](#)
 - Effects Addresses [78](#)
 - Load Option Byte Formats [81](#)
 - Name [78](#)
 - Reg Number [78](#)
 - Save Option Byte Formats [80](#)
- Save and Load Status Commands [82](#)
 - Effects Addresses [82](#)
 - Status Message [82](#)
- SD file size
 - Still Store format [136](#)
- SD image data
 - Still Store format [136](#)
- Selection Tally [12](#)
- serial data word [41](#)
- Serial Interface Protocol State [43](#)
- Serial Tally [15](#)
- shaped video
 - Still Store format [134](#)
- SMTPT 274M & 296M
 - Still Store format [131](#)
- software download from web [4](#)
- Software Version Commands [90](#)
 - Model Numbers [90](#)
 - Version Number [90](#)
- Source
 - IDs [14](#)
 - names [14](#)
- Source Bus Commands [50](#)
 - Effects Addresses [51](#), [52](#)
 - Source Numbers [53](#), [54](#)

- Source ID Contribution Tally
 - Kalypso 17
 - Zodiak 17
 - Source Name Contribution Tally
 - Kalypso 23
 - Kayenne 37
 - Zodiak 23
 - Source Numbers
 - Preview Bus Command 57
 - Source Bus Commands 53, 54
 - Split Key Commands 59
 - Source Select Command 55
 - Effects Addresses 56, 57
 - Source Selection
 - Hold 110
 - Near/Far Front/Back 109
 - Split Key Commands 58
 - Effects Addresses 58
 - Source Numbers 59
 - status message 46
 - Status Message, Save and Load Status
 - Commands 82
 - Status Replies 46
 - Still Store
 - colorimetry 133
 - Still Store Contribution Tally
 - Kalypso 20
 - Zodiak 20
 - Still Store format
 - .gva file 132
 - .gvi file 132
 - CCIR 601 131
 - changes 133
 - crop 144
 - data file 132
 - fenced image 132
 - FTP 132
 - HD file size 139
 - HD image data 139
 - HD pixel group 139
 - image handle 132
 - key offset 145
 - line rate for HD 142
 - loops 145
 - mark in/out 145
 - metadata 141
 - metadata file 132
 - output control 143
 - output freeze 144
 - packed data 131
 - paired files 132
 - position 144
 - progressive scan bit for HD 142
 - SD file size 136
 - SD image data 136
 - shaped video 134
 - SMTPT 274M & 296M 131
 - software release compatibility 133
 - video conversion 133
 - Still Store hardware types 131
 - Subscribe
 - Parameter Service 116
 - Subscription, Digital Effects I/F 112
 - Switcher Model Features Summary 91
 - switcher models supporting 93
- ## T
- Tally
 - Background Information 10
 - Calculation Basics 12
 - Iso 12
 - Kalypso System Overview 9
 - Kayenne System Overview 10
 - Look-ahead 11
 - On-air 11
 - Output 11
 - Relay Outputs 14
 - Selection 12
 - Serial 15
 - System 13
 - Types 11
 - Zodiak System Overview 9
 - Tally Port
 - changing serial settings 15
 - Tally Source IDs
 - Kayenne 31
 - Tally, Digital Effects I/F 110
 - Timeline Control Commands 83
 - Data Field Format 86
 - Effects Addresses 83
 - Transition Mixer Modifications 13
 - Transition Mode Command 65
 - Effects Addresses 65
 - Mode Byte 66
 - Mode Byte-DSK 66

Index

Transition Rate [68](#)
Transmission Media and Protocols, Digital
 Effects I/F [123](#)
Trigger
 PBus II [95](#)
 PBusII [101](#)

U

Update Contribution Tally
 Kalypso [25](#)
 Kayenne [38](#)
 Zodiak [25](#)

V

Version Number
 Software Version Commands [90](#)
video conversion
 Still Store format [133](#)

W

web site documentation [4](#)
web site FAQ database [4](#)
web site Grass Valley [4](#)
web site software download [4](#)
Wipe Pattern Command [63](#)
 Effects Addresses [63](#)
 Wipe Numbers [64](#)
Write
 PBusII [104](#)
Write Response
 PBusII [104](#)