



**Oxtel Series
Automation
Protocol**

01035-16-001.Provisional
© October 2011

Oxtel v16-001 PROVISIONAL

Copyright © 1999-2011 Miranda Technologies Inc.. All rights are reserved and, under the copyright laws, this manual may not be reproduced in any form, in whole or in part, without the prior written consent of Miranda Technologies Inc..

Oxtel v16-001 PROVISIONAL

Associated Publications

Description

Presmaster Automation Protocol
Imagestore 750 User Manual
LGK-3901/DSK-3901 User Guide

Part number

01235
M872-9900-300
M887-9900-200

Contacts

For technical assistance, please contact your nearest Miranda Technical Support centre:

Americas (9:00am - 9:00pm EST)
Telephone +1-800-224-7882
Fax: +1-514-335-1614
techsupp@miranda.com

Europe, Middle East, Africa, UK (9:00am - 6:00pm GMT)
Telephone +44 (0) 118 952 3444
Fax: +44 (0) 118 952 3401
eurotech@miranda.com

France (9:00am - 5:00pm GMT+1)
Telephone +33 1 55 86 87 88
Fax: +33 1 55 86 00 29
eurotech@miranda.com

Asia (9:00am - 5:00pm GMT+8)
Telephone +852-2539-6987
Fax: +852-2539-0804
asiatech@miranda.com

China
Telephone: +86-10-5873-1814
asiatech@miranda.com

www.miranda.com

Table of Contents

Associated Publications	3
Contacts	3
Table of Contents	4
Introduction	23
Applicability	23
Overview	24
Deprecation Notice	24
Interconnections	25
Cable	25
Pin Assignments	26
Data Link Layer	28
Serial Data Layout	28
Data Transfer Rate	28
Coded Binary	28
Transport Layer	29
Serial Commands	29
Response Commands	31
Unsolicited Tally Commands	32
TCP/IP Automation	33
Network-Format (Preferred)	33
Serial-Format	33
Command Packets	34
Response Commands	34
Unsolicited Tallies	34

Command Formatting	35
Space-Separated	35
Fixed Length	35
Response Commands and Tallies	36
String Parameters / Escaped Characters	36
Floating-Point Parameters	37
Keying Layers	38
Layer Numbers	38
Layer Modes	39
Automation Commands	41
Keyers	42
Commands and Responses	42
0 Fade Keyer to Black 0	42
1 Fade Keyer 1	42
2 Cut Keyer to Black 2	43
3 Cut Keyer 3	44
4 Set Layer Mask 4	44
5 Layer Mask Enable 5	45
6 Swap PVW and PGM Image using Cut 6	45
7 Swap PVW and PGM Image using V-fade 7	46
B Set Transition Duration B	46
C Set Self Key, Separate Key or No Key C	47
D Set Key Linear or Full D	48
E Set Key Normal or Invert E	49
F Set Clip, Gain and Transparency F	49
F Enquire Clip, Gain and Transparency F	50
Unsolicited Tallies	51
3 Keyer Position Tally 3	51
Yc Layer Rate Tally Yc	52
Yf Fade-To-Black Position Tally Yf	52
Stores	54
Commands and Responses	56

R0	Load Image	R0	56
R0	Enquire Load Image	R0	57
R3	Enquire File Info	R3	58
R4	Query First File	R4	59
R5	Query Subsequent File	R5	59
R6	Enquire Extended File Information	R6	60
R7	Preload Image	R7	62
R9	Emergency to Air	R9	62
Ra	Enquire Image Count	Ra	63
Rm	Image Load Mode	Rm	63
Rm	Enquire Image Load Mode	Rm	64
RA	Validate Template	RA	64
RC	Enquire Number of Media Files	RC	66
A	Erase Store	A	66
G	Set Image Position	G	66
G	Enquire Image Position	G	67
H	Load Live Video	H	68
H	Enquire Live Video	H	69
O	Enquire Image Status	O	69
Xi	Set Store-Keyer Configuration	Xi	70
Xi	Enquire Store-Keyer Configuration	Xi	71
Unsolicited Tallies			72
Y9	Image Load Tally	Y9	72
YA	Image Preload Tally	YA	72
YB	Media Tally	YB	72
YB	Enable Media Tallies	YB	73
YB	Enquire Media Tallies	YB	75
YG	Load Status Tally	YG	75
O	Image Status Tally	O	75
YO	Enable Image Status Tallies	YO	76
YO	Enquire Image Status Tallies	YO	77
Xi	Store-Keyer Configuration Tally	Xi	77
YS	Enable Store-Keyer Configuration Tallies	YS	78
YS	Enquire Store-Keyer Configuration Tallies	YS	78

Animations

79

Commands and Responses

79

S0	Start Animation	S0	79
S1	Stop Animation	S1	79
S2	Select Animation Frame	S2	80
S4	Restart Animation	S4	80
Timers			82
Commands			82
T0	Stop Timer	T0	82
T1	Start Timer	T1	82
T2	Reset Timer	T2	83
T3	Set Timer to Value	T3	83
Easytext			84
Introduction			84
Summary			84
Easytext Straps			86
Commands and Responses (Settings)			87
Z0	Update Text Field	Z0	87
Z1	Change Box Size and Position	Z1	87
Z2	Set Text Font and Colour	Z2	88
Z3	Render Box	Z3	89
Z4	Change Image	Z4	89
Z9	Set Text Drop Shadow	Z9	90
ZA	Set Transparency	ZA	90
ZB	Set Text Tracking	ZB	91
ZE	Set Text Alignment	ZE	92
ZF	Set Text Wrapping	ZF	92
Za	Enquire Textbox	Za	93
Zb	Enquire Template	Zb	94
Zd	Text Box Update	Zd	94
Ze	Image Update	Ze	95
Commands (Backgrounds)			96
Z6	Set Text Background to Matte	Z6	96
Z7	Set Text Background to Gradient	Z7	96
Z8	Set Text Background to Clear	Z8	98
ZD	Set Template Background	ZD	98

Commands and Responses (Straps)	100
Z5 Run Strap Z5	100
Zg Pause/Restart Strap Zg	100
ZC Set Strap Speed ZC	100
Zc Enquire Crawl Loop Count Zc	101
ZH Set Datasource Timer ZH	101
Commands and Responses (Intuition[+])	102
Zc Enquire Crawl Count Loop Zc	102
Zf Stop Animation Zf	102
Zg Pause/Restart Strap Zg	103
Zh Set Crawl Loop Count Zh	103
Unsolicited Tallies	104
Yd Strap Loop Count Yd	104
Yg Text Crawl Count Tally Yg	104
Yj Animation/Clip Cycle Tally Yj	105
Datasources	106
Custom Datasources	106
Commands and Responses	108
m0 Set Global Datasource Value m0	108
m1 Delete Global Datasource m1	109
m2 Set Layer Datasource Value m2	109
m3 Delete Layer Datasource m3	109
m4 Enquire Global Datasource List m4	109
m5 Enquire Global Datasource Value m5	110
m6 Enquire Layer Datasource List m6	110
m7 Enquire Layer Datasource Value m7	111
m80 Delete All Datasources m80	111
Internal Datasources	112
Emergency Alert System (EAS)	117
Commands	117
n1 EAS On n1	117
n2 EAS Off n2	118
n3 EAS Set Text n3	118
n4 EAS Set Priority n4	118

SDI Video	119
Inputs	119
Commands and Responses	120
1 Set Input Masks 1	120
1 Enquire Input Masks 1	120
XM Enquire Input Mode XM	121
Xq Set Post-A/B Mix Video Delay Xq	122
Xq Enquire Post-A/B Mix Video Delay Xq	122
Xs Enquire Video Standard Xs	123
Xt Set Output Frame Delay Xt	123
Xt Enquire Output Frame Delay Xt	124
A/B Mixer	125
Commands and Responses	125
U0 Cut to A U0	125
U1 Cut to B U1	125
U2 Fade to A U2	126
U3 Fade to B U3	126
U4 Cut AB U4	126
U5 Fade AB U5	127
U6 Set Transition Type U6	127
U6 Enquire Transition Type U6	128
U8 Asymmetric V-Fade AB U8	129
UA Asymmetric Transition UA	129
UB Transition Complete UB	129
UC Fade to Specified Position UC	130
UD Set V-Fade Colour UD	130
UD Enquire V-Fade Colour UD	131
UE Select Mixer Input UE	132
UE Enquire Mixer Input UE	132
UF Select Channel Input UF	133
UF Enquire Channel Input UF	133
UH Set A/B Mix Profile UH	134
UH Enquire A/B Mix Profile UH	135
UI Set Mixer Transition Duration UI	136
UI Enquire Mixer Transition Duration UI	137

UJ	Set Channel Transition Mode	UJ	137
UJ	Enquire Channel Transition Mode	UJ	138
Ua	Enquire Mix Mode	Ua	138
Ub	Set A/B Cut Mode	Ub	139
Ub	Enquire A/B Cut Mode	Ub	140
Unsolicited Tallies			141
Y6	Video Tally	Y6	141
Y6	Enable Video Tallies	Y6	143
Y6	Enquire Video Tallies	Y6	143
U6	Transition Type Tally	U6	144
YU6	Enable Transition Type Tallies	YU6	145
YU6	Enquire Transition Type Tallies	YU6	145
UE	Mixer Input Tally	UE	145
UF	Channel Input Tally	UF	146
UI	Mixer Transition Duration Tally	UI	146
YUI	Enable Mixer Transition Duration Tallies	YUI	147
YUI	Enquire Mixer Transition Duration Tallies	YUI	147
UJ	Channel Transition Mode Tally	UJ	147
YUJ	Enable Channel Transition Mode Tallies	YUJ	148
YUJ	Enquire Channel Transition Mode Tallies	YUJ	148

DVE

149

Commands and Responses			149
W1	Set PGM DVE Mode	W1	149
W2	Load and Run DVE Sequence	W2	151
W7	Run DVE Sequence	W7	152
W8	Run DVE Sequence in Reverse	W8	152
WP	Set PVW DVE Mode	WP	153
WR	Set DVE Routing Input	WR	154
WR	Enquire DVE Routing Input	WR	155
WV	Set DVE Load Action	WV	155
WW	Enquire DVE Load Action	WW	156
WX	Set DVE Configuration	WX	156
WX	Enquire DVE Configuration	WX	157
WY	Load DVE Sequence with Action	WY	157
WZ	Enquire DVE Sequence Name	WZ	158
Wb	Play DVE Sequence	Wb	158

Wd	Set DVE ID	Wd	159
Wd	Enquire DVE ID Count	Wd	159
Wd	Enquire DVE ID	Wd	159
Unsolicited Tallies			161
YWb	Enable DVE Tallies	YWb	161
YWb	Enquire DVE Tallies	YWb	161
Wb	DVE Sequence Tallies	Wb	161
W1	PGM DVE Mode Tallies	W1	162
WP	PVW DVE Mode Tallies	WP	162

Video Preview **163**

Commands and Responses			163
Y1	Set Preview Keyer	Y1	163
Y3	Set Preview A/B Mix	Y3	163
Yp	Set Active Preview Keyers	Yp	164
Yp	Enquire Active Preview Keyers	Yp	164
XH	Set Preview Output	XH	165
XH	Enquire Preview Output	XH	166
XP	Set Clean Feed Output	XP	166
XP	Enquire Clean Feed Output	XP	168
XT	Set Monitor Feed Output	XT	169
XT	Enquire Monitor Feed Output	XT	170

Audio **172**

Other Advanced Audio Commands			173
Channel/Pair/Group			173
Gain Adjustments			173
Commands (Inputs)			175
j0_0	Set Input Shuffle	j0_0	175
jk	Set Input Shuffle Bitwise	jk	176
j0_1	Set Input Mute	j0_1	177
j0_2	Set Input Trim	j0_2	178
j0_3	Set Input Phase	j0_3	178
j0_4	Set Input Level	j0_4	179
j0_5	Set Input Source	j0_5	180
Commands (Outputs)			182

j1_0	Set Output Level	j1_0	182
j1_3	Set Output Level via Gain Type	j1_3	183
j1_1	Set Output Mute	j1_1	184
j1_2	Output Embed	j1_2	184
Commands (Voice-Overs)			186
j2_0	Set Voice-over Duck	j2_0	187
j2_1	Set Voice-over Preset	j2_1	188
j2_2	Set Voice-over Position	j2_2	188
Commands (Transition Rates)			190
j30	Set Silence Fade Rate	j30	190
j31	Set A/B Mixer Fade Rate	j31	190
j32	Set Voice-over 1 Fade Rate	j32	191
j33	Set Voice-over 2 Fade Rate	j33	191
j34	Set Voice-over 3 Fade Rate	j34	192
j35	Set Voice-over 4 Fade Rate	j35	192
Commands (Transitions)			194
j40	Cut A/B	j40	194
j41	Fade A/B	j41	194
j42	Cut To Silence	j42	195
j43	Fade To Silence	j43	195
j441	Fade Voice-over 1	j441	195
j442	Fade Voice-over 2	j442	196
j451	Fade Voice-over 3	j451	196
j452	Fade Voice-over 4	j452	197
j461	Cut Voice-over 1	j461	198
j462	Cut Voice-over 2	j462	198
j471	Cut Voice-over 3	j471	199
j472	Cut Voice-over 4	j472	199
Commands (Audio Follow)			201
j50	Set Silence Follow Fade-to-Black	j50	201
j51	Set Audio A/B Follow Video A/B	j51	201
j52	Set Easyplay Follow Keyers	j52	202
j53	Voice-over Follow Easyplay	j53	202
Commands (Output Preview)			204
j6	Set Audio Preview	j6	204
jh	Audio Clean Feed Select	jh	205
ji	Audio Monitor Feed Select	ji	206

Commands and Responses (Enquires)	207
j70 Enquire Audio Input j70	207
j71 Enquire Voice-over j71	208
j72 Enquire Outputs j72	209
j73 Enquire Rates j73	210
j74 Enquire Audio Follow Video j74	210
j75 Enquire Audio Preview j75	211
j76 Enquire Audio Clean Feed j76	212
j77 Enquire Audio Monitor j77	213
j78 Enquire Audio AES Status j78	214
j79 Enquire Output Gain j79	214
Commands (A/B Mixer)	216
ja Audio A/B Position ja	216
jb Audio A/B Mix Mode jb	216
jc Audio A/B Asymmetric Transition jc	216
jd Audio A/B Fade to Position jd	217
je Set Audio A-Fade Position je	217
jf Set Audio B-Fade Position jf	218
jj Set A/B Transition Switch Point jj	218
Commands (Delays)	219
jl Set Audio Delay jl	219
Commands (Miscellaneous)	220
j8 Set Audio FX Send j8	220
j9 Set Audio Data Path j9	221
Unsolicited Tallies	222
Y0 Audio Metering Tally Y0	222
Y0 Enable Audio Metering Tallies Y0	222
Y8 Audio Tally Y8	223
Y8 Enable Audio Tallies Y8	223
Y8 Enquire Audio Tallies Y8	224
j7 Audio Tallies j7	224
Advanced Audio	226
Bus Types	227
Deprecated Audio Commands	228
Commands and Responses (Multiplexer)	230

jn0	Set Multiplexer Source	jn0	230
jn0	Enquire Multiplexer Source	jn0	230
jn1	Enquire Multiplexer Source Count	jn1	231
jn2	Enquire Multiplexer Source Name	jn2	232
Commands and Responses (Shuffles)			233
jo0	Set Shuffle Crosspoint	jo0	233
jo1	Load Shuffle Preset	jo1	234
jo1	Enquire Shuffle Preset	jo1	234
jo2	Save Shuffle Preset	jo2	235
jo3	Enquire Shuffle Preset List	jo3	235
jo4	Enquire Shuffle Block Size	jo4	236
Commands and Responses (Gains)			237
jp0	Set Mute	jp0	237
jp1	Set Trim	jp1	237
jp2	Set Phase Invert	jp2	238
jp3	Set Level	jp3	238
jp4	Set Master Gain	jp4	239
jp5	Load Gain Preset	jp5	239
jp5	Enquire Gain Preset	jp5	240
jp6	Save Gain Preset	jp6	240
jp7	Enquire Gain Preset List	jp7	241
jp8	Enquire Gain Block Size	jp8	241
Commands and Responses (Voiceovers)			243
jq0	Set Voice-over Duck Level	jq0	244
jq0	Enquire Voice-over Duck Level	jq0	244
jq1	Set Voice-over Preset Level	jq1	244
jq1	Enquire Voice-over Preset Level	jq1	245
jq2	Set Voice-over Fade Rate	jq2	245
jq2	Enquire Voice-over Fade Rate	jq2	245
jq3	Set Voice-over to Position	jq3	246
jq4	Fade Voice-over	jq4	246
jq5	Cut Voice-over	jq5	246
jq5	Enquire Voice-over Position	jq5	247
jq6	Set Voice-over Fade Rate (Panel)	jq6	247
jq6	Enquire Voice-over Fade Rate (Panel)	jq6	248
jq7	Enquire PST Voice-over Position	jq7	249
jt0	Enquire Audio Meter Width	jt0	249

Unsolicited Tallies	250
YC Enable Advanced Audio Tallies YC	250
YC Enquire Advanced Audio Tallies YC	251
YC_jo0 Enable Shuffle Crosspoint Tallies YC_jo0	251
YC_jp0 Enable Audio Mute Tallies YC_jp0	253
YC_jp1 Enable Audio Trim Tallies YC_jp1	254
YC_jp2 Enable Audio Invert Tallies YC_jp2	255
YC_jp3 Enable Audio Level Tallies YC_jp3	256
YC_jq6 Enable Voice-over Fade Rate Tallies YC_jq6	257
js Audio Meter Tally js	257
YC_js Enable Audio Meter Tallies YC_js	258
YC2js Enquire Audio Meter Tallies YC2js	259

Easyplay 260

Commands and Responses	260
k0 Load Audio Clip (Stream 1) k0	260
k1 Start Audio Playout (Stream 1) k1	261
k2 Stop Audio Playout (Stream 1) k2	261
k3 Unload Audio Clip (Stream 1) k3	262
k4 Enquire Audio Clip (Stream 1) k4	262
R8 Enquire Audio File Information R8	263
RB Get Audio File Duration RB	264

Easyplay 2 265

Streams and Queues	265
Commands and Responses	266
k6 Enquire Easyplay Stream Count k6	266
k7 Enquire Easyplay Queue Count k7	266
k8 Set Easyplay Associated Voiceover k8	267
k8 Enquire Easyplay Associated Voiceover k8	267
k9 Set Easyplay Play Follow Mode k9	267
k9 Enquire Easyplay Play Follow Mode k9	268
ka Load Audio Clip ka	269
ka Enquire Audio Clip ka	269
kb Play Audio Clip kb	270
kc Stop Audio Clip kc	270

kd Unload Audio Clip kd	271
ke Enquire Easyplay Stream Valid ke	271
kf Set Easyplay Load Follow Mode kf	271
kf Enquire Easyplay Load Follow Mode kf	272
Unsolicited Tallies	273
k8 Easyplay Associated Voiceover Tally k8	273
k9 Easyplay Play Follow Mode Tally k9	273
ka Load Audio Clip Tally ka	274
kb Play Audio Clip Tally kb	274
kc Stop Audio Clip Tally kc	275
kf Easyplay Load Follow Mode Tally kf	275
YE Enable Easyplay Tallies YE	276
YE Enquire Easyplay Tallies YE	276
Dolby and Up-Mix	278
Commands	278
q00 Set Up-Mix Mode q00	278
q10 Set Dolby E Encode Mode q10	278
q20 Set Dolby Digital (AC3) Encode Mode q20	279
q30 Set Dolby Decode Mode q30	279
Metadata	280
Commands and Responses (Bulk)	280
oT Load Metadata Preset oT	280
oT Enquire Metadata Preset oT	280
Commands and Responses (Granular)	281
o0 Set Pitch Shift Code o0	281
o0 Enquire Pitch Shift Code o0	281
o1 Set Program Description o1	281
o1 Enquire Program Description o1	282
o2 Set Program Bitstream Mode o2	282
o2 Enquire Program Bitstream Mode o2	282
o3 Set Program Audio Coding Mode o3	283
o3 Enquire Program Audio Coding Mode o3	283
o4 Set Program Centre Down-Mix Level o4	283
o4 Enquire Program Centre Down-Mix Level o4	284

o5	Set Program Surround Down-Mix Level	o5	284
o5	Enquire Program Surround Down-Mix Level	o5	285
o6	Set Program Dolby Surround Mode	o6	285
o6	Enquire Program Dolby Surround Mode	o6	285
o7	Set Program LFE Enable	o7	286
o7	Enquire Program LFE Enable	o7	286
o8	Set Program Dialogue Normalisation	o8	286
o8	Enquire Program Dialogue Normalisation	o8	287
o9	Set Program Production Mix Level	o9	287
o9	Enquire Program Production Mix Level	o9	287
oA	Set Program Production Room Type	oA	287
oA	Enquire Program Production Room Type	oA	288
oB	Set Program Copyright Flag	oB	288
oB	Enquire Program Copyright Flag	oB	289
oC	Set Program Original Bitstream Format	oC	289
oC	Enquire Program Original Bitstream Format	oC	289
oD	Set Program Down-Mix Mode	oD	290
oD	Enquire Program Down-Mix Mode	oD	290
oE	Set Program Down-Mix Lt/Rt Centre Mix Level	oE	290
oE	Enquire Program Down-Mix Lt/Rt Centre Mix Level	oE	291
oF	Set Program Down-Mix Lt/Rt Surround Mix Level	oF	291
oF	Enquire Program Down-Mix Lt/Rt Surround Mix Level	oF	292
oG	Set Program Down-Mix Lo/Ro Centre Mix	oG	292
oG	Enquire Program Down-Mix Lo/Ro Centre Mix	oG	292
oH	Set Program Down-Mix Lo/Ro Surround Mix Level	oH	293
oH	Enquire Program Down-Mix Lo/Ro Surround Mix Level	oH	293
oI	Set Program Dolby Surround EX™ Mode	oI	293
oI	Enquire Program Dolby Surround EX™ Mode	oI	294
oJ	Set Program Dolby Headphone Mode	oJ	294
oJ	Enquire Program Dolby Headphone Mode	oJ	295
oK	Set Program A/D Converter Type	oK	295
oK	Enquire Program A/D Converter Type	oK	295
oL	Set Program High Pass Filter Enable	oL	295
oL	Enquire Program High Pass Filter Enable	oL	296
oM	Set Program Low Pass Filter Enable	oM	296
oM	Enquire Program Low Pass Filter Enable	oM	297
oN	Set Program LFE LPF Enable	oN	297

oN	Enquire Program LFE LPF Enable	oN	297
oO	Set Program Surround 90 Degree Phase-Shift	oO	297
oO	Enquire Program Surround 90 Degree Phase-Shift	oO	298
oP	Set Program Surround 3dB Attenuation	oP	298
oP	Enquire Program Surround 3dB Attenuation	oP	299
oQ	Set Program RF Pre-emphasis	oQ	299
oQ	Enquire Program RF Pre-emphasis	oQ	299
oR	Set Program Compression Profile/Word	oR	300
oR	Enquire Program Compression Profile/Word	oR	300
oS	Set Program Dynamic Range Value	oS	301
oS	Enquire Program Dynamic Range Value	oS	301
oU	Set Program Dynamic Range Type	oU	301
oU	Enquire Program Dynamic Range Type	oU	302
Passive Mode			303
	Unsolicited Tallies		303
Ya	Enable Passive Mode Tallies	Ya	303
Yb	Set Passive Mode Transmission Interval	Yb	304
Ye	Enquire Passive Mode	Ye	304
System and Status			305
	Commands and Responses		305
M	Enquire System Status	M	305
N	Enquire Video Layer Status	N	306
X1	Enquire Licence	X1	308
X2	Insert Log Message	X2	309
X3	Enquire Command Availability	X3	309
X4	Set Device Bypass	X4	310
X4	Enquire Device Bypass	X4	311
X5	Enquire Input Colour Field	X5	311
XA	Enquire Slave Layer Status	XA	313
XC	Enquire Serial Baud Rate	XC	314
XI	Enquire Reference	XI	315
XJ	Enquire Clock Offset	XJ	315
XK	Enquire Serial Number	XK	316
XN	Enquire Serial Protocol	XN	317

XO	Enquire Serial Type	XO	317
XR	Enquire Number of Fill/Key	XR	318
XS	Enquire Preview Count	XS	318
XU	Enquire Extended Serial Number	XU	318
XX	Enquire Input Status	XX	319
Xb	Enquire Full Version Number	Xb	319
Xc	Pass Close Caption	Xc	320
Xc	Enquire Close Caption	Xc	321
Xe	Set Date and Time	Xe	321
Xn	Enquire Product Name	Xn	322
Xu	Enquire AES Audio Signal Type	Xu	322
Xv	Enquire Embedded Audio Signal Type	Xv	323
Xz	Enquire URS Reference	Xz	323
	Unsolicited Tallies		325
Y7	Wake-up Tally	Y7	325
Y7	Request Wake-up Packet	Y7	325
YD	Heartbeat Tally	YD	325
YD	Set Heartbeat Tallies	YD	326
YD	Enquire Heartbeat Tallies	YD	326

Health

327

	Commands and Responses		327
X0	Enquire Temperature	X0	327
X6	Enquire Voltages	X6	327
XV	Enquire System Health	XV	329
Xh	Enquire Extended Health Commands	Xh	330
Xhf	Enquire Extended Fan Health	Xhf	330
Xhp	Enquire Extended PSU Health	Xhp	331
Xht	Enquire Extended Temperature Health	Xht	332
Xhv	Enquire Extended Voltage Health	Xhv	333

Scheduler

335

	Commands and Responses		335
i0	Add Scheduled Command	i0	335
i1	Delete Scheduled Command	i1	335
i2	Delete All Scheduled Commands	i2	336

i3	Enquire Next Scheduled Command	i3	336
General Purpose Interface			337
Commands and Responses			337
X7	Virtual GPI	X7	337
X8	Enquire GPI Status	X8	337
X9	GPI Special	X9	338
XB	GPI Output Status	XB	339
Macros			341
Commands and Responses			341
x0	Trigger Macro	x0	341
x1	Enquire Macro Count	x1	341
x2	Enquire Macro By ID	x2	342
x3	Enquire Macro By Index	x3	342
Master Control			343
Arm and Take			343
Commands			345
p1	Arm	p1	345
p0	Disarm	p0	345
p4	Take	p4	345
Unsolicited Tallies			347
Yip1	Arm Tally	Yip1	347
Yip0	Disarm Tally	Yip0	347
Yip4	Take-Start Tally	Yip4	348
Yip5	Take-End Tally	Yip5	348
Yi	Enable Arm/Take Tallies	Yi	349
Yi	Enquire Arm/Take Tallies	Yi	349
Routing Control			351
Source Type and Source Index			351
Destination Type and Destination Index			352
Commands and Responses			353
rs0	Make Source Selection	rs0	353
rs1	Enquire Source Selection	rs1	353
rs2	Enquire Channel Source Count	rs2	354

rs3 Enquire Source Information	rs3	354
rm0 Enquire Salvo Count	rm0	355
rm1 Enquire Salvo Information	rm1	356
rm2 Execute Salvo	rm2	356
Unsolicited Tallies		357
YF Enable Router Tallies	YF	357
YF Enquire Router Tallies	YF	357
rs0 Source Selection Tally	rs0	357
rs3 Source Information Tally	rs3	358
rm1 Salvo Information Tally	rm1	358
rm2 Salvo Tally	rm2	359
Intuition XG		360
Commands and Responses		360
Cue	V0	360
Take	V1	361
Clear	V2	362
Keyer On	V3	362
Keyer Off	V4	363
Set Property	V5	363
Cue Clip	V9	364
Enable External Command Tallies	Va	366
Unsolicited Tallies		366
Enable Vertigo Tallies	V8	367
Cue Tally	V0	367
Take Tally	V1	368
Clear Tally	V2	369
Keyer On Tally	V3	369
Keyer Off Tally	V4	369
Item Specifications		370
Scene Event Summary		371
Cue/Live Selection and Tallies in Action		371
Command Validity Table		373
CRC Generation		387

Automation Examples	389
Keyer Animation Cut/Fade	389
A/B Mixing	389
DVE Moves	390
Command Index	391

Oxtel v16-001 PROVISIONAL

Introduction

Applicability

This user guide describes the automation control protocol used by current Miranda's Oxtel range of broadcast products. The Oxtel series products that can be controlled using this protocol include:

- Imagestore 750 4.2
- Intuition XG
- DSK-3901 4.3
- LGK-3901 4.3
- ISM-3901 4.3

For older Miranda Oxtel products (IS-2/3/300/HD and Intuition), please refer to previous version of this document, Oxtel document 01035-15.

This document describes the Oxtel automation commands which are supported by these Oxtel products and is accurate for the latest software releases at the time of this publication. The latest software versions are shown in brackets, and we recommend that you attempt to keep devices running the most recent version whenever possible.

Individual differences in Oxtel protocol support between different devices are highlighted in the appropriate section and the [Command Validity Table on page 373](#).

The Presmaster Master Control Switcher uses a different protocol – ask for Presmaster Automation Protocol document (part no. 01235) for details.

Some document links:

[Keying Layers](#) [Keyers](#) [Stores](#) [Animations](#) [Easytext](#) [Datasources](#)
[Emergency Alert System](#) [SDI Video](#) [A/B Mixer](#) [Video Preview](#) [DVE](#)
[Audio](#) [Advanced Audio](#) [Easyplay](#) [Easyplay 2](#) [Dolby and Up-Mix](#) [Metadata](#)
[System and Status](#) [Health](#) [Scheduler](#) [General Purpose Interface](#) [Macros](#)
[Master Control](#)
[Intuition XG commands](#)
[Command Validity Table](#) [CRC Generation](#) [Automation Examples](#)
[Command Index](#)

Overview

The automation system is always the controller, and the Oxtel device is controlled. Oxtel automation commands are sent by the controller, and the Oxtel device actions these commands returning, acknowledgements, response commands or unsolicited tallies as required.

With RS-232 or RS-422 serial communications, the link between the controller and controlled device is point-to-point, with a single physical connection.

All products listed on page 23 also allow automation control over TCP/IP. Note however that whilst modern networks are much faster than serial connections, timeliness cannot be guaranteed without the use of scheduled commands. For this reason serial links are normally recommended.

Deprecation Notice

The semi-colon command separator has now been deprecated and will be removed from this protocol in the future. Please see page 30.

Numerous audio commands are deprecated. Please see page 228.

Interconnections

Oxtel automation control normally uses RS-232 or RS-422 asynchronous serial communications, although control is possible over TCP/IP.

Imagestore 750, LGK-3901 and DSK-3901 devices have RJ-45 connectors which are configurable as either RS-232 or RS-422.

The configuration of serial ports (protocol, baud rate, serial type) is covered in individual user manuals for each product.

Cable

All serial interconnection cables attached to Oxtel devices should be constructed following best practice for the purposes of electromagnetic compatibility (EMC).

Cables should always feature a screening braid fabrication, and always be terminated with metal shrouded connectors. The fully screened cable must have a metal or foil shroud completely surrounding the internal wiring within the terminating connectors. This shroud must be connected directly to the cable's screening braid. This connection must be as close to 360 degrees around the cable as possible. Connections using thin wires or drain wires should not be used, since they provide inadequate screening.

To conform with broadcast standard earthing/grounding arrangements, the cable screen should only be connected to ground at one end via the connector shroud.

Pin Assignments

Note: Imagestore 750, LGK-3901 and DSK-3901 products use RJ-45 connectors instead of 9-pin D-type. Adaptors are available to convert between types

RS-232

The RS-232 connector is a 9-way male D-type, which follows the same pin assignments as the standard IBM PC/AT serial port, as shown in the following table:

Pin	Function
2	RX
3	TX
5	GND

All other pins are not used and must be unconnected.

RS-422

The RS-422 pin-out is a 9-way male D-type with the following pin-out:

Pin	Function
1	TX -
2	TX +
3	RX +
4	RX -
5	GND

All other pins are not used and must be unconnected.

Note: The RS-422 pin-out for Imagestore 2/2U/3 products does not conform to the broadcast industry standard commonly known as 'Sony 9-pin'. Please see individual user manuals for details.

The next table shows a typical point-to-point RS-422 connection:

Controller	Oxtel Device
TX -	RX -
TX +	RX +
RX +	TX +
RX -	TX -

Ground	Ground
--------	--------

Note: Imagestore 750, LGK-3901 and DSK-3901 products use RJ-45 connectors instead of 9-pin D-type. Adaptors are available to convert between types.

[to Introduction](#)

Oxtel v16-001 PROVISIONAL

Data Link Layer

Serial Data Layout

1 start bit. 8 data bits. 1 stop bit. No parity bit.

Data Transfer Rate

The following Baud rates can be selected for RS-232 and RS-422 serial ports:

- 9600
- 19200
- 38400
- 57600
- 115200

Coded Binary

Data

Apart from the CRC, ACK, NAK and STX characters, all Oxtel automation protocol bytes are in the range of printable ASCII ($\geq 0x20$).

Control Codes

The control codes used are defined in the following table:

Code	Value
STX0	0x02
STX1	0x03
ACK0	0x04
ACK1	0x05
STX2	0x06
NAK	0x07

Note: The names for these codes do not correspond to ASCII code naming conventions.

Transport Layer

Serial Commands

The following section describes the command packets that are sent by automation to the Oxtel device over RS-232 or RS-422.

Command List

For transmission over point-to-point links, the command list is encapsulated in data packets as follows:

Field	Contents
Byte 0	STX0 or STX1 (alternating)
Byte 1 to n	Command list, where n is the length of the command list
Byte n+1	Command terminator ':' (colon)
Byte n+2	CRC low byte
Byte n+3	CRC high byte

STX0 / STX1

Oxtel devices receiving command packets on a serial connection expect the STX values in successive packets to alternate between STX0 and STX1.

On receipt of a "good packet" (with the correct STX and CRC), the Oxtel device acknowledges the command with an ACK or 'response command'. This occurs in one of two ways depending on whether the automation command was a "set" or "enquire" type of command:

Type	Acknowledgement or Response
Set	Oxtel device transmits a single-character response of either ACK0 or ACK1, where the ACK matches the STX that was originally sent (ACK0 for STX0, ACK1 for STX1). This may be followed by an unsolicited tally command (STX2) if the state of the device changes as a result of the set command. Note that ACKs will be returned even if the command list contains invalid commands or parameters, or errors occur during the processing of the commands.
Enquire	Oxtel device transmits a multi-character 'response command' packet that begins in STX0 or STX1 (where the STX matches the STX that was originally sent), followed by command list, command terminator and CRC bytes (see page 31). Note that there is no ACK in this case.

Automation should wait for an ACK or 'response command' before sending its next packet. However if no ACK or 'response command' is received, then automation must resend the same message with the same STX value. This procedure means that lost ACKs or responses will not cause message duplication.

If further message transmission fails (i.e. all retries either receive a NAK response or they time-out), then the next attempt by automation should use the alternate STX value. This is only important in older products and software when a system reset could cause the Oxtel device to be waiting for the wrong STX. However in most products and software either STX is accepted on any serial connection for:

- The first command packet received after device restart
- The first command packet received following a packet that had an incorrect STX or CRC

ACK / NAK / Response

The rules for transport layer responses depend on the STX and CRC:

STX	CRC	Description
Correct	Correct	ACK or 'response command' is returned by the device to automation (depending on whether the command was a set or enquire).
Incorrect	Correct	Nothing is returned by the device to automation, which must then deal with a time-out.
Correct	Incorrect	NAK is returned by the device to automation.

Command Terminator

The command terminator ':' (colon) marks the end of a command list. It is followed by the two CRC bytes.

Since this character has special meaning it must not occur anywhere else within the command packet. However, when an automation command includes a string parameter that requires a colon character (for example, the m003 command), it can be escaped as "\3A".

Command Separator (Deprecated)

Multiple automation commands can be concatenated into a single command list using the command separator character ';' (semi-colon).

Since this character has special meaning it must not occur anywhere else within the command list. However, when an automation command includes a string parameter that requires a semi-colon character (for example, the m003 command), it can be escaped as "\3B".

Note: We strongly recommend against using command separators to concatenate multiple automation commands. Having just one automation command per command list makes it clear which automation command an ACK or 'response command' is for.

The use of the semi-colon as a command separator is now deprecated. Later revisions of this protocol will remove support for it, though it will remain a reserved character and must continue to be escaped.

Cyclic Redundancy Check (CRC)

Cyclic redundancy check (CRC) bytes provide a means of verifying the transmission and receipt of command data packets.

The 16-bit CRC value must be generated by automation using the contents of the corresponding command list. The resultant value is appended to the command packet (following the command terminator) and transmitted to the Oxtel device as two 8-bit bytes. Using the same CRC algorithm at the receiving end, the Oxtel device verifies the integrity of the data transmitted.

The CRC is constructed from all characters that follow the STX. A software coding example in Appendix C (page 387) shows how to generate and use CRCs.

The same CRC technique is used on the reverse channel by the Oxtel device when it returns transmission packets (responses or unsolicited tallies, but not ACKs or NAKs) back to automation.

Response Commands

Enquiry command requests from automation cause the following 'response command' message to be returned by the Oxtel device:

Field	Contents
Byte 0	STX0 or STX1 (matching the STX of the enquire command)
Byte 1 to n	Response command, where n is the length of the command list
Byte n+1	Status terminator ':' (colon)
Byte n+2	CRC low byte
Byte n+3	CRC high byte

Response messages must not be acknowledged by automation, and cannot be re-transmitted by the Oxtel device. The STX value sent in the 'response command' matches the STX value of the original enquire command packet. If automation fails to receive a status message it must request the status again.

If automation included multiple enquires within a single command packet using command separators (not recommended) there will be multiple 'response command' packets with identical STX bytes.

Unsolicited Tally Commands

Automation can enable changes in various device states to be communicated back to automation as they happen without explicit request. This mechanism is referred to as unsolicited tallies.

Unlike 'response command' packets, which are returned in response to enquire commands from automation, unsolicited messages can be sent by the Oxtel device at any point based on changes in device state.

Since unsolicited tallies are enabled by automation, that automation connection receives a burst of current state information. Then as device states change subsequently (perhaps due to the actions of a panel user), further tallies are received by all registered connections. This way, each automation connection can ensure that it is up to date and in sync with the device states that it needs to know about.

Unsolicited messages are distinguished by the leading STX2 byte.

Field	Contents
Byte 0	STX2
Byte 1 to n	Tally response, where n is the length of the command list
Byte n+1	Status terminator ':' (colon)
Byte n+2	CRC low byte
Byte n+3	CRC high byte

Various types of unsolicited messages are generated by Oxtel devices, including audio metering data, video and audio tallies, and other status information. The transmission of tallies must be explicitly enabled by the automation system for its connection.

Note: Tallies are supported by all Oxtel products, although Imagestore 2/2U/3 must be running software version 2.08 or later.

[to Introduction](#)

TCP/IP Automation

Network-Format (Preferred)

All products listed on page 23 also accept network automation commands on port 5007.

Network-format automation is a simplified version of serial-format automation which takes advantage of TCP's end-to-end reliability. It differs as follows:

- It does not use an STX byte.
- It does not use CRC bytes.
- No ACKs or NAKs are returned.

Otherwise the command list and response command format is exactly the same. In particular the command terminator ':' (colon) is still required to mark the boundaries in the byte stream.

In products that support it, this is the preferred manner of control using the Oxtel protocol.

Serial-Format

Oxtel products have the ability to run the Oxtel automation protocol over a standard TCP/IP socket interface, provided the unit is equipped with Ethernet interface. The packet layout and protocols are exactly as described above for RS-232/RS-422 serial command packet flow, including STX and CRC checksum bytes.

Note: This can be thought of as “serial automation over network”.

Serial-format connections are accepted on port 5006.

Multiple connections can be established to the Oxtel device, allowing distributed or backup control without the need for additional cabling.

When considering using TCP/IP automation, please note the following:

- Any Ethernet-based protocol cannot guarantee frame-accurate reception and there may also be non-deterministic latencies due to network traffic.
- Each TCP/IP connection will have its own parameters for information such as tally responses and audio metering. By default these are turned off on new connections, and need to be enabled (if required) whenever a new connection is established.
- The Oxtel device may limit the number of simultaneous sessions.

Command Packets

For transmission over networks, the command packet is described below:

Field	Contents
Byte 1 to n	Command list, where n is the length of the command list
Byte n+1	Command terminator ':' (colon)

Response Commands

Enquiry command requests from automation cause the following 'response command' message to be returned by the Oxtel device:

Field	Contents
Byte 1 to n	Response command, where n is the length of the command list
Byte n+1	Status terminator ':' (colon)

Unsolicited Tallies

Unsolicited messages can be sent by the Oxtel device at any point based on changes in device state. They are no longer distinguished by the leading STX2 byte.

Field	Contents
Byte 1 to n	Tally response, where n is the length of the command list
Byte n+1	Status terminator ':' (colon)

Command Formatting

Note: Command characters are always case-sensitive.

Space-Separated

Space-separated commands start with a single ASCII command character followed by parameter(s) which are space-separated.

CMD	Param_1	Param_2	Param_n

Note: Blank columns are left in command definitions where a space character (0x20) is required.

The terms used in the above command list are listed below:

Term	Description
CMD	The command code, a single ASCII character.
Param_1, 2, n	The parameters for the particular command code.
	A space character (0x20).

The number of parameters in a command varies between different commands, as per the command definition.

Space-separated commands are limited to video commands (page 38) and some older audio commands which are no longer documented.

Fixed Length

Fixed length commands do not use spaces to separate parameters. Instead the parameters are all of fixed length, as defined by the command definition.

Example 1:

CMD	Param_1	Param_2
-----	---------	---------

Example 2:

CMD	Param_1	Param_2	Param_3
-----	---------	---------	---------

The command may use just one ASCII character (as in some audio commands) or two ASCII characters, which are referred to as 'extended commands'.

With extended commands, the first character typically specifies a command family, while the second character identifies a sub-command.

In all cases parameter formatting depends on the command, and so details of each command should be checked.

Response Commands and Tallies

Enquire commands that require a 'response command' (or status response) normally cause command packets to be generated in the 'fixed length' format:

CMD	Param_1	Param_2	Param_3	Param_n
-----	---------	---------	---------	---------

In the above table, 'cmd' echoes the enquire command. The actual number and types of parameters returned depends on the command issued.

There are a few tallies ('1' and '3') which use the space-separated format.

String Parameters / Escaped Characters

Any string parameters are of variable length, and so are grouped at the end of the command packet. When multiple string parameters are required, they are delimited by vertical bar characters ('|') which therefore cannot appear in the parameter.

String parameters can contain 'escape' sequences to include special characters which cannot otherwise be transmitted using the Oxtel protocol:

- Colon '\3A'
- Semi-colon '\3B'
- Vertical bar '\7C'
- Backslash '\5C'

These special characters are handled using C-style escape sequences using a backslash character followed by two hexadecimal bytes to form the code for a single byte. Note that backslash itself must therefore be transmitted as '\5C'

This code converts normal ASCII strings into a format suitable for sending:

```
char * put_remote_str(char *instr, char *outstr)
{
    char c;
    for(;;)
    {
        c = *instr++;
```

```
if (c == 0)
{
    *outstr = 0;
    return ostr;
}

if ((c == ';' ) || (c == ':' ) ||
    (c == '|' ) || (c == '\\'))
{
    *outstr++ = '\\';
    sprintf(outstr, "%02x", c);
}
else
{
    *outstr++ = c;
}
}
```

Floating-Point Parameters

Some floating-point parameters are converted to 16:16 fixed-point integer representation, and sent as an 8-character hex field, so 3.5 would be sent as 00038000.

```
long oxfix(float f)
{
    return f * 0x10000;
}
...
send_command("ZZ%08x%08x", oxfox(PI), oxfix(1.23456));
...
```

[to Introduction](#)

Keying Layers

Layer Numbers

Throughout this protocol document, automation command specifications make reference to layer numbers for keyers. The number of layers available to automation depends on the product, as defined below. Please refer to individual product manuals for further information.

The first keying layer keys graphics over the background video signal using a fill signal (graphic) and key signal (transparency). The resulting signal is then passed onto the next keying layer which keys its graphics over the top, and so on until the resulting signal is output with all graphics layers present. Each keying layer will only key its graphics if it is cut up.

Imagestore 2/2U/3, 300[+] and HD-TV

Each keyer can either be assigned an internal media file or an external fill/key.

- 0x0 = Keyer 1 (DSK1)
- 0x1 = Keyer 2 (DSK2)

Imagestore 750

Each keyer can either be assigned an internal media file or an external fill/key.

- 0x0 = Keyer 1 (DSK1)
- 0x1 = Keyer 2 (DSK2)
- 0x2 = Keyer 1 (DSK3)
- 0x3 = Keyer 2 (DSK4)

DSK-3901

Each keyer can only be assigned an external fill/key.

- 0x0 = Keyer 1 (DSK1)
- 0x1 = Keyer 2 (DSK2)

LGK-3901 and ISM-3901

The LGK-3901 and ISM-3901 have 3 stores that can be assigned to any 3 of the 5 keyers. See the Xi command for more details. All keyers can be assigned an external fill/key.

- 0x0 = Keyer 1 (DSK1)
- 0x1 = Keyer 2 (DSK2)
- 0x2 = Keyer 1 (DSK3)

- 0x3 = Keyer 2 (DSK4)
- 0x4 = Keyer 2 (DSK5)

Intuition[+]

- 0x0 – 0xF = Layers 1 to 16.

Layer Modes

The layer parameter passed to some video commands can have different meanings depending on the unit being controlled, and its current operating mode.

Cascade Mode

Note: Cascade mode is the default and recommended mode of operation for all Imagestore devices. Cascade mode is the only mode of operation for all Intuition devices.

In cascade mode each keying layer is used to insert an independent image, animations, Easytext or external fill/key onto the PGM output.

Since one store is shared between PGM and PVW for each keying layer, it is not possible to preview new media while existing media is still on-air, and so a different layer has to be used for previewing. However, more keying layers are available in cascade mode and so this is usually possible.

In cascade mode, automation uses layer numbers which are zero-based:

- Layer 0 DSK1
- Layer 1 DSK2
- Layer 2 DSK3 (Imagestore 750, LGK-3901 only)
- Layer 3 DSK4 (Imagestore 750, LGK-3901 only)
- Layer 4 DSK5 (LGK-3901 only)

Imagestore 2/2U/3 devices have just two keying layers, which are sometimes referred to as mid-ground (DSK1) and foreground (DSK2).

Swap Preview Mode

In swap preview mode two stores are shared between PGM and PVW for a single keying layer only. It is possible to load and preview new media while existing media is still on-air on PGM. Once you are happy with the new media it can be swapped from PVW to PGM using a cut or fade.

The number of available keying layers is halved in swap preview mode because of the way that the stores are allocated.

In swap preview mode, automation uses layer numbers with the following meaning:

- Layer 0 PVW layer
- Layer 1 PGM layer

This allows independent control of both PGM and PVW outputs.

[to Introduction](#)

Automation Commands

Oxtel Automation commands are described in the sections below.

[Keying Layers](#) [Keyers](#) [Stores](#) [Animations](#)

[Easytext](#) [Datasources](#)

[Emergency Alert System](#)

[SDI Video](#) [A/B Mixer](#) [Video Preview](#)

[DVE](#)

[Audio](#) [Advanced Audio](#)

[Easyplay](#) [Easyplay 2](#)

[Dolby and Up-Mix](#) [Metadata](#)

[System and Status](#) [Health](#) [Scheduler](#)

[General Purpose Interface](#) [Macros](#)

[Master Control](#)

[Intuition XG commands](#)

[Command Validity Table](#)

[CRC Generation](#)

[Automation Examples](#)

[Command Index](#)

Keyers

Commands and Responses

0 Fade Keyer to Black 0

This command causes the total output from the selected keying layer to be faded to or from black.

If a prior fade is not finished when a reverse instruction is received, the fade will reverse direction and continue at the same rate.

The transition duration of the fade to/from black can be set using the B command, or by using the optional rate parameter (Imagestore 300[+], Imagestore HD-TV, Imagestore 750, LGK-3901 and DSK-3901).

The rate value sets the number of fields (interlaced) or frames (progressive) to complete the action. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal.

CMD	Param_1	Param_2	Param_3
0	%x: Layer	%x: Direction 0x0 = To black 0x1 = From black	%x: Rate 0x1 - 0x3E7 (optional)

Example:

```
void FadeToBlack(int Layer, bool Black)
{
    remote_send("0%x %x", Layer, Black ? 0:1);
}
```

1 Fade Keyer 1

This command causes the selected keying layer to be faded up or down. When the keyer is faded up it's contents are keyed over the background video using a fill and key. When the keyer is faded down, the background video is passed unchanged.

If a prior fade is not finished when a reverse instruction is received, the fade will reverse direction and continue at the same rate.

The transition duration of the fade keyer up/down can be set using the B command, or by using the optional rate parameter (Imagestore 300[+], Imagestore HD-TV, Imagestore 750, LGK-3901 and DSK-3901).

The rate value sets the number of fields (interlaced) or frames (progressive) to complete the action. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal.

CMD	Param_1	Param_2	Param_3
1	%x: Layer	%x: Direction 0x0 = Down 0x1 = Up	%x: Rate 0x1 - 0x3E7 (optional)

Example:

```
void FadeKeyer(int Layer, bool Up)
{
    remote_send("1%x %x", Layer, Up ? 1:0);
}
```

2 Cut Keyer to Black 2

This command causes the total output from the selected keying layer to be cut to or from black.

This is identical to an instant 'Fade to Black / Fade from Black' command with rate of 1 field/frame.

CMD	Param_1	Param_2
2	%x: Layer	%x: Direction 0x0 = To black 0x1 = From black

Example:

```
void CutToBlack(int Layer, bool Black)
{
    remote_send("2%x %x", Layer, Black ? 0:1);
}
```

3 Cut Keyer 3

This command causes the selected keying layer to be cut up or down. When the keyer is cut up its contents are keyed over the background video using a fill and key. When the keyer is cut down, the background video is passed unchanged.

CMD	Param_1	Param_2
3	%x: Layer	%x: Direction 0x0 = Down 0x1 = Up

Example:

```
void CutKeyer(int Layer, bool Up)
{
    remote_send("3%x %x", Layer, Up ? 1:0);
}
```

This is identical to an instant 'Fade Keyer Up / Fade Keyer Down' command with rate of 1 field/frame.

4 Set Layer Mask 4

This command sets the dimensions of the mask for the image that is loaded into the store for the specified keying layer.

The left and right values range from 0 to the width of the loaded image. Similarly the top and bottom values range from 0 to the height of the loaded image.

In cascade mode, the mask values may be set for all layers. In swap-preview mode, only the preview layer mask values may be set.

CMD	Param_1	Param_2	Param_3	Param_4
4	%x: Layer	%x: Left 0x0 - width	%x: Right 0x0 - width	%x: Top 0x0 - height

Param_5	Param_6
%x: Bottom 0x0 - height	%x: Speed - (optional)

The speed parameter is optional for Imagestore 2/2U/3 and is measured in pixels/lines per field.

Example:

```
void SetMask(int Layer, int Left, int Right,
             int Top, int Bottom)
{
    remote_send("4%x %x %x %x %x", Layer, Left,
               Right, Top, Bottom);
}
```

5 Layer Mask Enable 5

This command enables or disables the mask for the image that is loaded into the store for the specified layer. The mask uses the dimensions that are set by the 4 command.

In cascade mode, the mask may be enabled for all layers. In swap-preview mode, only the preview layer mask may be enabled.

CMD	Param_1	Param_2
5	%x: Layer	%x: Enable mask 0 = Disable 1 = Enable

Example:

```
void MaskEnable(int Layer, bool Enable)
{
    remote_send("5%x %d", Layer, Enable);
}
```

6 Swap PVW and PGM Image using Cut 6

This command is also known as “Cut Swap” and is only relevant to Imagestore devices used in swap-preview mode.

Note: Cascade mode is recommended instead of swap-preview mode. This command is largely deprecated.

When executed, the PGM keyer is cut down (if required), the PVW store is then swapped with the PGM store, and the PGM keyer output is cut back up to maximum. The switching action takes place within the vertical blanking interval and so there is no perceived delay between the cut down and the cut up.

CMD
6

7 Swap PVW and PGM Image using V-fade 7

This command is also known as “Fade Swap” and is only relevant to Imagestore devices used in swap-preview mode.

Note: Cascade mode is recommended instead of swap-preview mode. This command is largely deprecated.

When executed, the PGM keyer is faded down (if required), the PVW store is then swapped with the PGM store, and the PGM keyer output is faded back up to maximum.

CMD
7

B Set Transition Duration B

This command sets the duration for the selected transaction type on the selected layer.

The type parameter defines which transition (FTB or Keyer) is to be affected.

The rate value sets the number of fields (interlaced) or frames (progressive) to complete the fader transition. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal.

In swap-preview mode, the value applies to the PGM layer, whatever layer number is selected.

CMD	Param_1	Param_2	Param_3
B	%x: Layer	%x: Type 0x0 = FTB 0x1 = Fade	%x: Rate 0x0 - 0x3E7

Example 1: Set transition rate.

```
void SetTransRate(int Layer, int Fields)
{
    remote_send("B%x 1 %x", Layer, Fields);
}
```

Example 2: Set Fade-To-Black transition.

```
void SetFTBTransRate(int Layer, int Fields)
{
    remote_send("B%x 0 %x", Layer, Fields);
}
```

C Set Self Key, Separate Key or No Key C

This command sets the keying type for the selected layer. To self, separate or no key. It can apply to a keying layer being fed graphics by an internal store or external fill/key.

Oxtel products implement a keying scheme where the Key signal for a layer is derived from the luminance content (Y value) of an SDI signal as follows:

- Self key Derived from the luminance of the Fill signal.
- Separate key Derived from the luminance of a separate Key.
- No key Entire fill becomes visible without transparency.

Note: In swap-preview mode this value is associated with the preview store. The value will remain with the image as it is stored or transferred to the program channel.

CMD	Param_1	Param_2
C	%x: Layer	%x: Key mode 0x0 = Separate Key 0x1 = Self Key 0x2 = No Key

Example:

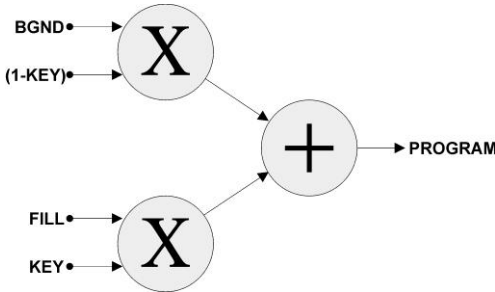
```
void SetKeyType(int Layer, int Type)
{
    remote_send("C%x %x", Input, Type);
}
```

D Set Key Linear or Full D

This command sets the keying mode for the selected layer to linear or full keying. It can apply to a keying layer being fed graphics by an internal store or external fill/key.

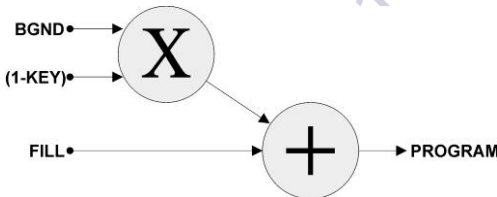
Full key refers to the standard keying algorithm where the Fill signal is multiplied by the Key signal prior to adding it to the background video signal.

The equation for this is: $\text{Output} = \text{Key} \times \text{Fill} + (1 - \text{Key}) \times \text{Background}$



Linear key is more typically called *Additive Keying*. This requires the foreground fill image to be 'pre-composited' onto a black matte.

The equation for this is: $\text{Output} = \text{Fill} + (1 - \text{Key}) \times \text{Background}$



CMD	Param_1	Param_2
D	Layer (%x)	Key type (%x) 0x0 = Full key 0x1 = Linear key

Example:

```

void SetKeyMode(int Layer, int Mode)
{
    remote_send("D%x %x", Input, Mode);
}
  
```


E Set Key Normal or Invert E

This command sets the keying sense for the selected layer to normal or inverted keying. It can apply to a keying layer being fed graphics by an internal store or external fill/key.

Normal key implies that increasing key values will introduce more of the fill signal. Key invert implies that increasing key values introduce less fill. The Key output is replaced with (1-Key), thereby inverting the action of the Key.

CMD	Param_1	Param_2
E	%x: Layer	%x: Key sense 0x0 = Normal 0x1 = Invert

Example:

```
void SetKeySense(int Layer, int Sense)
{
    remote_send("E%x %x", Input, Sense);
}
```

F Set Clip, Gain and Transparency F

This command sets the clip, gain and transparency values for a particular keyer layer. The parameter values are hexadecimal representations of unsigned 10-bit luminance values.

Clip is the lowest value of key signal which will cause the fill signal to contribute to the output. In a keying process this will result in a total absence of fill video in areas where the incoming key signal is less than the clip level.

Gain sets the level of key signal above which 100% of the fill signal is used. In a keying process this will result in the fill signal being less transparent than it otherwise would be in areas with a grey key signal.

Transparency controls the luminance content (grey scale value) of the key signal and therefore the amount of background video that will 'seep' through the keyed image.

Note: In swap-preview mode this value is associated with the preview store. The value will remain with the image as it is stored or transferred to the program channel.

CMD	Param_1	Param_2	Param_3	Param_4
F	%x: Layer	%x: Clip 0x0 - 0x3FF	%x: Gain 0x0 - 0x3FF	%x: Trans 0x0 - 0x200

Example:

```
void SetCGT(int Layer, int C, int G, int T)
{
    remote_send("F%x %x %x %x", Layer, C, G, T);
}
```

Note: For optimum results the clip, gain and transparency values should be set to 64, 940 and 512 respectively. For all products apart from Imagestore 2/2U/3, clip values set between 0-63 and gain values set between 941-1023 are clamped to 64 and 940 respectively. Clip can never be greater than gain.

F Enquire Clip, Gain and Transparency F

This command enquires the status of the clip, gain and transparency for the specified layer.

CMD	Param_1
F	%x: Layer

This will return:

CMD	Param_1	Param_2	Param_3	Param_4
F	%x: Layer	%x: Clip 0x0 - 0x3FF	%x: Gain 0x0 - 0x3FF	%x: Trans 0x0 - 0x200

Unsolicited Tallies

3 Keyer Position Tally 3

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about PGM keyer positions as keyers are cut or faded up/down using the 3 and 1 commands.

On registration, tallies for all layers are returned so that automation can record the initial keyer position state.

The format of the tally is:

CMD	Param_1	Param_2
3	%1x: Layer	%1x: Keyer position 0x0 = Off (down) 0x1 = On (up) 0x2 = In between

Example:

```

Y61          // Enable video tallies
-> 30 1      // Initial state DSK 1 is cut up
-> 31 0      // Initial state DSK 2 is cut down
-> 32 0      // Initial state DSK 3 is cut down
-> 33 0      // Initial state DSK 4 is cut down

30 0        // Cut DSK 1 down
-> 30 0      // DSK 2 is cut down

11 1        // Fade DSK 2 up
-> 31 2      // DSK 2 is in transition
-> 31 1      // DSK 2 is cut up

```

Note: The Y6 tally (see page 141) returns keyer position state for DSK1 and DSK2. The 3 tally is therefore only implemented on products with more than two keying layers – (Intuition, Imagestore 750 and LGK-3901), but the Y6 tally is still used for A/B mixer position.

Yc Layer Rate Tally Yc

Note: This command is only valid for Imagestore 2/2U/3.

This tally is enabled with the 'Enable Passive Mode Tallies' (Ya1) command.

Once enabled, it returns information whenever one of the layer fade rates or FTB rates changes.

The format of the tally is:

CMD	Param_1	Param_2
Yc	%03x: DSK1 fade rate 0x000 - 0x3E7	%03x: DSK2 fade rate 0x000 - 0x3E7

Param_3	Param_4
%03x: DSK1 FTB rate 0x000 - 0x3E7	%03x: DSK2 FTB rate 0x000 - 0x3E7

Yf Fade-To-Black Position Tally Yf

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about PGM keyer fade-to-black positions as keyers are cut or faded to/from black using the 2 and 0 commands.

On registration, tallies for all layers are returned so that automation can record the initial keyer position state.

The format of the tally is:

CMD	Param_1	Param_2
Yf	%1x: Layer	%1x: Fade-to-Black Position 0x0 = On 0x1 = Off 0x2 = In between

Example:

```

Y61          // Enable video tallies
-> Yf01      // Initial state DSK 1 is not black
-> Yf11      // Initial state DSK 2 is not black
-> Yf21      // Initial state DSK 3 is not black
-> Yf31      // Initial state DSK 4 is not black

23 0        // Cut DSK 4 to black

```

```
-> Yf30    // DSK 4 is black  
  
03 1      // Fade DSK 4 from black  
-> Yf32    // DSK 4 (FTB) is in transition  
-> Yf31    // DSK 4 is not black
```

Note: The Y6 tally (see page 141) returns FTB position state for DSK1 and DSK2. The Yf tally is therefore only implemented on products with more than two keying layers – (Intuition, Imagestore 750 and LGK-3901), but the Y6 tally is still used for A/B mixer position.

Stores

Imagestore devices copy media files from hard disk (or compact flash) into dedicated store memory before graphic are keyed onto the video output. Each keying layer has a store associated with it for such media.

Oxtel devices can reference media files as either:

- Named images (preferably with file extension).
- Numbered images.

Note: Named images are the preferred type since they provide more flexibility than numbered images.

Note:

Recommend using named images with file extension

Named Images

The following file types are supported for named images:

- OXT Still images
- OXA Animations, Easytext (crawls), or digital/analogue clocks
- TEM Intuition[+] templates
- VAF Intuition XG scenes

To load and save named files (the preferred option), use the “R” family of filename-based extended commands.

Media filenames are alphanumeric and limited to 39 characters in length (including the file extension).

File extensions should always be included within automation commands to ensure that the correct file is used. Filenames should contain only the characters A-Z, a-z, 0-9 and _ (underscore). Spaces should be avoided.

There is inconsistency in case-sensitivity between different Oxtel Series products (and even some software versions). It is therefore recommended that media filenames are never duplicated with different cases, and that automation commands maintain case when referencing media filenames.

Numbered Images

Note: Use of numbered images is not recommended.

The file name of a numbered image is constructed in the form 'V_{nnn}.OXT' or 'V_{nnn}.OXA', where the _{nnn} is the specified decimal number.

When sent to Intuition SD/HD[+], a numbered file name is constructed in the form 'V_{nnn}.TEM', where the _{nnn} is the specified decimal number.

If the specified decimal number has 4 digits, then the file name is constructed in the form 'V_{xxxx}.OXT', 'V_{xxxx}.OXA' or 'V_{xxxx}.TEM'.

Note: Some of the commands listed in this section only work for numbered images.

Commands and Responses

R0 Load Image R0

This command allows media files to be loaded into a keying layer prior to being cut or faded up on air. Media types include static images, animations, text straps, digital/analogue clocks, or multi-object templates. Please refer to documentation for individual products to determine which media types are supported.

Imagestore products load media files into a store associated with a keying layer; which means that load times vary depending on the size of animation media. Once media has completed loading, the associated keying layer can be cut or faded up to reveal the graphics on-air. If the keying layer was already faded up then the media will only replace the previous graphics once the media load completes.

Intuition products do not suffer from the variable load time of Imagestores since animations and MPEG clips are streamed directly from disk instead of to stores.

CMD	Param_1	Param_2
R0	%1x: Layer	%s: Filename

The following file types are supported by the R0 command:

- OXT Still images
- OXA Animations, Easytext (crawls), or digital/analogue clocks
- TEM Intuition[+] templates
- VAF Intuition XG scenes

Note: The R0 command supersedes the command 8, which remains supported. Unlike the 8 command, the image is loaded by filename rather than by number.

Example 1:

```
void LoadEmergencySlide(int Layer)
{
    remote_send("R0%1x%s", Layer, "Emergency.oxa");
}
```


Example 2:

```
void LoadNumberedImage(int Layer, int num)
{
    // Equivalent to remote_send("8%x %x", Layer, num);
    remote_send("R0%1xV%03d", Layer, num);
}
```

R0 Enquire Load Image R0

This command enquires which media file is currently loaded into the store for the specified keying layer.

Note: To unload an image, use the 'A' command.

CMD	Param_1
R0	%1x: Layer

Example:

```
void EnquireImageLoaded(int Layer)
{
    remote_send("R0%1x", Layer);
}
```

The information returned is:

CMD	Param_1	Param_2
R0	%1x: Layer	%s: Filename

Note: When no media is loaded, the filename will be '> Empty <'. When an external fill/key is active for the keying layer, the filename will be '> F1+K1 <', '> F2+K2 <', etc.

Note: When video tallies (Y61) are enabled on a connection, there will be an 'R0' tally issued every time a store load completes.

```
# Assume we start with video tallies enabled (Y61)

R00map.oxt          # Load map.oxt into layer 0 (DSK1)
-> R00map.oxt       # Tally confirming load completed

R00emergency.oxa    # Load map.oxt into layer 0 (DSK1)
-> R00emergency.oxa # Tally confirming load completed

A0                  # Unload layer 0 (DSK1)
```

```
→ R00> Empty < # Tally confirming unload
```

R3 Enquire File Info R3

This command enquires information on whether the specified file exists (or not) in the media library on the device's internal disk.

Note: Use the R8 command to query the existence of audio files.

CMD	Param_1
R3	%s: Filename

Note: This command supersedes the P command, which remains supported.

Example:

```
void EnquireEmergencySlide()
{
    remote_send("R3%s", "Emergency.oxa");
}
```

The information returned is:

CMD	Param_1	Param_2
R3	%1x: File exists 0x0 = No 0x1 = Yes	%s: Filename

R4 Query First File R4

This command enquires the name of the first file within the specified media folder.

This is the first step automation takes if it wishes to enumerate the media files currently on the system, followed by the R5 enquire command which retrieves names of subsequent files.

The ordering of the filenames retrieved is alphabetical.

CMD	Param_1
R4	%s: Folder name alias \$VIDEO = Image folder \$AUDIO = Audio folder \$FONTS = Font folder

Example:

```
void QueryFirstFile ()
{
    remote_send("R4%s", "$VIDEO");
}
```

The information returned is:

CMD	Param_1	Param_2
R4	%1x: End of directory reached 0x0 = No 0x1 = Yes	%s: Filename

'Filename' is the name of the file, "unknown" if the folder alias does not exist, or "end" if the end of the folder is reached.

R5 Query Subsequent File R5

This command is used repeatedly to enumerate the names of files within the specified media folder.

It should be used following a single instance of the R4 enquire command using the same file type.

Once all files have been enumerated, subsequent changes to media files can be tracked using YB tallies. This method is used by automation and panels. Media management systems use alternative methods for tracking media on the system.

The ordering of the filenames retrieved is alphabetical.

CMD	Param_1
R5	%s: Folder name alias \$VIDEO = Image folder \$AUDIO = Audio folder \$FONTS = Font folder

The information returned is in exactly the same format as for the 'R4' command, but with 'R5' as the command code.

CMD	Param_1	Param_2
R5	%1x: End of directory reached 0x0 = No 0x1 = Yes	%s: Filename

'Filename' is the name of the file, "unknown" if the folder alias does not exist, or "end" if the end of the folder is reached.

R6 Enquire Extended File Information R6

This command enquires for information about a specified media file from the internal disk of the Imagestore device,

CMD	Param_1
R6	%s: Filename

Example:

```
void EnquireEmergencySlide()
{
    remote_send("R6%s", "Emergency.oxa");
}
```

The information returned is:

Format	Field	Description
%c%c	CMD	'R6'
%1x	File exists	0x0 = No 0x1 = Yes
%03x	X-position	Horizontal position of the image
%03x	Y-position	Vertical position of the image
%03x	Width	Width of the image

%03x	Height	Height of the image
%03x	Clip	Clip value of the image
%03x	Gain	Gain value of the image
%03x	Transparency	Transparency value of the image
%02x	Image Type	0x01 = Still (OXT) 0x02 = Animation (OXA) 0x04 = Easytext (OXA) 0x05 = Easytext crawl or roll (OXA) 0x08 = Clock (OXA)
%04x	Frames	Number of frames (animations)
%1x	Animation mode	Animation play mode (animations): 0x0 = Cycle 0x1 = Single shot 0x2 = In loop out 0x3 = Linear control 0x4 = Ping pong 0x5 = Multi loop
%02x	Load time	Estimated time to load image in seconds
%1x	Associated audio	Associated audio file present: 0x0 = No 0x1 = Yes
%s	Filename	Confirmation of filename

Example of parsing the information returned (using C).

```

sscanf(&buff[1],
"R6%1x%03x%03x%03x%03x%03x%03x%02x%04x%1x%02x%1x%s",
&Info->file_exists, &Info->xpos, &Info->ypos,
&Info->width, &Info->height, &Info->clip, &Info->gain,
&Info->trans, &Info->image_type, &Info->no_frames,
&Info->anim_mode, &Info->load_time,
&Info->associated_audio, Info->FileName);

```

Note: Most return parameters have no meaning for .TEM or .VAF files, and so Intuition products should only use this command for determining file existence.

R7 Preload Image R7

This command pre-loads (or pre-rolls) a media file (typically an animation) into the specified layer. Once the pre-load has completed, the media can be cut-swapped with the on-air media in a frame-accurate manner.

The media file is copied into an off-screen (unused/free) area of the video store memory. When a subsequent R0 image load command is issued with the same filename, the pre-loaded image is immediately swapped onto the layer. If a different filename is requested in the R0 image load command that follows, then the pre-loaded image is cleared.

Note: The store memory must be large enough to store both the current (on-air) animation and the next (pre-loaded) animation. If this is not the case then the R7 command will fail. Larger video stores memory or smaller animations will address this problem.

The progress of the preload can be monitored by checking the "disk_busy" bit returned by the O enquire command - (part of the key_type parameter).

When using the R7 command the new media can not be previewed prior to it being shown on-air.

Note: On Imagestore 2/2U/3 products, there is a single process which handles all file loads, saves, deletes and pre-loads in strict order of reception. In other words, if a load is in progress on layer 0, load (or save, etc.) requests on another layer will be deferred until the first load completes.

CMD	Param_1	Param_2
R7	%1x: Layer	%s: Filename

Example 1:

```
void PreloadEmergencySlide(int Layer)
{
    remote_send("R7%1x%s", Layer, "Emergency.oxa");
}
```

R9 Emergency to Air R9

This command automatically displays a still image named V000.OXT.

This emergency-to-air image appears on the most downstream keying layer (with associated store) so that it is seen on-top-of the background video, other graphics and DVE effects.

When emergency-to-air is enabled, the name of the previous media is saved, the emergency-to-air image (V000.OXT) is loaded and the keyer is cut up (if required). When emergency-to-air is disabled, the previous media is restored and the keyer is cut down.

CMD	Param_1
R9	%1x: Emergency to air 0x0 = Off 0x1 = On

Example:

```
void EmergencyToAir(bool onOff)
{
    remote_send("R9%1x", onOff);
}
```

Ra Enquire Image Count Ra

Note: This command only applies to Imagestore 2/2U/3.

This command enquires how many images are currently used up, and the maximum available number of images that can be saved to the hard disk.

CMD
Ra

The response command has the following format:

CMD	Param_1	Param_2
Ra	%04x: Images used	%04x: Maximum images

Rm Image Load Mode Rm

This command defines behaviour of a keying layer when new media is loaded (via the R0 command) over existing visible media. This applies to all media types including images, animations, Easytext straps, and clocks.

Clean load mode ensures that the keyer is cut down before the new media starts loading. Once the load completes the keyer is cut up again to reveal the new media.

Cut load mode ensures that the old media is held on-air while the new media starts loading. Once the load completes the images are swapped over. This is similar to the R7 pre-load image command, except that the cut-swap is not

field-accurate, particularly for animations which have a finite load time that depends on the file size.

CMD	Param_1	Param_2
Rm	%1x: Layer	%1x: Image load mode 0x0 = Clean 0x1 = Cut

Example:

```
void ImageLoadMode(int Layer, int Mode)
{
    remote_send("Rm%1x%x", Layer, Mode);
}
```

Rm Enquire Image Load Mode Rm

This enquire command is used to discover the behaviour of a keying layer when new media is loaded (via the R0 command) over existing visible media.

CMD	Param_1
Rm	%1x: Layer

Example:

```
void EnquireImageLoadMode(int Layer)
{
    remote_send("Rm%1x", Layer);
}
```

The information returned is:

CMD	Param_1	Param_2
Rm	%1x: Layer	%1x: Image load mode 0x0 = Clean 0x1 = Cut

RA Validate Template RA

The RA enquire command allows all media assets referenced by an Intuition template to be validated.

Intuition templates typically reference many other media files; including OXT images, and OXI animations. It is important to confirm that all of these media references exist and are valid before bringing the template to air.

CMD	Param_1
RA	%s: Filename

The information returned is:

Format	Field	Description
%c%c	CMD	RA
%s	Template name	Filename (followed by pipe separator)
%1x	File exists	0x0 = No 0x1 = Yes
%04x	Missing assets	Number of missing assets Filenames in the subsequent list
%s %s...	Assets file list	Piped list of missing media filenames

The following are examples of information returned by the RA command for three different Intuition templates:

```
# banner1.tem exists, and has no missing assets
RAbanner1.tem
→ RAbanner1.tem|10000

# banner2.tem exists, but has two missing assets
RAbanner2.tem
→ RAbanner2.tem|10002|logo.oxi|bug.oxt

# banner3.tem does not exist
RAbanner3.tem
→ RAbanner3.tem|00000
```

Any valid templates with missing assets should have the assets replaced by media management.

Any missing templates should be replaced by media management, and then the RA enquire command should be resent to confirm whether all related media is present.

Note: The RA command is only supported by Intuition SD/HD[+] and Intuition XG.

RC Enquire Number of Media Files RC

This command enquires for the number of files located in the specified folder.

CMD	Param_1
RC	%s: Folder name alias \$VIDEO = Image folder \$AUDIO = Audio folder \$FONTS = Font folder

The response gives a count for the number of files:

CMD	Param_1
RC	%04x: File count

A Erase Store A

This command empties the store associated with a keying layer.

Both the fill and key parts of the store are filled with the colour black, and an '> Empty <' image is selected.

CMD	Param_1
A	%x: Layer

Example:

```
void EraseStore(int Layer)
{
    remote_send("A%x", Layer);
}
```

G Set Image Position G

This command sets the position of the loaded image relative to the viewing screen.

The origin is the upper left-hand corner, and positive increments move the picture to the right and downwards. Images are positioned onto the nearest even pixel steps, but can not be positioned onto odd pixels.

Pictures may be placed off screen in all four directions. The ranges for different standards are:

- PAL (625) ± 720 horizontally, ± 576 vertically

- NTSC (525) ± 720 horizontally, ± 486 vertically
- 720p ± 1280 horizontally, ± 720 vertically
- 1080i ± 1920 horizontally, ± 1080 vertically
- 1080p ± 1920 horizontally, ± 1080 vertically

CMD	Param_1	Param_2	Param_3
G	%x: Layer	%x: Horizontal	%x: Vertical

Param_4
%x: Speed - (optional)

The speed parameter is optional for Imagestore 2/2U/3 and is measured in pixels/lines per field.

Example:

```
void SetImagePosition(int Layer, int Hor, int Ver)
{
    remote_send("G%x %x %x 0", Layer, Hor, Ver);
}
```

G Enquire Image Position G

This command enquires the image position for the specified layer.

CMD	Param_1
G	%x: Layer

Example:

```
void EnquireImagePosition(int Layer)
{
    remote_send("G%x", Layer);
}
```

The Oxtel device will return the following command to automation:

CMD	Param_1	Param_2	Param_3
G	%x: Layer	%x: Horizontal	%x: Vertical

H Load Live Video H

This command loads (or unloads) a live video feed into the specified keying layer from a pair of SDI inputs that provide fill and key signals.

CMD	Param_1	Param_2
H	%x: Layer	%x: Video feed 0x0 = Unload Live 0x1 = Load Live (F1 / K1) 0x2 = Load Live (F2 / K2) 0x3 = Load Live (F3 / K3)

Loading any image file into the layer's store will implicitly unload live video.

Please check individual product manuals to determine how many external fill/key SDI inputs are supported.

Example:

```
void LoadLive(int Layer, int Feed)
{
    remote_send("H%x %x", Layer, Feed);
}
```

H Enquire Live Video H

This command enquires which live video fill/key signals (if any) feed the specified keying layer.

CMD	Param_1
H	%x: Layer

The response command has the following format:

CMD	Param_1	Param_2
H	%x: Layer	%x: Video feed 0x0 = Unload Live 0x1 = Load Live (F1 / K1) 0x2 = Load Live (F2 / K2) 0x3 = Load Live (F3 / K3)

O Enquire Image Status O

This command returns a status block about the image loaded onto the store for the specified layer.

CMD	Param_1
O	%x: Layer

Example:

```
void EnquireLoadedImageStatus(int Layer)
{
    remote_send("O%x", Layer);
}
```

The information returned is:

Format	Field	Description
%c	CMD	'O'
%03x	Clip	
%03x	Gain	
%03x	Transparency	
%03x	Horizontal position	Image X-position
%03x	Vertical position	Image Y-position
%03x	Horizontal size	Image width

Format	Field	Description
%03x	Vertical size	Image height
%03x	Mask top	
%03x	Mask bottom	
%03x	Mask left	
%03x	Mask right	
%03x	File slot number	For numbered images, when loaded
%c	Key type	0x40 OR'd with... 0x01 for self key 0x02 for linear key 0x04 for invert 0x08 for disk busy (no longer supported) 0x10 for "key source = none"
%1x	Masked	
%1x	(reserved)	Always zero
%1x	Easypay playing	0x0 = not playing, 0x1 = playing
%1x	Layer	Layer number

Note: Unsolicited O command responses on state change can be requested via the YO tally (see page 76).

Xi Set Store-Keyer Configuration Xi

This command sets the store-to-keyer configuration.

This command is only supported on the LGK-3901 and ISM-3901 which have more keying layers than stores. In the configurations shown below, the keying layers followed by 'S' are assigned a store.

CMD	Param_1
Xi	%1x: Store-keyer configuration 0x0 = 1S 2S 3S 4 5 0x1 = 1S 2S 3 4S 5 0x2 = 1S 2S 3 4 5S 0x3 = 1S 2 3S 4S 5 0x4 = 1S 2 3S 4 5S 0x5 = 1S 2 3 4S 5S 0x6 = 1 2S 3S 4S 5

0x7	=	1	2S	3S	4	5S
0x8	=	1	2S	3	4S	5S
0x9	=	1	2	3S	4S	5S

Note: Changing the store-keyer configuration unloads all stores. The store-keyer configuration cannot be changed when EAS is active.

Xi Enquire Store-Keyer Configuration Xi

This command enquires the current store-keyer configuration.

CMD
Xi

The store-to-keyer configuration is returned in the response command:

CMD	Param_1
Xi	%1x: Store-keyer configuration
	0x0 = 1S 2S 3S 4 5
	0x1 = 1S 2S 3 4S 5
	0x2 = 1S 2S 3 4 5S
	0x3 = 1S 2 3S 4S 5
	0x4 = 1S 2 3S 4 5S
	0x5 = 1S 2 3 4S 5S
	0x6 = 1 2S 3S 4S 5
	0x7 = 1 2S 3S 4 5S
	0x8 = 1 2S 3 4S 5S
	0x9 = 1 2 3S 4S 5S

Unsolicited Tallies

Y9 Image Load Tally Y9

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about media filenames as they are loaded into the stores for keying layers using R0, H or A commands.

On registration, tallies for all layers are returned so that automation can record the initial store load state.

The format of the tally is:

CMD	Param_1	Param_2
Y9	%1x: Layer	%s: Image filename

Note: When a layer is unloaded, the image filename will be "> Empty <".

YA Image Preload Tally YA

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about media filenames as they are pre-loaded into the stores for keying layers using the R7 command.

The format of the tally is:

CMD	Param_1	Param_2
YA	%1x: Layer	%s: Image filename

YB Media Tally YB

This tally is enabled with the 'Enable Media Tallies' (YB1) command.

Once enabled, it returns information about media filenames on the device's hard disk (or compact flash) as they are added, deleted and renamed by media management.

Automation should use the R4 and R5 commands to enumerate through the media file list, and then use media tallies to track subsequent changes.

The format of the tally is:

CMD	Param_1	Param_2
YB	%06x: Media type 0x000001 = Images 0x000002 = Fonts 0x000004 = Sounds 0x000008 = Mixfiles 0x000010 = DVE sequences 0x000020 = Metadata presets 0x000040 = Configurations 0x000080 = Shuffle presets 0x000100 = Gain presets 0x000200 = Master Control configuration 0x000400 = Salvo configuration	%1x: Action 0x0 = Deleted 0x1 = Added 0x2 = Modified

Param_3
%s: Media filename

The first parameter (media type) reports the media file type. This can not be a bitwise combination of media types since the tally is for a single media file.

The second parameter (action) defines whether the image was added, deleted or modified (over-written) within the file system.

The third parameter (media filename) defines the media file name for which a change took place.

Note: If media files are renamed then two tallies will be received; one for the deletion, and one for the addition (or modification).

YB Enable Media Tallies YB

This command enables or disables media tallies for the connection on which the command was received.

Media tallies are used to track media management as files are added, deleted or modified on the file system or image library.

CMD	Param_1
-----	---------

YB	%06x: Media type (bitwise) 0x000001 = Images 0x000002 = Fonts 0x000004 = Sounds 0x000008 = Mixfiles 0x000010 = DVE sequences 0x000020 = Metadata presets 0x000040 = Configurations 0x000080 = Shuffle presets 0x000100 = Gain presets 0x000200 = Master Control configurations 0x000400 = Salvo configuration
----	--

Notes on media types supported by different products:

- This command is not supported by Imagestore 2/2U/3.
- 'Sounds' relate to Easyplay files, and are not supported by Intuition.
- 'Mixfiles' are not supported by Intuition, DSK-3901, LGK-3901 or ISM-3901.
- 'DVE sequences' are not supported by Intuition, DSK-3901, LGK-3901 or ISM-3901.
- 'Metadata presets' only apply to Imagestore 750 (v2.0 or above).
- 'Configurations', 'Shuffle presets' and 'Gain presets' only apply to Imagestore 750, DSK-3901, LGK-3901 and ISM-3901.
- 'Master Control configurations' and 'Salvo configurations' only apply to Imagestore 750 and ISM-3901 with Master Control option.

The bitwise parameter defines a combination of media type for which tallies are required. If the file type parameter is set as 0x000000 then all media tallies are disabled.

The following example registers for tallies when either images or fonts are added or removed:

```
void EnableMediaTalliesImagesFonts ()
{
    remote_send("YB000003");
}
```

After media tallies are enabled, unsolicited tallies are issued every time a matching media file is added, deleted or modified.

YB Enquire Media Tallies YB

This command enquires the status of the YB tallies for the connection:

CMD
YB

The format of the response command is:

CMD	Param_1
YB	%06x: Media type (bitwise) As defined above

YG Load Status Tally YG

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about the start of store loads, plus any store loads that fail.

When a store load starts a 'load start' YG tally is sent by the device. If the media load succeeds then a 'image load' Y9 tally is sent on completion, but if the media load fails a 'load fail' YG tally is sent instead. This tally is designed to help panels represent store load state correctly, but can equally be used by automation to flag when a media load fails.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
YG	%1x: Layer	%1x: Flag 0x0 = Load start 0x1 = Load Fail	%s: Image filename

O Image Status Tally O

This tally is enabled with the 'Enable Image Status Tallies' (YO1) command.

Once enabled, it returns information about loaded images (CGT, position, mask, etc.) as they are updated by automation or panel users.

On registration, tallies for all layers are returned so that automation can record the initial state.

The format of the tally is:

Format	Field	Description
%c	CMD	'O'
%03x	Clip	
%03x	Gain	
%03x	Transparency	
%03x	Horizontal position	Image X-position
%03x	Vertical position	Image Y-position
%03x	Horizontal size	Image width
%03x	Vertical size	Image height
%03x	Mask top	
%03x	Mask bottom	
%03x	Mask left	
%03x	Mask right	
%03x	File slot number	For numbered images, when loaded
%c	Key type	0x40 OR'd with... 0x01 for self key 0x02 for linear key 0x04 for invert 0x08 for disk busy (no longer supported) 0x10 for "key source = none"
%1x	Masked	
%1x	(reserved)	Always zero
%1x	Easyplay playing	0x0 = Not playing, 0x1 = Playing
%1x	Layer	Layer number

YO Enable Image Status Tallies YO

This command enables 'image status' (O) tallies for the connection which received the command.

CMD	Param_1
YO	%1x: Enable tally 0x0 = No 0x1 = Yes

An 'O' command tally (page 75) is sent back to automation when this tally is enabled, and then further 'O' command tallies are sent out whenever any parameter contained in the 'O' response changes.

YO Enquire Image Status Tallies YO

This command queries whether 'image status' (O) tallies have been enabled for the connection which received the command.

CMD
YO

The format of the response command is:

CMD	Param_1
YO	%1x: Enable tally 0x0 = No 0x1 = Yes

Xi Store-Keyer Configuration Tally Xi

This tally is enabled with the 'Enable Store-Keyer Configuration Tallies' (YS1) command.

Once enabled, it returns information about the current store-to-keyer configuration.

On registration, a tally is returned so that automation can record the initial store-keyer configuration state.

The format of the tally is:

CMD	Param_1
Xi	%1x: Store-keyer configuration 0x0 = 1S 2S 3S 4 5 0x1 = 1S 2S 3 4S 5 0x2 = 1S 2S 3 4 5S 0x3 = 1S 2 3S 4S 5 0x4 = 1S 2 3S 4 5S 0x5 = 1S 2 3 4S 5S 0x6 = 1 2S 3S 4S 5 0x7 = 1 2S 3S 4 5S 0x8 = 1 2S 3 4S 5S 0x9 = 1 2 3S 4S 5S

- Note:** This command is only supported on the LGK-3901 and ISM-3901 which have more keying layers than stores. In the configurations show above, the keying layers followed by 'S' are assigned a store.
- Note:** Changing the store-keyer configuration unloads all stores. The store-keyer configuration cannot be changed when EAS is active.

YS Enable Store-Keyer Configuration Tallies

YS

This command enables 'store-keyer configuration' (Xi) tallies for the connection which received the command.

CMD	Param_1
YS	%1x: Enable tally 0x0 = No 0x1 = Yes

An 'Xi' command tally is sent back to automation when this tally is enabled, and then further 'Xi' command tallies are sent out whenever the store-to-keyer configuration changes subsequently.

YS Enquire Store-Keyer Configuration Tallies

YS

This command queries whether 'store-to-keyer configuration' (Xi) tallies have been enabled for the connection which received the command.

CMD
YS

The format of the response command is:

CMD	Param_1
YS	%1x: Enable tally 0x0 = No 0x1 = Yes

Animations

The animation control commands are used to control the playout of animation sequences which have been loaded into store memory using the 'Stores' commands. They can be used to control different types of animation including cycle, single-shot, in-loop-out, linear control, ping-pong and multi-loop.

On Intuition units, animations occupy a template box rather than a full layer. These commands imply a box number of 0. See also the Zf command on page 102.

Note: Animations are not supported by DSK-3901.

Commands and Responses

S0 Start Animation S0

This command starts or restarts an animation.

The animation starts playing from the current field (interlaced) or frame (progressive). To force the animation to play from the start, set the current field before issuing this command.

CMD	Param_1
S0	%1x : Layer

Example:

```
void StartAnimation(int layer)
{
    remote_send("S0%1x", layer);
}
```

S1 Stop Animation S1

This command causes an animation to stop playing on the specified layer.

Note: The equivalent command for Intuition[+] is 'Zf' (see page 102).

If the immediate flag is set, the animation halts immediately at the current frame. If the 'complete cycle' flag is set, the animation completes a cycle

before stopping. With in-loop-out animations, this may cause a special 'stop section' to execute.

Animations always stop showing two frames to form a field. If there is intra-field motion, the image will flicker.

CMD	Param_1	Param_2
S1	%1x: Layer	%1x: Stop flag 0x0 = Complete cycle, then stop 0x1 = Immediate stop

Example:

```
void StopAnimationImmediate(int layer)
{
    remote_send("S1%1x1", layer);
}
```

S2 Select Animation Frame S2

This command sends the animation to the specified frame.

If the animation is running, the animation will continue to run from the specified frame. If the animation is stopped, then the specified frame will be displayed but the animation will remain halted.

CMD	Param_1	Param_2
S2	%1x: Layer	%4x: Field (interlaced) or frame (progressive)

Example:

```
void JumpToField(int layer, int field)
{
    remote_send("S2%1x%04x", layer, field);
}
```

S4 Restart Animation S4

This command restarts an animation, playing it out from the first field or frame.

This command is a convenient alternative to the S0 command when the animation has to always start from the first field.

CMD	Param_1
S4	%1x: Layer

Example:

```
void ReStartAnimation(int layer)
{
    remote_send("S4%lx", layer);
}
```

Timers

These commands allow the control of the timer features of a digital or analogue clock image, for units where the 'Bugclock' option is purchased.

Clocks can be loaded, positioned, keyed, etc. using the 'Stores' commands for stills and animations (R0, G, 3, etc.).

Commands

T0 Stop Timer T0

This command pauses a clock timer on the specified layer.

CMD	Param_1
T0	%1x : Layer

Example:

```
void StopTimer(int layer)
{
    remote_send("T0%1x", layer);
}
```

T1 Start Timer T1

This command restarts a clock timer that was previously stopped on the specified layer.

CMD	Param_1
T1	%1x : Layer

Example:

```
void StartTimer(int layer)
{
    remote_send("T1%1x", layer);
}
```

T2 Reset Timer T2

This command resets the clock timer to its default time value on the specified layer. For count-up timers, this is normally zero.

CMD	Param_1
T2	%1x : Layer

Example:

```
void ResetTimer(int layer)
{
    remote_send("T2%1x", layer);
}
```

T3 Set Timer to Value T3

This command sets a clock timer to the specified value on the specified layer. Time-of-day clocks and time-of-day countdowns are not affected.

CMD	Param_1	Param_2	Param_3	Param_4
T3	%1x : Layer	%02x : Hour	%02x : Minute	%02x : Second

Example:

```
void SetTimer(int layer, int hour, int min, int sec);
{
    remote_send("T3%1x%02x%02x%02x", layer, hour,
                min, sec);
}
```

Easytext

Introduction

Easytext commands allow text stills and crawls to be generated and controlled. Please refer to individual product documentation to see which devices support Easytext. The Easytext licence always needs to be present.

Summary

Easytext templates are built for Imagestore products using the Textbuilder software application from the Media Conversion Software (MCS) suite. Templates consist of a number of different 'boxes', each of which can contain a piece of text in a single style, or alternatively a still image.

The number, positions, parameters and styles of these boxes are defined in the Easytext template, along with optional 'static' text – (the default text shown when the template is first loaded).

To bring text to air, automation performs the following three tasks:

- Load the Easytext template.
- Define new text to be displayed in boxes.
- Bring the template to air using the keyer controls.

Various problems may be found as this is attempted:

- Easytext template loads happen as a 'background' operation, and the 'ACK' returned by the Imagestore signifies that the load request was received correctly; not that it has completed.
- Because of this, text updates should not be sent until automation knows that the template load has completed. This can be determined by using Y9 tallies. It can also be determined using the 'disk busy' bit which is part of the 'Enquire Image Status' (O) command. Alternatively, a simple delay of one second will normally suffice.
- Once the Easytext template is loaded, new text strings can then be sent to populate it. Box numbers must be known in advance by automation to ensure that the new text is inserted in the correct locations. The rendering of new text into the template is comparatively slow, and it may be necessary to wait for the text to be rendered before cutting up the template and bringing it to air. Currently this is handled with a delay, which depends on the complexity of the template and the number of characters to be rendered.

The Z0 command is the main command for updating text strings, but datasources can also be used for updating strings via the m00 commands. Easytext commands take a parameter which defines which keying layer is being adjusted. Many Easytext commands also specify a box within the Easytext template. Box numbers can range from 0x00 to 0xFE. Box 0xFF is treated as a special case for the Z3 command.

Intuition[+] uses the same set of commands for controlling boxes within Intuition templates. Intuition templates are built using the Intuition Builder software application from the Media Conversion Software (MCS) suite. In addition to still images, Intuition templates allow animations and MPEG clips to be loaded within boxes.

International Character Set Support

Easytext uses Unicode rather than ASCII to represent characters. Unlike ASCII (which is limited in its scope to common western characters) Unicode is an encoding standard that encapsulates all the major scripts used throughout the world. Unicode characters are typically represented with either 16- or 32-bit codes, rather than the 8 bits of ASCII.

To transfer Unicode data into Easytext, a format known as UTF-8 is used. This encodes each Unicode code into a sequence of one or more 8-bit values. For common Latin characters there is a one-to-one mapping between UTF-8 and standard ASCII and the UTF-8 code for these characters is a single byte. Codes for other characters can vary in length between 2 and 4 bytes.

In the case of the Easytext automation command 'Update Text Field', it is therefore valid to use:

```
remote_send("Z0%1x%02x%x%s", Layer, BoxNo,
            Render, "Hello World");
```

However when characters outside the standard 7-bit ASCII character set are required (including special European and accented characters) the text must be converted to the UTF-8 format. Routines to perform this conversion can be found by searching on the Internet.

After encoding to UTF-8, the byte-stream must be processed to replace protocol-specific control codes with alternative coding. See page 36 for a list of Oxtel protocol special characters which must be escaped.

Easytext Style Control

Some Oxtel commands provide automation systems with functions for overriding template parameters, such as changing font sizes, font colours, and adjusting text box positions on screen.

Note: To render one or more updated parameters, the 'Render Box' command (Z3) has to be sent.

Since there are many parameters which can be altered for any one text box, the process of setting these parameters is normally separated from the process of 'rendering' the text. Most Easytext commands will not automatically cause the text to be rendered, but a separate 'Render Box' command (Z3) is provided. This allows multiple parameters to be set for a variety of boxes, and then for the individual box (or the complete template) to be re-rendered.

Example:

```
// Change box number 1 on layer 2 to Image.oxt
Z4201Image.oxt:
// Render the box so the above change take effect
Z3201:
```

Sending Colours via Automation

Some Easytext commands have colour parameters. All colours are sent as 6 bytes (%06x) in 'RRGGBB' format. Each triplet has the range of 0x00 to 0xFF.

Examples:

- Black 0x000000
- White 0xFFFFFFFF
- Red 0xFF0000
- Green 0x00FF00
- Blue 0x0000FF
- Mid-grey 0x808080

Easytext Straps

Further to stationary text boxes, Easytext also supports text rolls and crawls, which are together known as 'straps'.

Easytext straps can scroll horizontally (right-to-left or left-to-right) or vertically (top-to-bottom or bottom-to-top). On Imagestore devices, all boxes within a template scroll together as one, so it is not possible to mix static and scrolling elements without using additional keying layers or using the richer text capabilities of an Intuition XG.

Text strings always update as the strap 'wraps around' and restarts. Consequently there can be a significant delay between sending the text update message (Z0), and seeing that text on-air. Also there is no 'queue' for text strap updates. If a text update is sent before pending text has been displayed then the pending text will never be seen.

Commands and Responses (Settings)

Z0 Update Text Field Z0

This command updates the text in a specified text box. Formatting and style of the text is unchanged.

CMD	Param_1	Param_2	Param_3	Param_4
Z0	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Flags (bitwise) 0x1 = Render 0x2 = Append	%s: String

If the 'render' flag is set the specified text string will update on screen. This includes all changes to font, size, colour and position that have been specified since the last call to Render Box (Z3) or Update Text Field (Z0) with 'render' set.

If the 'render' flag is not set then changes will not appear on screen until the next call to Render Box (Z3) or Update Text Field (Z0) with 'render' set.

Intuition[+] does not use this bitwise flag so calls to Z0 will render immediately.

If the 'append' flag is set the specified text string is appended to the existing text. This allows long text strings to be defined over several command packets. The first of several command packets should have 'flags' set to 0x0 (do not render, do not append), middle packets should have 'flags' set to 0x2 (do not render, do append), and the final packet should have 'flags' set to 0x3 (do render, do append) to display the completed text string.

Example:

```
void UpdateTextBox(int Layer, int BoxNo,
                  int Flags, char* String)
{
    remote_send("Z0%1x%02x%1x%s", Layer, BoxNo,
               Flags, String);
}
```

Z1 Change Box Size and Position Z1

This command adjusts the position and dimensions of the specified text (or image) box. The X and Y positions indicate the top left position of the text box. All values are defined in pixels.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3	Param_4
Z1	%1x: Layer	%02x: Box 0x00 - 0xFE	%03x: X position	%03x: Y position

Param_5	Param_6
%03x: Width	%03x: Height

Example:

```
void ChangeDims(int Layer, int BoxNo, int X,
               int Y, int Width, int Height)
{
    remote_send("Z1%1x%02x%03x%03x%03x%03x", Layer,
               BoxNo, X, Y, Width, Height);
}
```

Z2 Set Text Font and Colour Z2

This command sets the font and colour of the text contained in a text box.

The font name parameter is a font filename saved on the Imagestore device. For example 'Times New Roman' should be addressed as 'times.ttf'.

The font 'point size' does not directly equate to pixels. The font's design determines this.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3	Param_4
Z2	%1x: Layer	%02x: Box 0x00 - 0xFE	%02x: Point size	%06x: Colour (page 86)

Param_5
%s: Font name

Example:

```
void ChangeFont(int Layer, int BoxNo, int Size,
               int Col, char *FontFileName)
{
    remote_send("Z2%1x%02x%02x%06x%s", Layer,
               BoxNo, Size, Col, FontFileName);
}
```



```
}

```

Z3 Render Box Z3

This command updates the specified text (or image) box on screen using all settings made since the last update.

Note: This command must be sent to force boxes to be re-rendered after adjusting their parameters.

CMD	Param_1	Param_2
Z3	%1x: Layer	%02x: Box 0x00 – 0xFE 0xFF = Update all boxes

Example:

```
void RenderBox(int Layer, int BoxNo)
{
    remote_send("Z3%1x%02x", Layer, BoxNo);
}

```

Z4 Change Image Z4

This command allows the image associated with an box to be replaced with another image held on disk. Only still image files (.OXT) are supported for this command on Imagestore devices. Intuition[+] also supports animations and MPEG clips.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
Z4	%1x: Layer	%02x: Box 0x00 - 0xFE	%s: Image filename

Example:

```
void UpdateBoxImage(int Layer, int BoxNo)
{
    remote_send("Z4%1x%02x%s", Layer, BoxNo,
               "NewLogo.oxt");
}

```

Z9 Set Text Drop Shadow Z9

This command adjusts the drop shadow of the text in the specified box.

The X and Y offsets are the distance in pixels from the text to which the drop shadow falls. Positive numbers are down and right, and negative numbers (in two's complement) are up and left. Offsets should always be set as even digits. Setting both to zero turns off the drop shadow.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3	Param_4
Z9	%1x: Layer	%02x: Box 0x00 - 0xFE	%02x: X offset	%02x: Y offset

Param_5	Param_6
%02x: Transparency 0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)	%06x: Colour (page 8686)

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                    int Xoff, int Yoff,
                    int Transparency, int Colour)
{
    remote_send ("%1x%02x%02x%02x%02x%06x", Layer,
                 BoxNo, Xoff & 0xFF, Yoff & 0xFF,
                 Transparency, Colour);
}
```

ZA Set Transparency ZA

This command sets the transparency of the text (or image) in the specified box.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
ZA	%1x: Layer	%02x: Box 0x00 - 0xFE	%02x: Transparency 0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                   int Transparency)
{
    remote_send ("%1x%02x%02x", layer, BoxNo,
                Transparency);
}
```

ZB Set Text Tracking ZB

This command changes the tracking (internal inter-character spacing) of text in the specified box.

Positive tracking numbers move the letters further apart, while negative tracking closes up the gaps between letters. The unit of tracking is 1/64 pixel. The signed value should be AND'ed with 0xFFFF to ensure it is only four hex digits long.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
ZB	%1x: Layer	%02x: Box 0x00 - 0xFE	%04x: Tracking

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                   int Tracking)
{
    remote_send ("%1x%02x%04x", layer, BoxNo,
                Tracking & 0xffff);
}
```

ZE Set Text Alignment ZE

This command sets the horizontal and vertical alignment of text within a box. The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
ZE	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Horizontal alignment 0x0 = Align left 0x1 = Align centre 0x2 = Align right

Param_4
%1x: Vertical alignment 0x0 = Align top 0x1 = Align centre 0x2 = Align bottom

ZF Set Text Wrapping ZF

This command sets the text wrapping mode of the specified box.

A 'wrap mode' of 0x0 means no automatic wrapping occurs. If the text is too wide for the box the point size is automatically reduced to fit. Existing line breaks in the text are maintained.

A 'wrap mode' of 1 allows additional line breaks to be inserted between words to use the size of the text box without excessive shrinking of the text.

This command only makes sense for static text which is not scrolling.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
ZF	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Wrap mode 0x0 = No automatic wrapping 0x1 = Allow line breaks

Za Enquire Textbox Za

This command enquires for information about a particular box within the Easytext template loaded into the specified layer.

CMD	Param_1	Param_2
Za	%1x: Layer	%02x: Box 0x00 - 0xFE

The command response returns the following information if the text box is valid:

Format	Field	Description
%c%c	CMD	Za
%03x	X position	In pixels
%03x	Y position	In pixels
%03x	Width	In pixels
%03x	Height	In pixels
%02x	Font point size	
%06x	Box colour	RGB (see page 86)
%02x	Transparency	0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)
%04x	Tracking	Font tracking
%1x	Horizontal alignment	0x0 = Left 0x1 = Centre 0x2 = Right
%1x	Vertical alignment	0x0 = Top 0x1 = Centre 0x2 = Bottom
%02x	Drop shadow X offset	In pixels
%02x	Drop shadow Y offset	In pixels
%02x	Drop shadow transparency	0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)
%06x	Drop shadow colour	RGB (see page 86)
%1x	Word wrap	0x0 = Off 0x1 = On
%s	Font name	

Zb Enquire Template Zb

This command enquires for information about the Easytext template loaded into the specified layer.

CMD	Param_1
Zb	%1x: Layer

If an Easytext template is found in that layer, the information returned is:

Format	Field	Description
%c%c	CMD	Zb
%02x	Number of objects	Number of text + image boxes
%06x	Background colour	RGB (see page 86)
%02x	Transparency	0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)
%1x	Background gradient	0x0 = Solid 0x1 = Down 0x2 = Up 0x3 = Centre 0x4 = Edge
%02x	Edge	Size of edge in pixels (if background gradient = 0x4).

Zd Text Box Update Zd

This command updates a text box within the specified Easytext template.

This is similar to the Z0 command except that the text box update only proceeds if the 'filename' matches the Easytext template that is currently loaded in the specified layer. This prevents text updates from being sent for the wrong template.

CMD	Param_1	Param_2	Param_3
Zd	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Flags (bitwise) 0x1 = Render 0x2 = Append

Param_4	Param_5	Param_6
---------	---------	---------

%02x: Filename string length	%s: Easytext filename	%s: Text
------------------------------	-----------------------	----------

Example:

```
"Zd10010dTransTest.oxaNumbers-123456789"
```

```
Zd = Command
1 = Layer (DSK 2)
00 = Text box number
1 = Flags (render)
0d = String length of filename to follow 'TransTest.oxa'
TransTest.oxa = Easytext filename
Numbers-123456789 = New text to render
```

This results in the string "Numbers-123456789" being rendered in the template 'TestTrans.oxa' if it is loaded into DSK2.

Ze Image Update Ze

This command updates an image within a loaded template on the specified layer.

This is similar to the Z4 command except that the box update only proceeds if the 'filename' matches the Easytext template that is currently loaded in the specified layer. This prevents image updates from being sent for the wrong template.

Only still image files (.OXT) are supported for this command on Imagestore devices. Intuition[+] also supports animations and MPEG clips.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
Ze	%1x: Layer	%02x: Box 0x00 - 0xFE	%02x: Filename string length

Param_4	Param_5
%s: Filename	%s: Image filename

Example:

```
Ze1010CTemplate.oxaImage.oxt:
```

```
Updates Easytext template 'Template.oxa' (if it is currently loaded on DSK2) by placing 'Image.oxt' into box 1.
```

Commands (Backgrounds)

Z6 Set Text Background to Matte Z6

This command sets a coloured background behind a box. This is automatically drawn (and resized) when the box is rendered.

The text background can have a transparency. The horizontal and vertical border parameters describe the border placed around the background. Border values of 0x00 produce a rectangle which fits exactly behind the rendered text. Values of 0x0A produce a 10 pixel border outside the text rectangle.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_4
Z6	%1x: Layer	%02x: Box 0x00 - 0xFE	%06x: Background colour (page 86)

Param_5	Param_6	Param_7
%02x: Transparency 0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)	%02x: Horizontal border 0x00 – 0xFE (steps of 2 pixels)	%02x: Vertical border 0x00 – 0xFE (steps of 2 pixels)

Example:

```
void TextBgnd_Matte(int Layer, int BoxNo, int Col,
                   int trans, int hborder,
                   int vborder)
{
    remote_send ("Z6%1x%02x%06x%02x%02x%02x",
                layer, BoxNo, Col, trans,
                hborder, vborder);
}
```

Z7 Set Text Background to Gradient Z7

This command sets a gradient background behind a box. This is automatically drawn (and resized) when the box is rendered.

The gradient starts from the source colour, and finishes at the destination colour, and has the defined direction. The border colour is drawn around the gradient-filled text box. The text background can have a transparency. The

horizontal and vertical border parameters describe the border placed around the background. Border values of 0x00 produce a rectangle which fits exactly behind the rendered text. Values of 0x0A produce a 10 pixel border outside the text rectangle.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
Z7	%1x: Layer	%02x: Box 0x00 - 0xFE	%06x: Source colour (page 86)

Param_4	Param_5	Param_6
%06x: Destination colour (page 86)	%06x: Border colour (page 86)	%02x: Transparency 0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)

Param_7	Param_8	Param_9
%02x: Border width	%03x: Direction 0x000 = Vertical 0x001 = Horizontal	%02x: Horizontal border 0x00 – 0xFE (steps of 2 pixels)

Param_10
%02x: Vertical border 0x00 – 0xFE (steps of 2 pixels)

Example:

```
void TextBgnd_Gradient(int Layer, int BoxNo,
                      int SourceCol, int DestCol,
                      int BorderCol, int Trans,
                      int BorderWidth, int Dir,
                      int hborder, int vborder)
{
    remote_send("Z7%1x%02x%06x%06x%06x%02x%02x%03x"
               "%02x%02x", Layer, BoxNo, SourceCol,
               DestCol, BorderCol, Trans,
               BorderWidth, Dir, HBorder, VBorder);
}
```

Z8 Set Text Background to Clear Z8

This command removes a coloured or gradient background.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2
Z8	%1x: Layer	%02x: Box 0x00 - 0xFE

Example:

```
void TextBgnd_Clear(int Layer, int BoxNo)
{
    remote_send ("Z8%1x%02x", layer, BoxNo);
}
```

ZD Set Template Background ZD

This command specifies a background for the entire Easytext template. (This is in addition to specifying backgrounds for individual boxes).

Template backgrounds can be a solid colour, with controllable transparency, or can have transparency that fades in and/or out. The background size will be the size of the template created in Text Builder.

The new settings take effect when the 'Render Box' command (Z3) is issued.

CMD	Param_1	Param_2	Param_3
ZD	%1x: Layer	%06x: Colour (page 86)	%02x: Transparency 0x00 – 0xFF (0x00 = Transparent) (0xFF = Opaque)

Param_4	Param_5
%1x: Mode 0x0 = Solid 0x1 = Fade down 0x2 = Fade up 0x3 = Fade both 0x4 = Fade edge	%02x: Fade edge size (width of the faded area for mode = 0x4)

Example:

```
void ChangeTextBgnd(int Layer, int Colour,  
                    int Trans, int mode,  
                    int EdgeSize)  
{  
    remote_send ("ZD%1x%06x%02x%1x%02x", layer,  
                Colour, Trans, mode, EdgeSize);  
}
```

Oxtel v16-001 PROVISIONAL

Commands and Responses (Straps)

Z5 Run Strap Z5

This command causes an Easytext strap to be stopped or started. When an Easytext strap is stopped and re-started it will run from its original start point rather than from where it was stopped.

CMD	Param_1	Param_2
Z5	%1x: Layer	%1x: Run flag 0x0 = Stop 0x1 = Start

Example:

```
void StartStrap(int Layer, bool Start)
{
    remote_send("Z5%1x%1x", Layer, Start);
}
```

Zg Pause/Restart Strap Zg

This command pauses or restarts a roll or crawl on the specified layer.

CMD	Param_1	Param_2
Zg	%1x: Layer	%1x: Flag 0x0 = Restart 0x1 = Pause

ZC Set Strap Speed ZC

This command sets the Easytext strap scrolling speed in pixels (horizontal scrolls) or lines (vertical scrolls) per field (interlaced) or frame (progressive).

Only positive even speeds (0x02, 0x04, 0x06, 0x08, 0x0A, etc.) are supported. Speeds faster than 20 (0x14) are not accepted by TextBuilder and so should be avoided. Odd numbers are accepted, but are rounded up. Zero (0x00) is invalid and should not be used.

CMD	Param_1	Param_2
ZC	%1x: Layer	%02x: Strap speed

	0x02 – 0x14 (even)
--	--------------------

Zc Enquire Crawl Loop Count Zc

This command returns information about how many times the specified crawl has looped. The loop count is reset to zero when a crawl is cut/faded up, or reloaded.

CMD	Param_1
Zc	%1x: Layer

Example:

```
void Enq_Crawl_Loop_Count(int Layer)
{
    remote_send("Zc%1x", Layer);
}
```

The response command includes the crawl count. Note that the layer number is not echoed in the response.

CMD	Param_1
Zc	%4d: Loop count

ZH Set Datasource Timer ZH

This command resets the start time that is used by STOPWATCH and COUNTDOWN internal datasources (see page 112).

CMD	Param_1	Param_2
ZH	%1x: Layer	%02x: Box (optional)

The command acts on the specified 'box' in the Easytext template that is currently loaded into the specified layer. If no 'box' parameter is present then the command acts on all boxes in the template.

Commands and Responses (Intuition[+])

The following commands only control boxes within Intuition[+] templates.

Zc Enquire Crawl Count Loop Zc

This command returns information about how many times the specified crawl has looped. The loop count is reset to zero when a crawl is cut/faded up, or reloaded.

Compared with the Imagestore Zc command, Intuition[+] requires an extra 'box' parameter because there can be more than one crawl loaded into each layer.

CMD	Param_1	Param_2
Zc	%1x: Layer	%02x: Box

Example:

```
void Enq_Crawl_Count_Loop(int Layer, int Box_no)
{
    remote_send("Zc%1x %02x", Layer, Box_no);
}
```

The response command includes the crawl count. Note that the layer number is not echoed in the response.

CMD	Param_1
Zc	%4d: Loop count

Zf Stop Animation Zf

This command stops an animation from playing in the specified box and layer.

The command behaves similarly to the S1 command. However it includes a box number parameter because Intuition[+] animations occupy a template box rather than a full layer.

CMD	Param_1	Param_2	Param_3
Zf	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Stop flag 0x0 = Complete cycle 0x1 = Stop immediately 0x2 = Complete cycle and restart (in-loop-out animations only)

Zg Pause/Restart Strap Zg

This command pauses or restarts a roll or crawl on the specified layer.

This is similar to the Imagestore 'Zg' command except that Intuition[+] requires an extra 'box' parameter because there can be more than one crawl loaded into each Intuition[+] layer, each one scrolling independently.

CMD	Param_1	Param_2	Param_3
Zg	%1x: Layer	%02x: Box 0x00 - 0xFE	%1x: Flag 0x0 = Restart 0x1 = Pause

Zh Set Crawl Loop Count Zh

This command specifies a text crawl loop count for a box in a layer.

When this is reached the crawl stops, thereby overriding any default template 'loop count' or the 'continuous' looping mode. The Intuition template file will not be updated as a result of the Zh command.

The crawl count can be sent dynamically whilst the crawl is already in progress. If the new crawl count is less than the actual count, then the text crawl will end.

CMD	Param_1	Param_2	Param_3
Zh	%1x: Layer	%02x: Box 0x00 - 0xFE	%04x: Crawl count

Param_4
%1x: Cut strap down 0x0 = Never 0x1 = On reaching crawl count

Unsolicited Tallies

Yd Strap Loop Count Yd

Note: This command is only supported by Imagestore 2/2U/3.

This unsolicited tally returns information every time one of the strap loop counts increments.

The Yd tally is enabled when passive mode (Ya1) is enabled.

The format of the Yd tally response is:

CMD	Param_1	Param_2
Yd	%08x: DSK1 loop count	%08x: DSK2 loop count

Yg Text Crawl Count Tally Yg

Note: This command is only supported by Intuition SD/HD[+].

This tally is enabled with the 'Enable Video Tallies' (Y61) or 'Enable Passive Mode' (Ya1) commands.

Once enabled, it returns information every time a text crawl cycle completes, provided that the text box has been configured to send this tally – (by setting the 'Tally' checkbox within Intuition Builder).

This is used by automation as a trigger to know when to cut down the layer, or to place new text into a box, and is used in conjunction with the Zh command. Intuition XG uses timelines to achieve the same effect, and so does not support this command.

The format of the tally is:

CMD	Param_1	Param_2	Param_3	Param_4
Yg	%1x: Layer	%02x: Box	%1x: Final crawl 0x0 = No 0x1 = Yes	%04x: Count

Note: If no crawl limit is set with the 'Zh' command then the 'count' parameter remains at zero with 'final crawl' = 0x0. This way it is possible to distinguish whether a crawl is done, or if no limit was set.

Yj Animation/Clip Cycle Tally Yj

Note: This command is only supported by Intuition SD/HD[+].

This tally is enabled with the 'Enable Video Tallies' (Y61) or 'Enable Passive Mode' (Ya1) commands.

Once enabled, it returns information every time a cell animation or clip cycle completes, provided that the box has been configured to send this tally – (by setting the 'Tally' checkbox within Intuition Builder).

This is used by automation as a trigger to know when to cut down the layer, or to place new animation/clip into a box. Intuition XG uses timelines to achieve the same effect, and so does not support this command.

The format of the tally is:

CMD	Param_1	Param_2	Param_3	Param_4
Yj	%1x: Layer	%02x: Box	%1x: Final cycle 0x0 = No 0x1 = Yes	%04x: Count

The 'final cycle' flag works differently for each loop mode, for example for layer 2, object 4:

```
// Single shot: Send single tally when the animation/clip ends
Yj20410001

// Cycle: Send tally each time the animation/clip cycles,
// giving the current crawl count.
Yj2040????

// In-Loop-Out: Send single tally when the whole animation/
// clip has completed, and does not tally on inner loops.
Yj20410001

// Linear control: Send single tally every time a move
// completes, giving the current crawl count.
Yj2041????

// Ping pong: Send tally at either end of the animation,
// giving the current crawl count.
Yj2040????
```

Datasources

Datasources simplify delivery of textual information to Easytext templates loaded into Imagestore keying layers (and Intuition templates loaded into Intuition keying layers).

Datasources come in two forms:

- Custom datasources – updated with user-defined strings by automation. These can be global or layer-specific.
- Internal datasources – provided by the system for showing information such as time and date.

The main advantage of using datasources within Easytext (and Intuition) templates is that the process of delivering textual data can become entirely separated from the process of loading and cutting up templates.

For example, it is possible to have one PC feeding a news headline summary into a datasource called 'NEWS', while another PC feeds a stock ticker info into a datasource called 'STOCKS'. If there are two templates, one referring to 'STOCKS' and the other 'NEWS', then the appropriate information will always be displayed for whichever template is loaded and the PCs providing the information can remain unaware of what is happening 'on-air'.

Also, the process of authoring and issuing new templates becomes easier since the requirement for automation to address numbered layers and text boxes is removed.

Custom Datasources

Datasources are special tags which are embedded into the strings of Easytext boxes. Custom datasources can be created, updated or deleted by automation so as to dynamically update text (static and crawling) on-air.

A custom datasource name (in this case 'DS_NEWS') can be embedded into a text box with pipe '|' separators adjacent to static text or other datasources.

```
Breaking news... |DS_NEWS|
```

Multiple datasources can be inserted anywhere into the text string, thereby allowing composite messages to be generated.

```
The time is |TIME| on |DAYOFWEEK|
```

Internal datasources such as 'TIME' are described on page 112.

The vertical pipe '|' character obviously has special meaning within Easytext text boxes, and so it cannot be included as a normal character within text

strings. Instead it can be escaped as `\7C` (see page 36), or alternatively two vertical bars `||` will generate a single vertical bar.

Datasources created by network automation sessions remain valid after the session is disconnected (rather like global variables in C). However, custom datasource values set by automation are volatile in that they are lost when the device restarts.

All 'internal' (or pre-defined) datasources use upper-case names. All datasource names, option names and values are case-sensitive and are stored as Unicode values.

Oxtel v16-001 PROVISIONAL

Commands and Responses

m0 Set Global Datasource Value m0

This command assigns a new value to an custom global datasource which can be used by templates loaded into any keying layer.

The 'flags' parameter allows text strings from multiple commands to be appended to one another. This is because automation commands are limited in size to 2048 bytes, but scrolling text strings may need to be much longer than this.

The 'name' and 'value' parameters consist of the global datasource name and its new string value, separated by a vertical pipe '|'.

Certain special characters (see page 36) may not appear within string values unless escaped.

CMD	Param_1	Param_2	Param_3
m0	%02x: Flags (bitwise) 0x01 = First packet 0x02 = Last packet	%s : Name	%s: Value

Example showing use of 'flags' parameter to append text:

```
"m001DATASOURCE_1|This is the start"
"m000DATASOURCE_1|, and this is some more"
"m002DATASOURCE_1|, but this is the end!"
```

This results in DATASOURCE_1 = "This is the start, and this is some more, but this is the end!"

Example showing use of 'flags' parameter to send text in one command:

```
"m003DATASOURCE_1|This is the lot."
```

The datasource will not be used on-air until all packets have been received. Scrolling text will not update with a new datasource until the text box starts its next pass.

m1 Delete Global Datasource m1

This command deletes a global datasource, freeing up any memory it used.

CMD	Param_1
m1	%s: Name

m2 Set Layer Datasource Value m2

This command is very similar to the m0 command except that it has a 'layer' parameter. The new datasource value is assigned to the specified layer only.

When a template loaded into this layer makes reference to the layer-specific datasource name, its value will be used in preference to a global datasource that exists with the same name. However, if the layer-specific datasource name does not exist, then the global datasource will be used.

CMD	Param_1	Param_2	Param_3	Param_4
m2	%02x: Flags (bitwise) 0x01 = First packet 0x02 = Last packet	%1x: Layer	%s : Name	%s: Value

m3 Delete Layer Datasource m3

This command deletes a layer datasource, freeing up any memory it used.

CMD	Param_1	Param_2
m3	%1x: Layer	%s: Name

m4 Enquire Global Datasource List m4

This command enquires for a list of all the global datasource names that are currently defined.

Once the list of names has been retrieved by automation, it can query individual global datasource values using the 'm5' command.

CMD
m4

The response command includes a pipe-separated list of all available global datasource names:

CMD	Param_1 ... Param_n
m4	%s: Name 1 %s: Name 2 %s: Name 3 ...

m5 Enquire Global Datasource Value m5

This command enquires the value of the specified global datasource name.

CMD	Param 1
m5	%s: Name

The response command includes the global datasource value:

CMD	Param 1	Param 2
m5	%s : Name	%s: Value

If there is no value defined for this global datasource name, then the 'value' parameter will be an empty string.

m6 Enquire Layer Datasource List m6

This command enquires for a list of all the layer datasource names that are currently defined for the specified layer.

Once the list of names has been retrieved by automation, it can query individual layer datasource values using the 'm7' command.

CMD	Param 1
m6	%1x: Layer

The response command includes a pipe-separated list of all available layer datasource names:

CMD	Param 1	Param_2 ... Param_n
m6	%1x: Layer	%s: Name 1 %s: Name 2 %s: Name 3 ...

m7 Enquire Layer Datasource Value m7

This command enquires the value of the specified layer datasource name for the specified layer.

CMD	Param 1	Param 2
m7	%1x: Layer	%s: Name

The response command includes the layer datasource value:

CMD	Param 1	Param 2	Param 3
m7	%1x: Layer	%s : Name	%s: Value

If there is no value defined for this global datasource name, then the 'value' parameter will be an empty string.

m80 Delete All Datasources m80

This command deletes all datasources (global and layer-specific), freeing up any memory that was used for them.

CMD
m80

Internal Datasources

Internal datasource are provided by the system and cannot be overridden or deleted by automation. They provide a range of internal information that can be included within Easytext boxes and viewed on-air.

Internal datasources use the following format:

```
| INTERNAL_DATASOURCE_NAME[:option=value] |
```

Options are specified after a colon (if required). The option name is followed by a quoted value string. The interpretation of options and their values is entirely dependent on the internal datasource. For example:

```
| TIME | might produce "11:55"
| TIME:FORMAT "H:M:S" | might produce "11:55:01"
| TIME:FORMAT "H:M":OFFSET "-1" | might produce "10:55"
(multiple options, each of which could be omitted)
```

TIME

This datasource presents the current time of day from the device's system clock.

The FORMAT option determines the time format. All characters in the format are copied into the output, except the following special characters which insert the following time information:

- i Hours for 12 hour clock, leading zeroes.
- h Hours for 12 hour clock, no leading zeroes.
- H Hours for 24 hour clock, leading zeroes.
- M Minutes, 2 digits, leading zeroes.
- S Seconds, 2 digits, leading zeroes. (Use of this field will only work in very small, simple templates).
- A 'AM' or 'PM' accordingly.
- a 'am' or 'pm' accordingly.

The default FORMAT is "i:Ma"

Example:

```
| TIME:FORMAT="h:M" | // "09:15"
| TIME:FORMAT="i:M.S a" | // "9:15.42 am" ('S' not recommended)
```

The OFFSET option allows time-zone adjustments to be made without adjusting the system clock. The offset is specified in seconds.

```
| TIME:OFFSET= -3600 | // "08:15"
| TIME:OFFSET= 3600 | // "10:15"
```


The **FRAMES** parameter (similar to **OFFSET**, but in frames), allows the time to be synchronised to the nearest second. The allowed value ranges from +/- frame rate.

DATE

This datasource presents the current system date.

The **FORMAT** option determines the date format. All characters in the format are copied into the output, except the following special characters which insert the following date information:

- D Day of the month, leading zeroes
- d Day of the month, no leading zeroes
- M Month number, leading zeroes
- m Month number, no leading zeroes
- Y Year number, 4 digits ('2011')
- y Year number, last 2 digits ('11')
- W Day name ('Monday')
- n Day number with suffix ('1st', '2nd', '3rd', '4th', etc.)
- O Month name ('November')
- o Month short name ('Nov')

The default **FORMAT** is "D-o-Y"

Examples:

```
| DATE:FORMAT="D/M/Y" | // "31/04/2011"
| DATE:FORMAT="W n O" | // "Saturday 30th April"
```

The **OFFSET** parameter (also in seconds, like the **TIME** command), allows day rollover to be synchronised for other time zones.

The **FRAMES** parameter (similar to **OFFSET**, but in frames), allows day rollover to be synchronised to the nearest second.

STOPWATCH

This datasource presents a stopwatch which can be set to run from a user-defined start time (when the layer cuts/fades up) to a user-defined end time. If the end time is higher than the start time, the stopwatch counts up. If the end time is lower than the start time, the stopwatch counts down.

The **FORMAT** option determines the stopwatch format. All characters in the format are copied into the output, except the following special characters which insert the following stopwatch information:

- H Hours (modulus 24), 2 digits, leading zeroes.
- h Hours (modulus 24), no leading zeros.
- M Minutes (modulus 60), 2 digits, leading zeros.
- m Minutes (modulus 60), no leading zeros.

- S Seconds (modulus 60), 2 digits, leading zeros.
- s Seconds (modulus 60), no leading zeros.
- l Hours (total), 2 digits, leading zeros.
- i Hours (total), no leading zeros.
- N Minutes (total), 2 digits, leading zeros.
- n Minutes (total), no leading zeros.
- T Seconds (total), 2 digits, leading zeros.
- t Seconds (total), no leading zeros.

The default format is “h:M:S”.

Example counting up in total seconds:

```
| STOPWATCH:FORMAT="t" |
```

The START option determines the start time of the stopwatch. The default start value is “00:00:00”. The start time can be reset using the 'ZH' command (page 101).

The STOP option determines the end time of the stopwatch. The default stop value is “00:00:00”.

Example counting down from 60 to 30 seconds.

```
| STOPWATCH:FORMAT="t":START="00:01:00":STOP="00:00:30" |
```

COUNTDOWN

This datasource presents a countdown timer which can be run from a user-defined start time (when the layer cuts/fades up) to a user-defined end time.

The FORMAT option determines the countdown format, which matches the formatting used by the STOPWATCH datasource.

The default format is “h:M:S”.

Example:

```
| COUNTDOWN:FORMAT="M:S" | // Count down in min/sec
```

The START option determines the start time of the countdown. The default start value is “00:00:00”. The start time can be reset using the 'ZH' command (page 101).

Example counting down in seconds from 60 to 0.

```
| STOPWATCH:FORMAT="t":START="00:01:00" |
```

TEMPERATURE_CELSIUS

This datasource presents the temperature read by the temperature probe option in degrees Celsius. The number is shown as an integer.

Note: This datasource is only supported by products with the temperature probe option; Imagestore 750, LGK-3901 and ISM-3901.

TEMPERATURE_CELSIUS_TENTHS

This datasource presents the temperature read by the temperature probe option in degrees Celsius. The number is shown as an integer with one decimal place.

Note: This datasource is only supported by products with the temperature probe option; Imagestore 750, LGK-3901 and ISM-3901.

TEMPERATURE_FAHRENHEIT

This datasource presents the temperature read by the temperature probe option in degrees Fahrenheit. The number is shown as an integer.

Note: This datasource is only supported by products with the temperature probe option; Imagestore 750, LGK-3901 and ISM-3901.

TEMPERATURE_FAHRENHEIT_TENTHS

This datasource presents the temperature read by the temperature probe option in degrees Fahrenheit. The number is shown as an integer with one decimal place.

Note: This datasource is only supported by products with the temperature probe option; Imagestore 750, LGK-3901 and ISM-3901.

FADER0 through FADER7

This datasource presents the value of an analogue input fader, modified according to the options.

Note: This datasource is only supported by products with faders; Imagestore 2/2U/3, Imagestore 300[+] and Imagestore HD-TV.

Analogue faders return a nominal value between 0 and 1. The voltages representing 0 and 1 must be discovered empirically.

The SCALE option provides a scaling factor for the value. It defaults to 1.0.

The OFFSET option provides an offset value for the number. It defaults to 0.0.

The value shown by the datasource is $(FADERVALUE * SCALE) + OFFSET$

The DECIMALS option determines how many decimal places are displayed with the number. This value defaults to 2.

HOSTNAME

This datasource returns the hostname of the device.

IP_ADDR

This datasource returns the Ethernet IP address of the unit.

SERIAL

This datasource returns the serial number of the unit.

SYSNAME

This datasource returns a brief description of the software and version.

Oxtel v16-001 PROVISIONAL

Emergency Alert System (EAS)

The Emergency Alert System (EAS) is a national warning system in the United States that notifies the general public of pending or imminent situations such as weather emergencies. It can also be used by the President of the United States to deliver a message. All television stations in the USA are required to transmit EAS messages. Cable head-ends are also required to transmit EAS messages on all programmed channels. For more information on EAS, see the FCC website or the websites of the EAS receiver manufacturers.

A televised EAS alert consists of two components:

- An on-screen crawl displaying the nature of the emergency and the regions affected in the broadcast area
- An audio message up to two minutes in length

The EAS option provides a mechanism for the Oxtel device to receive information from an external EAS receiver and then generate the required video crawl and audio voice-over. The device provides flexible branding capabilities for the video crawl, using an Easytext template, such as adjustable font type, size, and colour, additional images or text, etc. It provides full GPI macro programmability and enforces keyer control so that the crawl is shown for the duration of the alert.

Please refer to individual product manuals for additional information.

Commands

n1 EAS On n1

This command brings an EAS crawl to air.

The crawl uses text specified by the n3 command, and the priority specified by the n4 command. The most downstream keyer is always used to ensure that the message is always remains visible.

CMD
n1

n2 EAS Off n2

This command switches off an EAS crawl that was previously on air.

In addition, the command restores any graphics that were loaded into the most downstream keying layer before the EAS crawl was brought to air.

CMD
n2

n3 EAS Set Text n3

This command sets the text that will be displayed for the next EAS crawl.

The text parameter is saved for the current priority level. The next time that the EAS On (n1) command is received, this text will be displayed.

CMD	Param_1
n3	%s: Text

Example:

```
n3This is an EAS crawl:
```

Note: After using the 'n3' command, it is recommended to wait for 3 seconds before issuing the next EAS On (n1) command.

n4 EAS Set Priority n4

This command sets the priority level of the next EAS crawl.

CMD	Param_1
n4	%02d: Priority

Examples:

```
n401: // Sets to the highest priority (1)
n402: // Sets to the medium priority (2)
n403: // Sets to the lowest priority (3)
```

Note: After using the 'n4' command, it is recommended to wait for 3 seconds before issuing the next EAS On (n1) command.

SDI Video

Inputs

The following table lists the physical SDI inputs that are present on different Oxtel devices, and the “input number” used by automation to reference them.

Physical SDI input	A	F1	K1	B	F2	K2	F3	K3
Input number	0	1	2	3	4	5	6	7
<i>Imagestore 2/2U/3</i>	x	x	x	x				
<i>Imagestore 300</i>	x	x	x	x				
<i>Imagestore 300+</i>	x	x	x	x	x	x		
<i>Imagestore HD-TV</i>	x	x	x	x	x	x		
Imagestore 750	x	x	x	x	x	x	x	x
DSK-3901	x	x	x		x	x		
LGK-3901	x	x	x		x	x		
ISM-3901	x	x	x		x			

Please note the following points that relate to Imagestore inputs:

- The naming of physical SDI differs slightly depending on the product. For example, “A” is known as “Background” on some older Imagestore products, and “PGM-In” on DSK-3901, LGK-3901 and ISM-3901. Please check individual user manuals for details.
- The “B” input may require an A/B mixer option to be enabled.
- Imagestore HD-TV requires an optional card for F2/K2 inputs.
- DSK-3901, LGK-3901 and ISM-3901 have a “Fill-2/B” SDI input which is used as a “B” input when the A/B mixer option is present. Otherwise it is used as a Fill-2 input. When the A/B mixer option is present, the Key-2 SDI input can not really be used.

Commands and Responses

1 Set Input Masks 1

This command (lower-case L) sets an input mask for the specified SDI input.

All of the mask parameters are measured in pixels so their range is dependent on the video standard.

Setting the input mask directly affects the input SDI source, so anything using the input source will be affected.

If an input is not currently using its input SDI source (because it has been set to a colour field or test pattern), the input will not be masked. Only the input SDI source is masked.

CMD	Param_1	Param_2	Param_3	Param_4
1	%x: Input (page 119)	%x: Left 0x0 - width	%x: Right 0x0 - width	%x: Top 0x0 - height

Param_5
%x: Bottom 0x0 - height

Example:

```
void SetInputMask(int Input, int Left, int Right,
                  int Top, int Bottom)
{
    remote_send("l%x %x %x %x %x", Input, Left,
                Right, Top, Bottom);
}
```

1 Enquire Input Masks 1

This command returns information about the input SDI masks:

CMD	Param_1
1	%x: Input

Example:

```
void EnquireInputMask(int Input)
{
```



```

    remote_send("l%x", Input);
}

```

The information returned will be:

CMD	Param_1	Param_2	Param_3	Param_4
1	%x: Input (page 119)	%x: Left 0x0 - width	%x: Right 0x0 - width	%x: Top 0x0 - height

	Param_5
	%x: Bottom 0x0 - height

XM Enquire Input Mode XM

This command enquires the mode that is associated with the specified input.

CMD	Param_1
XM	%1d: Input (see page 119)

The input mode is returned in the response command:

CMD	Param_1	Param_2
XM	%1d: Input (see page 119)	%1d: Mode <u>Imagestore 750, DSK-3901, LGK-3901, ISM-3901</u> 0x0 = Pass SDI 0x1 = Use colour field 1 0x2 = Use colour field 2 0x3 = Use colour field 3 0x4 = Use test pattern 1 0x5 = Use test pattern 2 <u>Imagestore 300[+], Imagestore HD-TV</u> 0x0 = Pass SDI

	0x1 = Use dedicated colour field
--	----------------------------------

Xq Set Post-A/B Mix Video Delay Xq

This command sets the post-A/B mix (or pre-keyer) video delay for the PGM or PVW bus. This is used on Imagestore 750 to compensate for DVE processing, or for bypassing an external graphics device (such as Intuition XG).

CMD	Param_1	Param_2
Xq	%1x: Bus 0x0 = PGM 0x1 = PVW	%1x: Delay in frames 0x0 - 0x8

Select 1 frame of delay when using a DVE in interlaced video standards and 2 frames of delay when using DVE in progressive video standards. When bypassing an external graphics device (such as Intuition XG) add further frames to compensate for the delay through this device.

Xq Enquire Post-A/B Mix Video Delay Xq

This command enquires the post-A/B mix (or pre-keyer) video delay for the PGM or PVW bus.

CMD	Param_1
Xq	%1x: Bus 0x0 = PGM 0x1 = PVW

The response includes the frame delay setting:

CMD	Param_1	Param_2
Xq	%1x: Bus 0x0 = PGM 0x1 = PVW	%1x: Delay in frames 0x0 - 0x8

Xs Enquire Video Standard Xs

This command enquires which video standard is currently set on the device.

CMD
Xs

The command response is formatted as follows:

CMD	Param_1
Xs	%1x: Video standard 0x0 = PAL (625) 0x1 = NTSC (525) 0x2 = 1080i 59.94Hz 0x3 = 1080i 50Hz 0x4 = 720p 59.94Hz 0x5 = 720p 50Hz 0x6 = 1080p 59.94Hz 0x7 = 1080p 50Hz

Xt Set Output Frame Delay Xt

This command sets the frame delay for the specified output. This is typically used to compensate for audio processing delays such as when decoding or encoding Dolby data, or performing a stereo-to-5.1 UpMix.

CMD	Param_1	Param_2
Xt	%1x: Output 0x0 = Program (PGM) 0x1 = Preview (PVW) 0x2 = Clean feed (CLN)	%02x: Delay in frames 0x00 - 0x10

Please note that there is no delay capability for the Monitor output in Imagestore 750.

Xt Enquire Output Frame Delay Xt

This command enquires the current output frame delay for the specified output.

CMD	Param_1
Xt	%1x: Output 0x0 = Program (PGM) 0x1 = Preview (PVW) 0x2 = Clean feed (CLN)

The response command includes the frame delay:

CMD	Param_1	Param_2
Xt	%1x: Output 0x0 = Program (PGM) 0x1 = Preview (PVW) 0x2 = Clean feed (CLN)	%02x: Delay in frames 0x00 - 0x10

A/B Mixer

Commands and Responses

These commands control the A/B mixer functionality of the Imagestore.

When mixing between A and B inputs (or sources), the mix may pass through a colour field (usually black) or just cross between the sources. These are termed V-fade and X-fade respectively.

For Imagestore 2/2U/3, Imagestore 300[+], DSK-3901, LGK-3901 and ISM-3901 products, SDI inputs A and B provide fixed inputs for the AB mixer.

Imagestore HD-TV and Imagestore 750 products have an input router and are able to route any of the available SDI inputs into the AB mixer.

The commands are not supported if the A/B mixer option is not purchased, such as for LOGO products.

U0 Cut to A U0

This command cuts the A/B mixer immediately to the A input.

CMD
U0

Example:

```
void CutToA()
{
    remote_send("U0");
}
```

U1 Cut to B U1

This command cuts the A/B mixer immediately to the B input.

CMD
U1

U2 Fade to A U2

This command fades the A/B mixer to the A input over a specified number of fields (interlaced) or frames (progressive).

CMD	Param_1
U2	%03x: Duration (fields/frames) 0x000 – 0x3E7

Example:

```
void FadeToA(int fields)
{
    remote_send("U2%03x", field);
}
```

Note: The duration parameter is optional for Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x since these products define A/B mix channel transition rates (UI) for automation, slow, medium and fast channel transition modes (UH).

U3 Fade to B U3

This command fades the A/B mixer to the B input over a specified number of fields (interlaced) or frames (progressive).

CMD	Param_1
U3	%03x: Duration (fields/frames) 0x000 - 0x3E7

Note: The duration parameter is optional for Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x since these products define A/B mix channel transition rates (UI) for automation, slow, medium and fast channel transition modes (UH).

U4 Cut AB U4

This command cuts to the opposite input from the one currently visible. For example, from A to B, or B to A.

CMD
U4

U5 Fade AB U5

This command fades the A/B mixer to the opposite input to the one currently visible over specified number of fields (interlaced) or frames (progressive).

The transition type used is defined by the U6 command.

CMD	Param_1
U5	%03x: Duration (fields/frames) 0x000 - 0x3E7

Note: The duration parameter is optional for Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x since these products define A/B mix channel transition rates (UI) for automation, slow, medium and fast channel transition modes (UH).

U6 Set Transition Type U6

This command selects the A/B mix transition type to be used in commands U2, U3, U5, U9, UH and UI.

CMD	Param_1
U6	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x06 = not used 0x07 = not used 0x10 = not used 0x11 = not used 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2

Note: Fade-cut, cut-fade, U-fade, V-fade preset 1 and V-fade preset 2 are only supported by Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 from software version 4.x.

Example:

```
void SelectCrossFade();
{
    remote_send("U603");
}
```

Note: To set the colour field for the mid-point of a V-fade, use the UD command.

U6 Enquire Transition Type U6

This command enquires the current A/B mix transition type to be used in commands U2, U3, U5, U9, UH and UI.

CMD

U6

The response command has the format:

CMD	Param_1
U6	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x06 = Wipe left-to-right 0x07 = Wipe top-to-bottom 0x10 = Wipe right-to-left 0x11 = Wipe bottom-to-top 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2

Note: Wipes are only supported by Imagestore 2/2U/3 and Imagestore 300[+].

U8 Asymmetric V-Fade AB U8

This command instructs the A/B mixer to V-fade from the current input to the other through the V-fade colour (see UD) over specified number of fields (interlaced) or frames (progressive).

Separate duration parameters define the 'up' and 'down' phases of the transition allow asymmetric V-fades to be defined.

CMD	Param_1	Param_2
U8	%03x: 'Down' duration (fields/frames) 0x00 - 0x3E7	%03x: 'Up' duration (fields/frames) 0x00 - 0x3E7

UA Asymmetric Transition UA

This command performs an A/B mixer transition (according to current Transition Mode, set with U6 command) to the destination specified.

For cuts, the duration is ignored.

CMD	Param_1	Param_1	Param_2
UA	%1x: Destination 0x0 = A input 0x1 = B input	%03x: 'Down' duration (fields/frames) 0x000 - 0x3E7	%03x: 'Up' duration (fields/frames) 0x000 - 0x3E7

Note: The duration parameters are optional for Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x since these products define A/B mix channel transition rates (UI) for automation, slow, medium and fast channel transition modes (UH).

UB Transition Complete UB

On completion of an A-to-B transition, Presmaster waits for a tally from the router confirming that the source feeding the A-input has switched to match the B source. It then issues this command to instruct the Imagestore to cut the A/B mixer back to 'A'.

The command is similar to 'Cut to A', but includes compensating delays for Squeezy, etc.

CMD
UB

UC Fade to Specified Position UC

This command transitions the A/B mixer to the specified position over the given number of fields (interlaced) or frames (progressive).

The position can be from 0x000 (A) to 0x200 (B) and all values in between.

This command can be used by a panel for controlling U-Fades. First select the V-fade transition type (U602), then transition to the mid-point of the V-fade (UC100032) followed by a transition to B some time later (UC200032) – perhaps when the take key is released.

CMD	Param_1	Param_1
UC	%03x: Destination 0x000 = A input 0x200 = B input	%03x: Duration (fields/frames)

UD Set V-Fade Colour UD

This command defines the colour field that A/B mixer V-fades will pass through at the 50% point of the fade. The default colour is black.

CMD	Param_2	Param_3	Param_4
UD	%02x: Red 0x00 – 0x64 (0x00 – 0xFF)	%02x: Green 0x00 – 0x64 (0x00 – 0xFF)	%02x: Blue 0x00 – 0x64 (0x00 – 0xFF)

Note: For Imagestore 2/2U/3 RGB values are between 0x00 and 0xFF. For all other products RGB values are between 0x00 and 0x64.

UD Enquire V-Fade Colour UD

This command enquires the colour field that A/B mixer V-fades will pass through at the 50% point of the fade.

CMD
UD

The return command has the following format:

CMD	Param_2	Param_3	Param_4
UD	%02x: Red 0x00 – 0x64 (0x00 – 0xFF)	%02x: Green 0x00 – 0x64 (0x00 – 0xFF)	%02x: Blue 0x00 – 0x64 (0x00 – 0xFF)

Note: For Imagestore 2/2U/3 RGB values are between 0x00 and 0xFF. For all other products RGB values are between 0x00 and 0x64.

UE Select Mixer Input UE

This command is used to route SDI video sources into the A and B inputs of the A/B mixer.

The source parameter depends on how many SDI video source inputs are connected. Please refer to individual product manuals for details. There are a maximum of 6 SDI inputs for Imagestore HD-TV, and 8 for Imagestore 750.

CMD	Param_1	Param_2
UE	%1x: Mixer input 0x0 = A input 0x1 = B input	%1x: SDI source 0x0 = A 0x1 = B 0x2 = Fill-1 0x3 = Key-1 0x4 = Fill-2 0x5 = Key-2 0x6 = Fill-3/C 0x7 = Key-3/D

Example:

```
void SelectMixer(int Mixer, int Source)
{
    remote_send("UE%1x%1x", Mixer, Source);
}
```

UE Enquire Mixer Input UE

This command enquires for the current source for the specified mixer input.

CMD	Param_1
UE	%1x: Mixer input 0x0 = A input 0x1 = B input

Example 2:

```
void GetMixer(int Mixer)
{
    remote_send("UE%1x", Mixer);
}
```

The information returned is:

CMD	Param_1	Param_2
UE	%1x: Mixer input 0x0 = A input 0x1 = B input	%1x: SDI source 0x0 = A 0x1 = B 0x2 = Fill-1 0x3 = Key-1 0x4 = Fill-2 0x5 = Key-2 0x6 = Fill-3/C 0x7 = Key-3/D

UF Select Channel Input UF

This command sets which of the SDI inputs (PGM In or Fill-2/B) will be shown on the PVW output.

Note: This command is only valid for LGK-3901 and DSK-3901 and is for use when there is no A/B mixer option. It is included in the A/B mixer command section because it is related to the UE command.

This value applies when the A/B mixer option is not present.

CMD	Param_1	Param_2	Param_3
UF	%1x: A/B mixer 0x0 (always)	%1x: Output 0x1 = PVW	%1x: Source 0x0 = PGM In 0x4 = Fill-2/B

UF Enquire Channel Input UF

This command enquires which of the SDI inputs (PGM or Fill-2/B) is currently routed to the PVW output.

CMD	Param_1	Param_2
UF	%1x: A/B mixer 0x0 (always)	%1x: Output 0x0 = PGM 0x1 = PVW

The information returned for LGK-3901 or DSK-3901 is:

Format	Field	Description
%c	CMD	'UF'
%1x	A/B mixer	0x0 (always)
%1x	Output	0x0 = PGM 0x1 = PVW
%1x	Source	0x0 = PGM In 0x4 = Fill-2/B (only applicable for PVW output)

UH Set A/B Mix Profile UH

This command sets an A/B mix profile for a transition type (also see U6).

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

The profile's values define 'down', 'hold' and 'up' values as percentages of the overall channel A/B mix transition duration (UI). These values are attributed to each part of the transition as follows:

- Down Transition from A or B to the 50% mix point
- Hold Hold at the 50% mix point – (used for U-fades)
- Up Transition from the 50% mix point to B or A

CMD	Param_1	Param_2	Param_3
UH	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2	%02x: Down (%) 0x00 - 0xC8	%02x: Hold (%) 0x00 - 0xC8

Param_4

%02x: Up (%)
0x00 - 0xC8

The default profile values are:

Transition Type	Down %	Hold %	Up %
V-fade	50	0	50
Fade-cut	100	0	0
X-fade	50	0	50
Cut-fade	0	0	100
Cut	0	0	0
U-fade	33	34	33
V-fade preset 1	50	0	50
V-fade preset 2	50	0	50

The sum of percentages for fades will normally total 100%; however it is possible to modify the down, hold and up percentages between 0% and 200%. For example, if you want fade-cuts to last half the duration of other fades, set its A/B mix profile to (50%, 0%, 0%)

The current channel transition duration depends on the rates set (UI) and the current channel transition mode (UJ), which can be “automation”, “slow”, “medium” or “fast”

UH Enquire A/B Mix Profile UH

This command enquires the A/B mix profile for a transition type (also see U6).

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

CMD	Param_1
UH	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2

The response provides the down, hold and up values as percentages of the overall channel A/B mix transition duration (UI).

CMD	Param_1	Param_2	Param_3
UH	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2	%02x: Down (%) 0x00 - 0xC8	%02x: Hold (%) 0x00 - 0xC8

Param_4
%02x: Up (%) 0x00 - 0xC8

UI Set Mixer Transition Duration UI

This command sets the A/B mix transition duration for one of the available channel transition modes (UJ).

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

CMD	Param_1	Param_2
UI	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast	%03x: Duration (fields/frames) 0x00 - 0x3E7

The transition mode (automation, slow, medium, fast) can change dynamically (UJ), so the A/B mix transition duration (UI) varies accordingly. This A/B mix transition duration is then multiplied with the profile (UH) for the current A/B mix transition type (U6). It is used when the U2 and U3 commands are sent without a rate parameter.

In summary, the A/B mix transition depends on all of the following:

- Channel transition mode (UJ)
- A/B mix transition duration (UI)
- A/B mix transition type (U6)

- A/B mix profile (UH)

UI Enquire Mixer Transition Duration UI

This command enquires the A/B mix transition duration for one of the available channel transition modes (UJ).

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

CMD	Param_1
UI	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast

The transition duration is returned in the following format:

CMD	Param_1	Param_2
UI	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast	%03x: Duration (fields/frames) 0x00 - 0x3E7

UJ Set Channel Transition Mode UJ

This command sets the channel transition mode to one of:

- Automation Should always be used by automation
- Slow Can be used by panels
- Medium Can be used by panels
- Fast Can be used by panels

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

Keyer fades (#B), audio voiceover fades (jq6) and A/B mixing (UI) all depend on the current channel transition mode. Each keyer and voiceovers has independent transition rates for each channel transition mode.

CMD	Param_1
UJ	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast

UJ Enquire Channel Transition Mode UJ

This command enquires what the current channel transition mode is set to.

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

CMD
UJ

The response has the following format:

CMD	Param_1
UJ	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast

Ua Enquire Mix Mode Ua

Note: This command is largely deprecated. The same parameters can be found using U6, UI and UD enquires, and Y6 tallies.

This command returns the status of various mix parameters, such as transition type, mix rate, wipe softness, current mix angle and v-fade colour.

CMD
Ua

The response command returned is:

CMD	Param_1	Param_2
Ua	%02x: Transition type As for U6, except for... <u>Imagestore 2/2U/3</u> 0x00 = Cut 0x01 = X-fade 0x02 = V-fade 0x03 = Wipe left-to-right 0x04 = Wipe right-to-left 0x05 = Wipe top-to-bottom 0x06 = Wipe bottom-to-top	%03x: A/B mix rate (fields/frames) 0x00 - 0x3E7

Param_3	Param_4	Param_5
%03x: Wipe softness 0x000 - 0x080	%03x: A/B mix angle 0x000 = Mixer at A 0x200 = Mixer at B	%06x: V-fade colour RGB value

For Imagestore 2/2U/3 the RGB range is 0x00 to 0xFF.

For other Imagestores the RGB range is 0x00 to 0x64.

Ub Set A/B Cut Mode Ub

This command determines whether A/B mixes (cuts or fades) are started on field 1, field 2, or any field.

CMD	Param_1
Ub	%02x: A/B cut mode 0x00 = Any field 0x01 = First field (default) 0x02 = Second field

Example:

```
void SetABCutMode(int Mode);
{
    remote_send("Ub%02x", Mode);
}
```

Ub Enquire A/B Cut Mode Ub

This command enquires the A/B cut mode that is currently set:

CMD
Ub

Example:

```
void EnquireABCutMode(void)
{
    remote_send("Ub");
}
```

The information returned is:

CMD	Param_1
Ub	%02x: A/B cut mode 0x00 = Any field 0x01 = First field (default) 0x02 = Second field

Unsolicited Tallies

Y6 Video Tally Y6

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about positions of A/B mixer, DSK1/DSK2 faders, DSK1/DSK2 FTB, DVE sequence or router mode as they change.

On registration, a tally is sent so that automation can record the initial video states.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
Y6	%1x: A/B mix at 0x0 = A 0x1 = B 0x2 = In between	%1x: DSK1 position 0x0 = Off 0x1 = On 0x2 = In between	%1x: DSK2 position 0x0 = Off 0x1 = On 0x2 = In between

Param_4	Param_5
%1x: DSK1 FTB position 0x0 = On (black) 0x1 = Off 0x2 = In between	%1x: DSK2 FTB position 0x0 = On (black) 0x1 = Off 0x2 = In between

Param_6	Param_7
%02x: DVE sequence See below for parameter details	%02x: DVE router mode See below for parameter details

Note: The Y6 command returns information on two keying layers only. Imagestore 750 and LGK-3901 have additional keying layers (4 and 5 respectively). Keyer and FTB states for these are broadcast using 3 and Yf tallies.

Note: Generally speaking we recommend automation use 3 and Yf tallies for keyer and FTB states (when available), and Y6 for A/B mixer position state only.

For Imagestore 2/2U, the DVE sequence is the current squeeze preset loaded. If the squeeze is running (mid-sequence) the preset number is 0xFF.

For Imagestore 3, Imagestore 300[+], Imagestore HD-TV and Imagestore 750, the DVE sequence is the current loaded sequence.

For Imagestore 2/2U, the standard DVE router modes are:

SQ_NONE	0x00
SQ_PIC_IN_PIC	0x01
SQ_REVEAL	0x02
SQ_STORED	0x03
SQ_REVEAL_MG	0x04
SQ_REVEAL_FG	0x05
SQ_STORED_MG	0x06
SQ_STORED_FG	0x07
SQ_B_OVER_A	0x08
SQ_BYPASS	0x09
SQ_MIX	0x0A

For Imagestore 3 and Imagestore 300[+] devices (fitted with a dual channel 2D/3D DVE), the available DVE router modes are:

DVE_ROUTER_NONE	0x00
DVE_ROUTER_MIDGROUND_REVEAL	0x04
DVE_ROUTER_BYPASS	0x09
DVE_ROUTER_MIDGROUND_MIX_REVEAL	0x0A
DVE_ROUTER_IS_MIGROUND_AB	0x0B
DVE_ROUTER_IS_MIGROUND_BA	0x0C
DVE_ROUTER_IS_MID_SQZ_A_MID	0x0D
DVE_ROUTER_IS_MID_SQZ_MID_A	0x0E
DVE_ROUTER_BEHIND	0x0F

For Imagestore HD-TV, the available DVE router modes are:

DVE_ROUTER_NONE	0x00
DVE_ROUTER_A_B_MIXER	0x01
DVE_ROUTER_A_AND_B_OVER_DSK1	0x04
DVE_ROUTER_BYPASS	0x09

For Imagestore 750, the available DVE router modes (PGM channel) are:

DVE_ROUTER_ENABLED	0x04
--------------------	------

DVE_ROUTER_DISABLED	0x09
---------------------	------

For LGK-3901 and DSK-3901, parameters 6 and 7 are always returned as zero since DVE is not supported.

Note: Please refer to individual user manuals for further information on DVE modes.

Y6 Enable Video Tallies Y6

This command enables or disables the Y6, Y9, YA, YG, 3, Yf and UE video tallies for the serial or network port on which the command was received.

CMD	Param_1
Y6	%1x: Tally enable 0x0 = No 0x1 = Yes

Example:

```
void EnableVideoTallies(bool Enable)
{
    remote_send("Y6%1x", Enable);
}
```

For Imagestore 750, LGK-3901 and DSK-3901 this command has an additional optional parameter that enables or disables returning of tallies for commands that are forwarded to an Intuition slave device. When Intuition tally forward is enabled, tallies are returned by the Intuition to the primary device.

CMD	Param_1	Param_2
Y6	%1x: Tally enable 0x0 = No 0x1 = Yes	%1x: Slave tally forward - (optional) 0x0 = No 0x1 = Yes

Note: Video tallies are also be enabled using Set Passive Mode (Ya) command (page 303).

Y6 Enquire Video Tallies Y6

This command enquires the status of the Y6 tallies:

CMD
Y6

The information returned for Imagestore 2/2U/3, Imagestore 300[+] and Imagestore HD-TV is:

CMD	Param_1
Y6	%1x: Tally enable 0x0 = No 0x1 = Yes

The information returned for Imagestore 750, LGK-3901 and DSK-3901 is:

CMD	Param_1	Param_2
Y6	%1x: Tally enable 0x0 = No 0x1 = Yes	%1x: Slave tally forward 0x0 = No 0x1 = Yes

U6 Transition Type Tally U6

This tally is enabled with the 'Enable Transition Type Tallies' (YU61) command.

Once enabled, it returns information about the transition type of the A/B mixer as this is changed using the U6 command.

On registration, a tally is sent so that automation can record the initial A/B mixer transition type state.

The format of the tally is:

CMD	Param_1
U6	%02x: Transition type 0x01 = V-fade 0x02 = Fade-cut 0x03 = X-fade 0x04 = Cut-fade 0x05 = Cut 0x1A = U-fade 0x1B = V-fade preset 1 0x1C = V-fade preset 2

YU6 Enable Transition Type Tallies YU6

This command enables or disables the 'transition type' (U6) tallies.

CMD	Param_1
YU6	%1x: Tally enable 0x0 = No 0x1 = Yes

YU6 Enquire Transition Type Tallies YU6

This command enquires the status of the 'transition type' (U6) tallies:

CMD
YU6

The format of the response command is:

CMD	Param_1
YU6	%1x: Tally enable 0x0 = No 0x1 = Yes

UE Mixer Input Tally UE

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about sources for the A/B mixer as they are changed using the UE command.

The format of the tally is:

CMD	Param_1	Param_2
UE	%1x: Mixer input 0x0 = A input 0x1 = B input	%1x: SDI source 0x0 = A 0x1 = B 0x2 = Fill-1 0x3 = Key-1 0x4 = Fill-2 0x5 = Key-2

	0x6 = Fill-3/C 0x7 = Key-3/D
--	---------------------------------

UF Channel Input Tally UF

Note: This command is only valid for LGK-3901 and DSK-3901.

This tally is enabled with the 'Enable Video Tallies' (Y61) command.

Once enabled, it returns information about which of the SDI inputs (PGM or Fill-2/B) is shown on the PVW output.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
UF	%1x: A/B mixer 0x0 (always)	%1x: Output 0x1 = PVW	%1x: Source 0x0 = PGM In 0x4 = Fill-2/B

UI Mixer Transition Duration Tally UI

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

This tally is enabled with the 'Enable Mixer Transition Duration Tallies' (YUI1) command.

Once enabled, it returns information about the A/B mixer transition rates (UI) for the channel transition modes; automation, slow, medium, fast.

On registration, tallies are sent for each mode so that automation can record the initial A/B mixer rates.

The format of the tally is:

CMD	Param_1	Param_2
UI	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast	%03x: Duration (fields/frames) 0x00 - 0x3E7

YUI Enable Mixer Transition Duration Tallies

YUI

This command enables or disables 'mixer transition duration' (UI) tallies.

CMD	Param_1
YUI	%1x: Tally enable 0x0 = No 0x1 = Yes

YUI Enquire Mixer Transition Duration Tallies

YUI

This command enquires the status of the 'mixer transition duration' (YUI) tallies:

CMD
YUI

The format of the response command is:

CMD	Param_1
YUI	%1x: Tally enable 0x0 = No 0x1 = Yes

UJ Channel Transition Mode Tally UJ

Note: UH, UI and UJ commands require Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 4.x and above.

This tally is enabled with the 'Enable Channel Transition Mode Tallies' (YUJ1) command.

Once enabled, it returns information about the current channel transition mode (UJ), automation, slow, medium or fast.

On registration, a tally is sent so that automation can record the initial channel transition mode state.

The format of the tally is:

CMD	Param_1
UJ	%02x: Transition mode 0x01 = Automation 0x02 = Slow 0x03 = Medium 0x04 = Fast

Note: In addition to A/B mixing, the channel transition mode applies to keyer fades, and voiceovers.

YUJ Enable Channel Transition Mode Tallies

YUJ

This command enables or disables 'channel transition mode' (UJ) tallies.

CMD	Param_1
YUJ	%1x: Tally enable 0x0 = No 0x1 = Yes

YUJ Enquire Channel Transition Mode Tallies

YUJ

This command enquires the status of the 'channel transition mode' (YUJ) tallies:

CMD
YUJ

The format of the response command is:

CMD	Param_1
YUJ	%1x: Tally enable 0x0 = No 0x1 = Yes

DVE

The current Oxtel series product to support DVE is Imagestore 750. The Imagestore 750 supports dual-window, 2D DVE's independently on PGM and PVW.

(For earlier Oxtel products (Imagestore 3/2) please refer to earlier Oxtel protocol documentation, where DVE options included "Squeezy" and a Ross 2D/3D DVE.)

These commands relate to the 2D DVE hardware option on Imagestore 750

The 'DVE sequence' represents a time-lined list of user defined keyframes. Each keyframe contains all the information for that particular frame, for example size, position, border softness, colour, lighting and transparency.

The DVE interpolates (morphs) between successive keyframes over a user specified duration. There is a total limit of 25 keyframes per DVE sequence.

All products that support DVE allow two DVE windows to be visible on the program channel. The Imagestore 750 has two additional independent DVE windows for the preview channel.

Commands and Responses

W1 Set PGM DVE Mode W1

This command selects the DVE mode which determines where in the PGM channel signal flow the DVE hardware is configured.

CMD	Param_1
W1	%02x: DVE mode (PGM)

The available DVE modes depend on the product.

Imagestore 750 v3.1 (and below)

Mode	Name	DVE Sources 1+2	DVE Position
0x00	None (delay)	-	-
0x04	Enabled	Configurable: AB-mix, A, B, C, D Fill-1, Key-1, Fill-2, Key-2	Configurable: K1K2 DVE K3K4 K1 DVE K2K3K4 DVE K1K2K3K4
0x09	Bypass	-	-

Note: For Imagestore 750, the 'W1' command only applies the PGM bus. The PVW bus DVE mode is set using the 'WP' command.

Note: When 'None (Delay)' is selected, 1 frame (interlaced) or 2 frames (progressive) of compensating video delay is added to the background video. This way, when another DVE mode (other than 'Bypass') is selected, there will be no timing glitches. Audio should also be delayed to compensate to prevent any lip-sync issues.

Note: When DVE is enabled, the DVE sources can be changed using the 'WR' command, and the DVE position can be changed using the 'WX' command.

Imagestore 750 v3.1.2 (and above)

Mode	Name	DVE Sources 1+2	DVE Position
0x04	Enabled	Configurable (WR): AB-mix, A, B, C, D Fill-1, Key-1, Fill-2, Key-2	Configurable (WX): K1K2 DVE K3K4 K1 DVE K2K3K4 DVE K1K2K3K4
0x09	Disabled	-	-

Note: For Imagestore 750, the 'W1' command only applies the PGM bus. The PVW bus DVE mode is set using the 'WP' command.

Note: The mode 'None (Delay)' is no longer supported from v3.1.2. The compensating video delay for the background video is instead configured using the post-AB-mix video frame delay on PGM (Xq). This should be set to 1 frame (interlaced) or 2 frames (progressive). This way, when another DVE mode is enabled and disabled, there

will be no timing glitches. Audio should also be delayed to compensate to prevent any lip-sync issues.

Note: When DVE is enabled, the DVE sources can be changed using the 'WR' command, and the DVE position can be changed using the 'WX' command.

W2 Load and Run DVE Sequence W2

This command loads a sequence and plays it over a specified duration.

CMD	Param_1	Param_2
W2	%02x: Sequence 0x00 = Reverse sequence 0x01 – 0x61 0x62 – 0x63 (Imagestore 3 only)	%03x : Duration (frames) (optional) 0x001 = Default duration 0x002 – 0x3E6 = Stretch 0x3E7 = Show first frame

Param_3
%1x: Bus (bitwise) (optional) – Imagestore 750 only 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

When the duration is not specified or 0x001, the sequence will play out over the default duration specified within the DVE Editor application.

When the duration is set between 0x002 and 0x3e6 (998 decimal) then the whole sequence will be stretched to play within the duration specified in frames.

Product	Interlaced	Progressive
Imagestore 750	Fields	Frames

If the duration is set as 0x3e7 (999 decimal) then the sequence will be loaded without playing, and the first keyframe of the new sequence will be shown. If the sequence number is 0x00 (i.e. play in reverse) then the last keyframe of the currently loaded sequence will be shown instead.

For Imagestore 750 there is an optional bitwise DVE bus mask that allows automation to define whether commands apply to the PGM and/or PVW buses. If the mask is set to 0x3 then both PGM and PVW DVE will play their loaded sequence together. For all other products (and if this parameter is omitted for Imagestore 750) the DVE command only applies to the PGM bus.

W7 Run DVE Sequence W7

This command runs the currently loaded sequence forward over a specified duration.

Setting the duration to 0x001 runs the sequence at the speed for which it was designed in the DVE Editor application. Otherwise the total sequence duration is set explicitly up to 999 (0x3E7). The duration is always in frames. The DVE mask option is as defined in the W2 section.

CMD	Param_1	Param_2 (Optional)
W7	%03x : Duration (frames) 0x001 = Default duration 0x002 - 0x3E7 = Stretch	%1x: Bus (bitwise) (optional) – Imagestore 750 only 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

W8 Run DVE Sequence in Reverse W8

This command runs the currently loaded sequence in reverse over a specified duration.

Setting the duration to 0x001 runs the sequence at the speed for which it was designed in the DVE Editor application. Otherwise the total sequence duration is set explicitly up to 999 (0x3E7). The duration is always in frames.

The DVE mask option is available is as defined in the W2 section.

CMD	Param_1	Param_2 (Optional)
W8	%03x : Duration (frames) 0x001 = Default duration 0x002 - 0x3E7 = Stretch	%1x: Bus (bitwise) (optional) – Imagestore 750 only 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

WP Set PVW DVE Mode WP

This command sets the DVE mode for the Preview bus. It is equivalent to the W1 command, but for the PVW bus.

CMD	Param_1
WP	%02x: DVE mode (PVW)

Imagestore 750 v3.1 (and below)

Mode	Name	DVE Sources 1+2	DVE Position
0x00	None (delay)	-	-
0x04	Enabled	Configurable: AB-mix, A, B, C, D Fill-1, Key-1, Fill-2, Key-2	Configurable: K1K2 DVE K3K4 K1 DVE K2K3K4 DVE K1K2K3K4
0x09	Bypass	-	-

Note: For Imagestore 750, the 'WP' command only applies the PVW bus. The PGM bus DVE mode is set using the 'W1' command.

Note: When 'None (Delay)' is selected, 1 frame (interlaced) or 2 frames (progressive) of compensating video delay is added to the background video. This way, when another DVE mode (other than 'Bypass') is selected, there will be no timing glitches. Audio should also be delayed to compensate to prevent any lip-sync issues.

Note: When DVE is enabled, the DVE sources can be changed using the 'WR' command, and the DVE position can be changed using the 'WX' command.

Imagestore 750 v3.1.2 (and above)

Mode	Name	DVE Sources 1+2	DVE Position
0x04	Enabled	Configurable (WR): AB-mix, A, B, C, D Fill-1, Key-1, Fill-2, Key-2	Configurable (WX): K1 K2 DVE K3 K4 K1 DVE K2 K3 K4 DVE K1 K2 K3 K4
0x09	Disabled	-	-

- Note:** For Imagestore 750, the 'WP' command only applies the PVW bus. The PGM bus DVE mode is set using the 'WP1' command.
- Note:** The mode 'None (Delay)' is no longer supported from v3.1.2. The compensating video delay for the background video is instead configured using the post-AB-mix video frame delay on PGM (Xq). This should be set to 1 frame (interlaced) or 2 frames (progressive). This way, when another DVE mode is enabled and disabled, there will be no timing glitches. Audio should also be delayed to compensate to prevent any lip-sync issues.
- Note:** When DVE is enabled, the DVE sources can be changed using the 'WR' command, and the DVE position can be changed using the 'WX' command.

WR Set DVE Routing Input WR

This command sets the inputs (or sources) for DVE1 and DVE2 on either the PGM or PVW bus.

Note: A dual-DVE licence option is required for using DVE2 inputs.

CMD	Param_1	Param_2	Param_3
WR	%1x: Bus 0x1 = PGM 0x2 = PVW	%1x: Source for 0x0 = DVE 1 0x1 = DVE 2	%1x: Source 0 = A 1 = B 2 = AB mix 3 = Fill 1 4 = Key 1 5 = Fill 2 6 = Key 2 7 = Fill-3/C 8 = Key-3/D

WR Enquire DVE Routing Input WR

This command enquires the inputs (or sources) for DVE1 and DVE2 on either the PGM or PVW bus.

CMD	Param_1	Param_2
WR	%1x: Bus 0x1 = PGM 0x2 = PVW	%1x: Source for 0x0 = DVE 1 0x1 = DVE 2

The information returned is as follows:

CMD	Param_1	Param_2	Param_3
WR	%1x: Bus 0x1 = PGM 0x2 = PVW	%1x: Source for 0x0 = DVE 1 0x1 = DVE 2	%1x: Source 0 = A 1 = B 2 = AB mix 3 = Fill 1 4 = Key 1 5 = Fill 2 6 = Key 2 7 = Fill-3/C 8 = Key-3/D

WV Set DVE Load Action WV

This command sets the DVE action that will occur the next time a sequence is loaded using the 'WY' command (on either the PGM or PVW bus). This provides a powerful alternative to the W2, W7 and W8 commands.

CMD	Param_1	Param_2
WV	%1x : Action 0x0 = Show first frame 0x1 = Show last frame 0x2 = Play forward 0x3 = Play reverse	%1x: Bus (bitwise) (optional) 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

If the DVE mask is missing then the action applies only to the PGM bus.

WW Enquire DVE Load Action WW

This command enquires the DVE action that will occur the next time a sequence is loaded using the 'WY' command (on either the PGM or PVW bus).

It returns the setting made by the WV command.

CMD	Param_1
WW	%1x: Bus (not bitwise) 0x1 = PGM 0x2 = PVW

The information returned is:

CMD	Param_1	Param_2
WW	%1x : Action 0x0 = Show first frame 0x1 = Show last frame 0x2 = Play forward 0x3 = Play reverse	%1x: Bus (not bitwise) 0x1 = PGM 0x2 = PVW

WX Set DVE Configuration WX

This command sets where the DVE windows are keyed relative to the keying layers. It can be used to reveal or hide graphics relative to the DVE windows.

This setting applies equally to PGM and PVW buses.

CMD	Param_1
WX	%02x : Action 0x0 = K1 K2 DVE K3 K4 0x1 = K1 DVE K2 K3 K4 0x2 = DVE K1 K2 K3 K4

Note: The DVE configuration is usually set to 0x0 or 0x1 so that there is at least one keying layer upstream of the DVE that can reveal graphics when a DVE is squeezed back.

WX Enquire DVE Configuration WX

This command enquires the position of the DVE windows between keying layers.

CMD
WX

The response command has the following format:

CMD	Param_1
WX	%02x : Action 0x0 = K1 K2 DVE K3 K4 0x1 = K1 DVE K2 K3 K4 0x2 = DVE K1 K2 K3 K4

WY Load DVE Sequence with Action WY

This command loads the specified DVE sequence to PGM or PVW, applying the DVE load action set by the WV command. The load action either shows the first frame, shows the last frame, plays the sequence forwards or in reverse.

This provides a powerful alternative to the W2, W7 and W8 commands.

CMD	Param_1	Param_2
WY	%02x: Sequence 0x01 – 0x61	%1x: Bus (bitwise) (optional) 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

If the DVE mask is missing then the action applies only to the PGM bus.

WZ Enquire DVE Sequence Name WZ

This command enquires DVE sequence names, which are defined within the DVE Editor. This is useful for control panels which need to represent sequences by a meaningful name.

CMD	Param_1	Param_2
WZ	%02x: Sequence 0x00 = Currently loaded 0x01 – 0x61	%1x: Bus (not bitwise) (optional) 0x1 = PGM 0x2 = PVW

The bus parameter only makes sense if the sequence is 0x00 (currently loaded). If the DVE mask is missing then the action applies only to the PGM bus.

The response command has the following format:

CMD	Param_1	Param_2	Param_3
WZ	%02x: Sequence 0x01 – 0x61	%1x: Bus (not bitwise) 0x1 = PGM 0x2 = PVW	%s: Sequence name

Wb Play DVE Sequence Wb

This command plays a sequence forwards or backwards on the PGM or PVW bus (or both).

It is a useful alternative to the W7 and W8 commands. Also the command is armable so it can be used by control panels.

CMD	Param_1	Param_2
Wb	%02x: Sequence 0x00 = Currently loaded 0x01 – 0x61	%1x: Bus (bitwise) (optional) 0x1 = PGM 0x2 = PVW 0x3 = PGM PVW

If the DVE mask is missing then the action applies only to the PGM bus.

Wd Set DVE ID Wd

This command associates a DVE sequence number with a DVE ID. Panels use fixed DVE IDs for each panel button, and the Imagestore device maintains mappings to the required DVE sequence.

CMD	Param_1	Param_2
Wd	%02x: DVE ID 0x00 – 0xFF	%02x: Sequence 0x01 – 0x61

Note: See the Imagestore 750 user manual for more details about configuration of “DVE Mode” using the Imagestore Configurator.

Wd Enquire DVE ID Count Wd

This command enquires for a count of DVE ID to sequence mappings.

CMD
Wd

The response gives the highest DVE ID in use, minus one. This then allows automation to enumerate the DVE sequences using the next 'Wd' command.

CMD	Param_1
Wd	%02x: DVE ID Count 0x00 – 0xFF

Wd Enquire DVE ID Wd

This command enquires for the DVE sequence number associated with a DVE ID. It is predominantly for use by control panels.

CMD	Param_1
Wd	%02x: DVE ID 0x00 – 0xFF

The response command has the following format:

CMD	Param_1	Param_2
-----	---------	---------

Wd	%02x: DVE ID 0x00 – 0xFF	%02x: Sequence 0x01 – 0x61
----	-----------------------------	-------------------------------

Oxtel v16-001 PROVISIONAL

Unsolicited Tallies

YWb Enable DVE Tallies YWb

This command enables or disables DVE tallies (Wb, W1 and WP) for the serial or network port on which the command was received.

CMD	Param_1
YWb	%1x: Tally enable 0x0 = No 0x1 = Yes

YWb Enquire DVE Tallies YWb

This command enquires whether DVE tallies (Wb, W1 and WP) are enabled, or not.

CMD
YWb

The response command shows the current tally state:

CMD	Param_1
YWb	%1x: Tally enable 0x0 = No 0x1 = Yes

Wb DVE Sequence Tallies Wb

This tally is enabled with the 'Enable DVE Tallies' (YWb) command.

Once enabled, it returns information about the playing state of DVEs on both PGM and PVW as they are driven using the Wb command.

On registration, tallies are returned so that automation can record the initial DVE position state.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
Wb	%02x: Sequence 0x00 = Currently loaded 0x01 – 0x61	%1x: Position 0x0 = First frame 0x1 = Last frame 0x2 = Between	%1x: Bus 0x1 = PGM 0x2 = PVW

W1 PGM DVE Mode Tallies W1

This tally is enabled with the 'Enable DVE Tallies' (YWb) command.

Once enabled, it returns information about the PGM DVE mode as it is driven using the W1 command.

On registration, a tally is returned so that automation can record the initial PGM DVE mode.

The format of the tally is:

CMD	Param_1
W1	%02x: DVE mode (PGM) 0x04 = Enabled 0x09 = Disabled

WP PVW DVE Mode Tallies WP

This tally is enabled with the 'Enable DVE Tallies' (YWb) command.

Once enabled, it returns information about the PVW DVE mode as it is driven using the WP command.

On registration, a tally is returned so that automation can record the initial PVW DVE mode.

CMD	Param_1
W9	%02x: DVE mode (PVW) 0x04 = Enabled 0x09 = Disabled

Video Preview

These commands define video preview functions such as controlling video on the Preview bus, or making video selections for PVW, CLN, MON outputs.

Commands and Responses

Y1 Set Preview Keyer Y1

This command cuts up or down the PVW keyer for the specified layer.

When a media file (image, animation, Easytext crawl or bugclock) is loaded into a store for a keying layer, or external fill/key is associated with the keying layer, the resulting graphics are shared between keyers on PGM and PVW buses. By cutting up the PVW keyer with this command, the media can be previewed before it is brought to air on the PGM bus.

CMD	Param_1	Param_2
Y1	%1d: Layer	%1d: Direction 0 = Keyer off 1 = Keyer on

Example:

```
void SetPreviewKeyer(int Layer, int Dir)
{
    remote_send("Y1%1d %1d", Layer, Dir);
}
```

Y3 Set Preview A/B Mix Y3

This command sets which of the A/B mixer inputs is shown on the PVW output.

This is useful for previewing a mix just before an A/B mix transition is performed on the PGM bus.

CMD	Param_1
Y3	%d: A/B mixer input for PVW 0 = A input

1 = B input

Example:

```
void SetPreviewMixer(int ABMixer)
{
    remote_send("Y3%d", ABMixer);
}
```

Yp Set Active Preview Keyers Yp

This command enables or disables active preview mode for keyers within the arm/disarm/take environment (see page 343).

When active preview is enabled, if a PGM keyer is armed to cut/fade up or down, the corresponding PVW keyer changes state to show what will occur on PGM when the next 'take' occurs. If the PGM keyer is then disarmed, the corresponding PVW keyer changes state to match the PGM keyer.

Note: The 'Y1' command should not be used when active preview of keyers is enabled.

CMD	Param_1
Yp	%x: Enable 0x0 = No 0x1 = Yes

Yp Enquire Active Preview Keyers Yp

This command enquires whether active preview mode for keyers is enabled.

CMD
Yp

The response shows the current active preview keyer state.

CMD	Param_1
Yp	%x: Enable 0x0 = No 0x1 = Yes

XH Set Preview Output XH

This command sets the current preview mode for the PVW output.

The preview modes supported depend on the product and licence options.

Please check individual product manuals for details.

CMD	Param_1
XH	%02x: Preview mode 0x00 = Store 1 Key 0x01 = Store 2 Key 0x02 = PGM DSK 1 output 0x03 = PGM output 0x04 = Store 2 Fill 0x05 = Store 1 Fill 0x06 = SDI A 0x07 = SDI Fill-1 0x08 = SDI Key-1 0x09 = SDI B 0x0A = PGM A/B mixer output 0x0B = PVW output (default) 0x0C = SDI Fill-2 0x0D = SDI Key-2 0x0E = PGM DVE output 0x0F = PGM DSK 2 output 0x10 = PGM DSK 3 output 0x11 = PGM DSK 4 output 0x12 = PGM DSK 5 output (unused) 0x13 = Store 3 Fill 0x14 = Store 3 Key

Example:

```
void SetPreview(int Preview)
{
    remote_send("XH%02x", Preview);
}
```

XH Enquire Preview Output XH

This command enquires for the current preview mode.

CMD
XH

The response has the following format:

CMD	Param_1
XH	%02x: Preview mode 0x00 = Store 1 Key 0x01 = Store 2 Key 0x02 = PGM DSK 1 output 0x03 = PGM output 0x04 = Store 2 Fill 0x05 = Store 1 Fill 0x06 = SDI A 0x07 = SDI Fill-1 0x08 = SDI Key-1 0x09 = SDI B 0x0A = PGM A/B mixer output 0x0B = PVW output (default) 0x0C = SDI Fill-2 0x0D = SDI Key-2 0x0E = PGM DVE output 0x0F = PGM DSK 2 output 0x10 = PGM DSK 3 output 0x11 = PGM DSK 4 output

XP Set Clean Feed Output XP

This command sets a source for the clean feed output, if one is supported.

CMD	Param_1
XP	%2x: Clean feed source <u>Imagestore 750</u> 0x00: Program (factory default)

0x01: SDI A
0x02: SDI B
0x03: SDI C
0x04: SDI D
0x05: Program AB Mixer
0x06: Program DVE output
0x07: Program DSK 1 output
0x08: Program DSK 2 output
0x09: Program DSK 3 output
0x0A: Program DSK 4 output (unused)
0x0B = Program DSK 5 output (unused)
0x0C = Store 1 Fill
0x0D = Store 1 Key
0x0E = Store 2 Fill
0x0F = Store 2 Key
0x10 = Store 3 Fill
0x11 = Store 3 Key
0x12 = Store 4 Fill
0x13 = Store 4 Key
0x14 = SDI Fill-1
0x15 = SDI Key-1
0x16 = SDI Fill-2
0x17 = SDI Key-2
<u>Imagestore 300+</u>
0x00: Program (factory default)
0x01: SDI A
0x02: SDI B
0x03: AB Mixer
0x04: AB Mixer + DVE
0x05: DSK 1

Note: For Imagestore 750 LOGO this command will be ineffective unless the ISCF licence is enabled.

Example:

```
void SetCleanFeed(int CleanFeed)
{
```

```
remote_send("XP%2x", CleanFeed);
}
```

XP Enquire Clean Feed Output XP

This command enquires the clean feed source that is currently selected.

CMD
XP

The clean feed source is returned in the response command:

CMD	Param_1
XP	%2x: Clean feed source <u>Imagestore 750</u> 0x00: Program (factory default) 0x01: SDI A 0x02: SDI B 0x03: SDI C 0x04: SDI D 0x05: Program AB Mixer 0x06: Program DVE output 0x07: Program DSK 1 output 0x08: Program DSK 2 output 0x09: Program DSK 3 output <u>Imagestore 300+</u> 0x00: Program (factory default) 0x01: SDI A 0x02: SDI B 0x03: AB Mixer 0x04: AB Mixer + DVE 0x05: DSK 1

XT Set Monitor Feed Output XT

This command sets the source for the monitor feed output.

CMD	Param_1
XT	%02x: Monitor feed source 0x00: Program output (factory default) 0x01: Preview output 0x02: SDI A 0x03: SDI B 0x04: SDI C 0x05: SDI D 0x06: PGM A/B Mixer 0x07: PVW A/B Mixer 0x08: Store 1 Fill 0x09: Store 1 Key 0x0A: Store 2 Fill 0x0B: Store 2 Key 0x0C: Store 3 Fill 0x0D: Store 3 Key 0x0E: Store 4 Fill 0x0F: Store 4 Key 0x10: SDI Fill-1 0x11: SDI Key-1 0x12: SDI Fill-2 0x13: SDI Key-2 0x14: Colour 1 0x15: Colour 2 0x16: Colour 3 0x17: Colour 4 (V-Fade) 0x18: Test Pattern 1 0x19: Test Pattern 2 0x1A: PGM DVE 1 0x1B: PGM DVE 2 0x1C: PVW DVE 1 0x1D: PVW DVE 2 0x1E: PGM DVE output

0x1F: PGM DSK 1 output
0x20: PGM DSK 2 output
0x21: PGM DSK 3 output

For IS750 LOGO this command will be ineffective unless the ISMF licence is enabled.

XT Enquire Monitor Feed Output XT

This command sets the source for the monitor feed output.

CMD
XT

The monitor feed source is returned in the response command:

CMD	Param_1
XT	%02x: Monitor feed source 0x00: Program output (factory default) 0x01: Preview output 0x02: SDI A 0x03: SDI B 0x04: SDI C 0x05: SDI D 0x06: PGM A/B Mixer 0x07: PVW A/B Mixer 0x08: Store 1 Fill 0x09: Store 1 Key 0x0A: Store 2 Fill 0x0B: Store 2 Key 0x0C: Store 3 Fill 0x0D: Store 3 Key 0x0E: Store 4 Fill 0x0F: Store 4 Key 0x10: SDI Fill-1 0x11: SDI Key-1 0x12: SDI Fill-2

0x13: SDI Key-2
0x14: Colour 1
0x15: Colour 2
0x16: Colour 3
0x17: Colour 4 (V-Fade)
0x18: Test Pattern 1
0x19: Test Pattern 2
0x1A: PGM DVE 1
0x1B: PGM DVE 2
0x1C: PVW DVE 1
0x1D: PVW DVE 2
0x1E: PGM DVE output
0x1F: PGM DSK 1 output
0x20: PGM DSK 2 output
0x21: PGM DSK 3 output

Audio

These audio commands are supported by the following Oxtel products provided they are equipped with an appropriate Easysound audio option:

- Imagestore 750
- DSK-3901
- LGK-3901
- ISM-3901

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 running software version 3.x (and higher) have additional support for Advanced Audio Commands.

Note: Advanced Audio Commands are described on page 226 . These take advantage of the Imagestore Configurator Tool's extensive 'Audio Graph' features, including support for wide audio.

Note: Automation should use the advanced audio commands whenever possible.

All audio commands listed in this section are preceded by a lower case 'j'. Further parameters allow automation to control the following:

- Audio inputs
- Audio outputs
- Voice-overs
- Transitions
- Audio preview

A set of enquiry commands are also included to enable feedback of the audio parameters.

Note: Earlier products used "mixfiles" to configure the audio.

For the Imagestore 750 and LGK-3901, DSK-3901 and ISM-3901 audio configuration must be set up via an "Audio Graph" with the Imagestore Configurator tool,

Many of the following audio commands rely on a mixfile having been set up correctly for them to work properly. This applies to Imagestore 300[+], Imagestore HD-TV and older Imagestore 750 products (where mixfiles are used instead of the 'Audio Graph').

Other Advanced Audio Commands

There is an [Advanced Audio Commands](#) section below, which describes newer audio commands, some of which replace older audio commands. There is a [command deprecation table](#) which lists which commands are deprecated and which advanced audio command replaces them:

Channel/Pair/Group

These audio commands only give control of 4 audio channels on inputs, voiceovers and outputs. This is equivalent to one audio group or two AES pairs, and is a limitation considering there are 16 embedded audio channels available on inputs and outputs.

Note: Embedded audio channels 5-16 can not be controlled independently because only the first 4 audio channels can be addressed by automation.

Several audio commands use a parameter called “Channel/Pair/Group” which selects how many of these 4 channels will be controlled together by the automation command:

- 0x1 = Channel 1 audio channel
- 0x2 = Pair 2 audio channels
- 0x4 = Group 4 audio channels

This is always followed by a “Channel number” parameter which selects the number of the channel, pair or group to be updated. This way, audio channels can either be updated individually (0x1), or in bulk (0x2 or 0x4).

When “Channel/Pair/Group” is 0x1, “Channel number” can range from 0x0 to 0x3. When “Channel/Pair/Group” is 0x2, “Channel number” can be 0x0 or 0x2 (pair 1 and 2 respectively). When “Channel/Pair/Group” is 0x4, “Channel number” is ignored and the whole audio group (4 channels) is used.

Channel/Pair/Groups	Valid “Channel numbers”
0x1 = Channel	0x0 0x1 0x2 0x3
0x2 = Pair	0x0 0x2
0x4 = Group	0x0

Gain Adjustments

Wherever gain values in decibels (dB) are used by automation, they are represented as an integer from 0 to 255, sent as 2 hex digits. A floating point dB value (in the range -99.5 dB to +28 dB, with 0.5 dB increments) is converted to an integer value in the following way:

```
int dBval(float dB) // dB is in range -99.5 to 28
{
    int val = (dB * 2) + 199;

    if (val < 0)
    {
        val = 0;
    }
    else if (val > 255)
    {
        val = 255;
    }

    // printf("val of dB %f is: %d", dB, val);
    return val;
}
```

A value of 0 (0x00) is equivalent to -99.5 dB, and is treated as “minus infinity”.

A value of 199 (0xC7) is equivalent to 0dB.

A value of 255 (0xFF) is equivalent to +28 dB.

Commands (Inputs)

These commands instruct the Oxtel device that some parameter of an audio input is to be changed to suit the required audio application.

Audio input channels can be shuffled, muted, trimmed, phase inverted, etc., using the parameters listed below.

CMD	Param_1	Param_2	Param_3 ... Param_n
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x0 = Shuffle 0x1 = Mute 0x2 = Trim 0x3 = Phase 0x4 = Level 0x5 = Source	Function parameters Depends on function

The first parameter is the audio input number. Each audio input consists of 4 audio channels, which is equivalent to one group or two AES pairs.

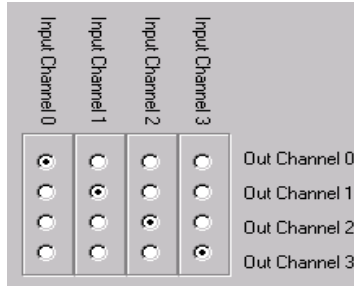
Note: Embedded audio channels 5-16 can not be controlled independently because only the first 4 audio channels can be addressed by automation.

The second parameter is the function that has to be performed.

j0_0 Set Input Shuffle j0_0

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jo' advanced audio command instead since this supports wide audio shuffles.

This command allows input channels (1-4) on the inputs to be moved around within the group. In this way pair swaps can be achieved. Each output channel can take its input from any one of the input channels.



CMD	Param_1	Param_2	Param_3	Param_4
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x0 = Shuffle	%1x: Input 0x0 - 0x3	%1x: Output 0x0 - 0x3

```
void SetInputShuffle(int Input, int ChannelIn,
                    int ChannelOut);
{
    remote_send("j0%1x0%1x%1x", Input,
                ChannelIn, ChannelOut);
}
```

Note: This command only allows each input to be routed to a single output in the shuffle matrix. This limitation is not present for the 'jg' command.

jg Set Input Shuffle Bitwise jg

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jo' advanced audio command instead since this supports wide audio shuffles.

This command allows input channels (1-4) on the inputs to be moved around within the group. In this way pair swaps can be achieved. Each output channel can take its input from any one of the input channels.

CMD	Param_1	Param_2	Param_3
jg	%1x: Audio input	%1x: Channel input	%1x: Bitwise mask

0x1 = A input 0x2 = VO input 0x3 = B input	0x0 – 0x3	0x01 = Output channel 0 0x02 = Output channel 1 0x04 = Output channel 2 0x08 = Output channel 3
--	-----------	--

This command is similar to the 'j0_0' command, but the bitwise mask parameter allows an input to be routed to multiple outputs. For example a mask value of 0xA (1010b) would connect output channels 1 and 3 to the specified input channel.

This command should be used in preference to the 'j0_0' command which only permits one output channel to be connected to each input channel.

j0_1 Set Input Mute j0_1

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jp' advanced audio command instead since this supports wide audio mutes.

The input mute command allows channels (1-4) on the inputs to be muted as channels, pairs or groups.

CMD	Param_1	Param_2
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x1 = Mute

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%1x: Mute 0x0 = No 0x1 = Yes

```
void SetInputMute(int Input, int ChannelPairGroup,
                 int Channel, int MuteOnOff);
{
    remote_send("j0%1x%1x%1x%1x", Input,
               ChannelPairGroup, Channel,
               MuteOnOff);
}
```

j0_2 Set Input Trim j0_2

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jp' advanced audio command instead since this supports wide audio trims.

The input trim command allows channels (1-4) on the inputs to be boosted or attenuated as channels, pairs or groups.

This is helpful for quiet or noisy audio sources. The adjustment of trim occurs after the 'input level'.

The range of adjustment is -99.5 dB to +28 dB in 0.5 dB steps. Whilst the large range for trim adjustment exists, it is recommended that only small values (-6 dB to +6 dB) are used.

CMD	Param_1	Param_2
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x2 = Trim

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%02x: Trim dB value 0x00 - 0xFF

```
void SetInputTrim(int Input, int ChannelPairGroup,
                 int Channel, float Trim);
{
    remote_send("j0%1x2%1x%1x%02x ", Input,
               ChannelPairGroup, Channel,
               dBval(Trim));
}
```

j0_3 Set Input Phase j0_3

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jp' advanced audio command instead since this supports wide audio phase adjustments.

The input phase command allows channels (1-4) on the inputs to be phase inverted as channels, pairs or groups. A phase invert value of 0x0 enables the selected audio to pass unaffected whilst a phase invert value of 0x1 inverts the wave form.

CMD	Param_1	Param_2
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x3 = Phase

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%1x: Invert 0x0 = No 0x1 = Yes

```
void SetInputPhase(int Input, int ChannelPairGroup,
                  int Channel, int PhaseInvert);
{
    remote_send("j0%1x3%1x%1x%1x ", Input,
               ChannelPairGroup, Channel,
               PhaseInvert);
}
```

j0_4 Set Input Level j0_4

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jp' advanced audio command instead since this supports wide audio level adjustments.

The input level command allows channels (1-4) on the inputs to be boosted or attenuated as channels, pairs or groups.

This is helpful for quiet or noisy audio sources. The adjustment of level occurs before the 'input trim'.

The range of adjustment is -99.5 dB to +28 dB in 0.5 dB steps. Whilst the large range for trim adjustment exists, it is recommended that only small values (-6 dB to +6 dB) are used.

CMD	Param_1	Param_2
j0	%1x: Audio input	%1x: Function

0x1 = A input 0x2 = VO input 0x3 = B input	0x4 = Level
--	--------------------

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%02x: Level dB value 0x00 - 0xFF

```
void SetInputLevel(int Input, int ChannelPairGroup,
                  int Channel, float Level);
{
    remote_send("j0%1x4%1x%1x%02x", Input,
                ChannelPairGroup, Channel,
                dbVal(Level));
}
```

j0_5 Set Input Source j0_5

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 do not require this command since advanced audio commands support wide audio shuffles.

The input source command allows the inputs to pick audio from either the incoming AES streams, or a group of audio embedded with their own SDI video.

CMD	Param_1	Param_2	Param_3	Param_4
j0	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input	%1x: Function 0x5 = Source	%1x: Pair 0x1 0x2	%1x: Source 0x0 = Group 1 0x2 = Group 2 0x4 = Group 3 0x6 = Group 4 0x8 = AES

```
void SetInputTrim(int Input, int Pair, int Source);
{
```

```
remote_send("j0%1x5%1x%1x", Input, Pair,  
            Source);  
}
```

Oxtel v16-001 PROVISIONAL

Commands (Outputs)

These commands instruct the Oxtel device that some parameter of the PGM audio output is to be changed to suit the required audio application.

All output command strings start with the string 'j1'. This command instructs the Oxtel device that some parameter of an audio output is to be changed to suit the required audio application.

Audio output channels can be gain adjusted, muted or embedded using the parameters listed below.

CMD	Param_1	Param_2	Param_3 ... Param_n
j1	%1x: Audio output 0x0 = PGM	%1x: Function 0x0 = Level 0x1 = Mute 0x2 = Embed 0x3 = Level via gain type	Function parameters Depends on function

j1_0 Set Output Level j1_0

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports wide audio level adjustments.

The output level command allows channels (1-4) on the PGM output to be boosted or attenuated as channels, pairs or groups.

This is helpful for quiet or noisy audio outputs.

The range of adjustment is -99.5 dB to +28 dB in 0.5 dB steps. Whilst the large range for trim adjustment exists, it is recommended that only small values (-6 dB to +6 dB) are used.

CMD	Param_1	Param_2
j1	%1x: Audio output 0x0 = PGM	%1x: Function 0x0 = Level

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2	%02x: Level dB value 0x00 - 0xFF

0x4 = Group

G = 0x0

```
void SetOutputLevel(int ChannelPairGroup,
                   int Channel, float Level);
{
    remote_send("j100%1x%1x%02x",
               ChannelPairGroup, Channel,
               dBVal(Level));
}
```

j1_3 Set Output Level via Gain Type j1_3

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports wide audio level adjustments.

This command extends output gain control for all audio channels (embedded and AES); not just channels within the first group.

It is supported by all Imagestore devices except Imagestore 2/2U/3.

CMD	Param_1	Param_2
j1	%1x: Audio output 0x0 = Unused	%1x: Function 0x3 = Level via Gain Type

Param_4	Param_4
%02x: Output gain type 0x0 = PROGRAM_GAIN	%02x: Level dB value 0x00 - 0xFF

```
void SetOutputLevelGainType(int GainType,
                           float Level);
{
    remote_send("j103%02x%02x", GainType, dBVal(Level));
}
```

Example automation commands for PROGRAM_GAIN:

- j1030000: -infinity dB
- j1030001: -99.0 dB
- j10300C7: 0.0 dB

- j10300FF: +28.0 dB

j1_1 Set Output Mute j1_1

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports wide audio mutes.

The input mute command allows channels (1-4) on the output to be muted as channels, pairs or groups.

CMD	Param_1	Param_2
j1	%1x: Audio output 0x0 = PGM	%1x: Function 0x1 = Mute

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%1x: Mute 0x0 = No 0x1 = Yes

```
void SetInputMute(int ChannelPairGroup,
                 int Channel, int Mute);
{
    remote_send("j101%1x%1x%1x",
               ChannelPairGroup, Channel, Mute);
}
```

j1_2 Output Embed j1_2

Note: This command is not supported by Imagestore 750, DSK-3901, LGK-3901 and ISM-3901. When using advanced audio commands these products would have no need for the command anyway.

This command embeds the outgoing audio into one of the four audio groups in the output video.

CMD	Param_1	Param_2	Param_3
j1	%1x: Audio output	%1x: Function	%1x: Group

	0x0 = PGM	0x2 = Embed	0x0 = Group 1 0x2 = Group 2 0x4 = Group 3 0x6 = Group 4
--	-----------	--------------------	--

```
void SetOutputEmbed(int Group);  
{  
    remote_send("j102%1x", (Group -1) * 2);  
}
```

Commands (Voice-Overs)

These commands instructs the Oxtel device to mix up to 2 audio voiceovers into the background source audio.

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on both PGM and PVW.

All voiceover command strings start with the string 'j2'. This command instructs the Oxtel device that some parameter of an audio voiceover is to be changed to suit the required audio application.

Audio voiceovers can be adjusted using the parameters listed below.

CMD	Param_1	Param_2	Param_3 ... Param_n
j2	%1x: VO 0x1 = VO1 0x2 = VO2	%1x: Function 0x0 = Voiceover duck 0x1 = Voiceover preset 0x2 = Voiceover level	Function parameters Depends on function

A variety of configurations are available for different products:

Imagestore 2/3

A/B mix mode provides two stereo pair voice-overs that can be inserted into the result of the mix. The voice-over in the A/B mix mode is named VO1.

Two voice-over mode provides four stereo pair voice-overs that can be inserted into the background or program audio. The voice-overs in 'two voice-over mode' are named VO1 and VO2. This mode is rarely used since one SDI input is lost to provide inputs for the second voice-over.

Other Imagestore devices

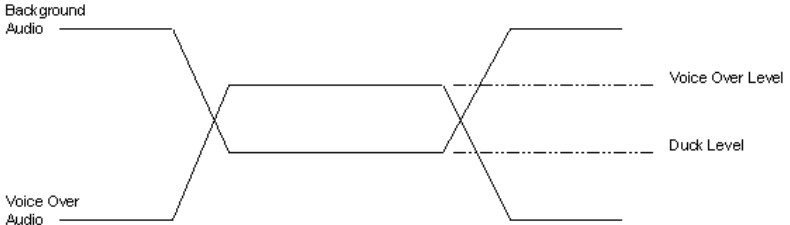
There are two voice-overs (one of which may be an Easyplay voice-over). Each voice-over may contain multiple channels. There is therefore no concept of being able to control separate stereo pairs within a single voice-over like with Imagestore 2/3.

Each voice-over has two parameters that need to be considered.

Firstly the duck level is the amount by which the background audio is attenuated by when a voice-over becomes active. This level is measured in dB. For example, when the voice-over is on, the background 'ducks' by -12dB.

The second parameter is the voice-over preset level. This represents the level of the voice-over audio when the voice-over becomes active. For example, when the voice-over is on, the voice-over audio comes up to 0dB.

These levels are represented by the following diagram:



j2_0 Set Voice-over Duck j2_0

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq0' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the attenuation of the background audio when the voice-over is on.

CMD	Param_1	Param_2
j2	%1x: Voicever 0x1 = VO1 0x2 = VO2	%1x: Function 0x0 = Duck level

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%02x: Duck level dB value 0x00 - 0xFF

```
void SetDuck(int VO, int ChannelPairGroup,
             int Channel, float Duck);
{
    remote_send("j2%1x0%1x%1x%02x ", VO,
               ChannelPairGroup, Channel,
               dBval(Duck));
}
```

j2_1 Set Voice-over Preset j2_1

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the preset level of the voice-over audio when the voice-over is on.

CMD	Param_1	Param_2
j2	%1x: Voicever 0x1 = VO1 0x2 = VO2	%1x: Function 0x1 = Voiceover preset

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%02x: VO level dB value 0x00 - 0xFF

```
void SetDuck(int VO, int ChannelPairGroup,
             int Channel, int Float);
{
    remote_send("j2%1x1%1x%1x%02x ", VO,
               ChannelPairGroup, Channel,
               dBval(Voice));
}
```

j2_2 Set Voice-over Position j2_2

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq3' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command allows the voice-over to be turned fully on (using the duck level and preset levels), off or set to an intermediate position.

Normally the voice-over would be controlled by the fade/cut voice-over transition commands, but this command allows voiceovers to be controlled via a T-bar or external fader. With level of 0 (0x00) the voiceover is off, and at 255 (0xFF) the voiceover is fully on – (the VO input will have reached its preset level and the background will be ducked to the duck level).

CMD	Param_1	Param_2
j2	%1x: Voicever 0x1 = VO1 0x2 = VO2	%1x: Function 0x2 = Voiceover position

Param_3	Param_4	Param_5
%1x: Channel/Pair/Group 0x1 = Channel 0x2 = Pair 0x4 = Group	%1x: Channel number C = 0x0, 0x1, 0x2, 0x3 P = 0x0, 0x2 G = 0x0	%02x: VO position 0x00 - 0xFF

```
void SetVOPos(int VoiceOver, int ChannelPairGroup,
              int Channel, int VoiceOverPosition);
{
    remote_send("j2%1x2%1x%1x%02x ", VoiceOver,
                ChannelPairGroup, Channel,
                VoiceOverPosition);
}
```

Commands (Transition Rates)

The Easysound audio engine can perform audio transitions such as cutting and fading mixes, voice-overs and output levels. The transition rates (or durations) are set in fields (interlaced) or frames (progressive). The maximum transition rate is 600 fields/frames, which gives a different duration depending on the refresh rate for the current video standard.

j30 Set Silence Fade Rate j30

This command sets the fade to/from silence rate for the PGM output.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j30	%03x: Duration 0x000 - 0x3E7

Example:

```
void SetSilenceRate(int duration);
{
    remote_send("j30%03x", duration);
}
```

j31 Set A/B Mixer Fade Rate j31

This command sets the fade rate for A/B mixes on the PGM bus.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j31	%03x: Duration 0x000 - 0x3E7

Example:

```
void SetABRate(int duration);
{
    remote_send("j31%03x", duration);
}
```

j32 Set Voice-over 1 Fade Rate j32

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the fade rate for voice-over 1.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls the fade rate for voice-over 1, pair 1 only.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j32	%03x: Duration 0x000 - 0x3E7

Example:

```
void SetVO1Rate(int duration);
{
    remote_send("j32%03x", duration);
}
```

j33 Set Voice-over 2 Fade Rate j33

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the fade rate for voice-over 2.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls the fade rate for voice-over 1, pair 2 only.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j33	%03x: Duration 0x000 - 0x3E7

Example:

```
void SetVO2Rate(int duration);
{
    remote_send("j33%03x", duration);
}
```

j34 Set Voice-over 3 Fade Rate j34

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the fade rate for voice-over 3.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls the fade rate for voice-over 2, pair 1 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j34	%03x: Duration 0x000 - 0x3E7

Example:

```
void SetVO3Rate(int duration);
{
    remote_send("j34%03x", duration);
}
```

j35 Set Voice-over 4 Fade Rate j35

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command sets the fade rate for voice-over 4.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls the fade rate for voice-over 2, pair 2 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

The duration is in fields (interlaced) or frames (progressive).

CMD	Param_1
j35	%03x: Duration 0x000 - 0x3E7

Example:


```
void SetVO4Rate(int duration);  
{  
    remote_send("j35%03x", duration);  
}
```

Oxtel v16-001 PROVISIONAL

Commands (Transitions)

The Easysound audio engine can perform audio transitions such as cutting and fading mixes, voice-overs and output levels. The following commands define these transitions. Transition durations are set using the j3 commands.

j40 Cut A/B j40

This command cuts audio between the A and B sources.

CMD	Param_1
j40	%1x: Destination 0x0 = A 0x1 = B

Example:

```
void ABCut(int Destination);
{
    remote_send("j40%1x", Destination);
}
```

j41 Fade A/B j41

This command fades audio between the A and B sources using the duration specified by the 'j31' command.

CMD	Param_1
j41	%1x: Destination 0x0 = A 0x1 = B

Example:

```
void ABFade(int Destination);
{
    remote_send("j41%1x", Destination);
}
```

j42 Cut To Silence j42

This command cuts the PGM output to/from silence.

CMD	Param_1
j42	%1x: Destination 0x0 = To silence 0x1 = From silence

Example:

```
void SilenceCut(int Silence);
{
    remote_send("j42%1x", Silence);
}
```

j43 Fade To Silence j43

This command fades the PGM output to/from silence using the duration specified by the 'j30' command.

CMD	Param_1
j43	%1x: Destination 0x0 = To silence 0x1 = From silence

Example:

```
void SilenceFade(int Silence);
{
    remote_send("j43%1x", Silence);
}
```

j441 Fade Voice-over 1 j441

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq4' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command fades up (or down) voice-over 1 using the duration specified by the 'j32' command.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls fades for voice-over 1, pair 1 only.

CMD	Param_1
j441	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO1Fade(int Destination);
{
    remote_send("j441%1x", Destination);
}
```

j442 Fade Voice-over 2 j442

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq4' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command fades up (or down) voice-over 2 using the duration specified by the 'j33' command.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls fades for voice-over 1, pair 2 only.

CMD	Param_1
j442	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO2Fade(int Destination);
{
    remote_send("j442%1x", Destination);
}
```

j451 Fade Voice-over 3 j451

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq4' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command fades up (or down) voice-over 3 using the duration specified by the 'j34' command.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls fades for voice-over 2, pair 1 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

CMD	Param_1
j451	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO3Fade(int Destination);
{
    remote_send("j451%1x", Destination);
}
```

j452 Fade Voice-over 4 j452

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq4' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command fades up (or down) voice-over 4 using the duration specified by the 'j35' command.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls fades for voice-over 2, pair 2 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

CMD	Param_1
j452	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO4Fade(int Destination);
{
```

```
remote_send("j452%1x", Destination);
}
```

j461 Cut Voice-over 1 j461

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq5' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command cuts up (or down) voice-over 1.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls cuts for voice-over 1, pair 1 only.

CMD	Param_1
j461	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO1Cut(int Destination);
{
    remote_send("j461%1x", Destination);
}
```

j462 Cut Voice-over 2 j462

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq5' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command cuts up (or down) voice-over 2.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls cuts for voice-over 1, pair 2 only.

CMD	Param_1
j462	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO2Cut(int Destination);
{
    remote_send("j462%1x", Destination);
}
```

j471 Cut Voice-over 3 j471

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq5' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command cuts up (or down) voice-over 3.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls cuts for voice-over 2, pair 1 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

CMD	Param_1
j471	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO3Cut(int Destination);
{
    remote_send("j471%1x", Destination);
}
```

j472 Cut Voice-over 4 j472

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jq5' advanced audio command instead since this supports up to 8 audio voiceovers on PGM and PVW.

This command cuts up (or down) voice-over 4.

Note: For Imagestore 2/2U/3 this command has different meaning since it controls cuts for voice-over 2, pair 2 only.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 this command is unsupported.

CMD	Param_1
j472	%1x: Destination 0x0 = VO down 0x1 = VO up

Example:

```
void VO4Cut(int Destination);  
{  
    remote_send("j472%1x", Destination);  
}
```


Commands (Audio Follow)

This section highlights commands that cause audio transitions to follow video transitions.

j50 Set Silence Follow Fade-to-Black j50

This command allows audio silence to automatically follow the video 'fade to black' state.

When enabled, if the PGM video is faded to black the audio will fade to silence. When the PGM video is faded from black, the audio will fade from silence.

When disabled, fading to silence must be controlled independently.

CMD	Param_1
j50	%1x: Silence follow FTB 0x0 = No 0x1 = Yes

Example:

```
void SilenceFollowFTB(int OnOff);
{
    remote_send("j50%1x", OnOff);
}
```

j51 Set Audio A/B Follow Video A/B j51

This command allows the audio A/B mixer to automatically follow the position of the video PGM A/B mixer.

When enabled, the audio A/B mixer will follow the video A/B mixer. If video uses a V-fade, the audio mix passes through silence at the mid-point. If video uses a X-fade, the audio mix passes through 0.5 x (A+B).

When disabled, the audio A/B mixer must be controlled independently.

CMD	Param_1
j51	%1x: Audio A/B follow video A/B 0x0 = No

0x1 = Yes

Example:

```
void AudioABFollowVideoAB(int OnOff);
{
    remote_send("j51%1x, OnOff);
}
```

j52 Set Easyplay Follow Keys j52

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'k9' advanced audio command instead since this supports multiple Easyplay streams and allows Easyplay to follow PST keys.

This command configures Easyplay (stream 1) to play out audio clips based on the position of the PGM video layers.

When enabled, if a PGM keyer is cut/faded up the corresponding queue of the Easyplay stream will play out. Note that only one queue can play out at a time, and so the last one triggered takes priority.

When disabled, Easyplay must be controlled independently.

CMD	Param_1
j52	%1x: Easyplay follow keyers 0x0 = No 0x1 = Yes

Example:

```
void EasyplayFollowKeyers(int OnOff);
{
    remote_send("j52%1x, OnOff);
}
```

j53 Voice-over Follow Easyplay j53

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'k8' and 'k9' advanced audio commands instead since these support associations between multiple voiceovers and Easyplay streams.

This command configures voice-over 1 to fade-up automatically when Easyplay (stream 1) is playing out, and fade-down when the audio clip stops.

When enabled the voice-over state follows Easyplay payout.

When disabled, the voice-over must be controlled independently.

CMD	Param_1
j53	%1x: Voice-over follow Easyplay 0x0 = No 0x1 = Yes

Example:

```
void VoiceoverFollowEasyplay(int OnOff);  
{  
    remote_send("j53%1x, OnOff);  
}
```

Commands (Output Preview)

j6 Set Audio Preview j6

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio command instead. This makes use of the 'Audio Output Preview Multiplexer' block which can be customised with up to 16 'named' sources.

This command sets the source for the audio preview which is sent to the PVW output.

This allows the operator to monitor a fixed list of audio mixes and levels before taking the desired effect 'to air'. The PVW output can also be used for effect send and return applications.

Mixfiles for Imagestore 300[+], Imagestore HD-TV and Imagestore 750 (if used) must be configured correctly to correspond with the fixed list of preview sources.

CMD	Param_1
j6	%02x: Preview source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 input 0x04 = Voice-over 1 mix 0x05 = Program output 0x06 = Silence 0x07 = Test tone 0x08 = Passthrough 0xFF = Embedded input (Imagestore 750 only) <u>Imagestore 2/2U/3</u> 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 (pair 1) 0x04 = Voice-over 1 (pair 2) 0x05 = Voice-over 2 (pair 1) 0x06 = Voice-over 2 (pair 2)

0x07 = Voice-over on
0x08 = PGM copy
0x09 = Silence

Example:

```
void SelectPreview(int Preview);
{
    remote_send("j6%02x, Preview);
}
```

jh Audio Clean Feed Select jh

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio command instead. This makes use of the 'Audio Output Cleanfeed Multiplexer' block which can be customised with up to 16 'named' sources.

This command sets the source for the audio clean feed which is sent to the CLN output.

This allows the operator to monitor a fixed list of audio mixes and levels before taking the desired effect 'to air'.

Mixfiles for Imagestore 750 (if used) must be configured correctly to correspond with the fixed list of clean feed sources.

CMD	Param_1
jh	%02x: Clean feed source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 input 0x04 = Voice-over 1 mix 0x05 = Program output 0x06 = Silence 0x07 = Test tone 0x08 = Passthrough 0xFF = Embedded input (Imagestore 750 only)

ji Audio Monitor Feed Select ji

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio command instead. This makes use of the 'Audio Output Monitor Multiplexer' block which can be customised with up to 16 'named' sources.

This command sets the source for the audio monitor feed which is sent to the MON output.

This allows the operator to monitor a fixed list of audio mixes and levels before taking the desired effect 'to air'.

Mixfiles for Imagestore 750 (if used) must be configured correctly to correspond with the fixed list of monitor sources.

CMD	Param_1
ji	%02x: Monitor source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 input 0x04 = Voice-over 1 mix 0x05 = Program output 0x06 = Silence 0x07 = Test tone 0x08 = Passthrough 0xFF = Embedded input (Imagestore 750 only)

Commands and Responses (Enquires)

The j7 commands allow internal parameters and settings of the Easysound audio engine to be interrogated.

j70 Enquire Audio Input j70

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use 'jp' advanced audio tallies instead of this command.

This command enquires for a variety of parameters associated with the specified audio input.

CMD	Param_1
j70	%1x: Audio input 0x1 = A input 0x2 = VO input 0x3 = B input

Example:

```
void EnquireInputParameters(int Input)
{
    remote_send("j70%1x", Input );
}
```

The information returned is:

Format	Field	Description
%c%c%c	CMD	j70
%1x	Audio input	0x1 = A input 0x2 = VO input 0x3 = B input
%02x	Shuffle channel 0	Bitwise mask 0x01 = Output channel 0 0x02 = Output channel 1 0x04 = Output channel 2 0x08 = Output channel 3
%02x	Shuffle channel 1	Bitwise mask (as above)
%02x	Shuffle channel 2	Bitwise mask

Format	Field	Description
		(as above)
%02x	Shuffle channel 3	Bitwise mask (as above)
%02x	Level (pair 1)	dB value (see page 173)
%02x	Level (pair 2)	dB value (see page 173)
%02x	Trim (pair 1)	dB value (see page 173)
%02x	Trim (pair 2)	dB value (see page 173)
%02x	Source (pair 1)	0x00, 0x02, 0x04, 0x06 or 0x08 (Only returned for inputs 1 & 3)
%02x	Source (pair 2)	0x00, 0x02, 0x04, 0x06 or 0x08 (Only returned for inputs 1 & 3)
%1x	Phase invert (pair 1)	0x0 or 0x1
%1x	Phase Invert (pair2)	0x0 or 0x1
%1x	Input Mute (pair 1)	0x0 or 0x1
%1x	Input Mute (pair 2)	0x0 or 0x1

The 4-bit shuffle mask values for each input channel correspond to Set Audio Input Shuffles (jg) on page 176.

j71 Enquire Voice-over j71

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use 'jq' advanced audio tallies instead of this command.

This command enquires for a variety of parameters associated with the specified voice-over.

CMD	Param_1
j71	%1x: VO 0x1 = VO1 0x2 = VO2

Example:

```
void EnquireVoiceOverParameters(int VoiceOver)
{
    remote_send("j71%1x", VoiceOver );
}
```

The information returned is:

CMD	Param_1	Param_2	Param_3	Param_4
j71	%1x: VO 0x1 = VO1 0x2 = VO2	%02x: Preset dB value 0x00 - 0xFF	%02x: Duck dB value 0x00 - 0xFF	%02x: VO position 0x00 - 0xFF

j72 Enquire Outputs j72

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use 'jq' advanced audio tallies instead of this command.

This command enquires for a variety of parameters associated with the specified audio input.

CMD	Param_1
j72	%1x: Audio output 0x0 = PGM

Example:

```
void EnquireOutputParameters(int Output)
{
    remote_send("j72%1x", Output);
}
```

The information returned is:

Format	Field	Description
%c%c%c	CMD	j72
%1x	Audio output	0x0 = PGM output
%02x	Level (pair 1)	dB value (see page 173)
%02x	Level (pair 2)	dB value (see page 173)
%1x	Mute (pair 1)	0x0 = On 0x1 = Off
%1x	Mute (pair 2)	0x0 = On 0x1 = Off
%1x	Embed (pair 1)	0x0, 0x2, 0x4 or 0x6
%1x	Embed (pair 2)	0x0, 0x2, 0x4 or 0x6

j73 Enquire Rates j73

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use 'jq' advanced audio tallies to get voiceover rates.

This command enquires for the audio transition durations set by the 'j3' commands. Rates are in fields (interlaced) or frames (progressive).

CMD

j73

Example:

```
void EnquireRates()
{
    remote_send("j73");
}
```

The response command contains the following information:

CMD	Param_1	Param_2	Param_3
j73	%03x: Silence rate 0x000 - 0x3E7	%03x: A/B mix rate 0x000 - 0x3E7	%03x: VO1 rate 0x000 - 0x3E7

Param_4	Param_5	Param_6
%03x: VO2 rate 0x000 - 0x3E7	%03x: VO3 rate 0x000 - 0x3E7	%03x: VO4 rate 0x000 - 0x3E7

Note: For Imagestore 2/2U/3 this command has different meaning. Voiceovers 1-4 should read 'VO1 pair 1', 'VO1 pair 2', 'VO2 pair 1' and 'VO2 pair 2' respectively.

Note: For Imagestore 300[+], Imagestore HD-TV, Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 VO3 and VO4 are unsupported.

j74 Enquire Audio Follow Video j74

This command enquires the current audio follow video settings.

CMD

j74

Example:

```
void EnquireAudioFollow()
```

```
{
    remote_send("j74");
}
```

The response command contains the following information:

CMD	Param_1	Param_2
j74	%1x: Silence follow FTB 0x0 = No 0x1 = Yes	%1x: Audio A/B follow video A/B 0x0 = No 0x1 = Yes

j75 Enquire Audio Preview j75

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio tallies instead. This makes use of the 'Audio Output Preview Multiplexer' block which can be customised with up to 16 'named' sources.

This command enquires the source for the audio preview (j6) which is sent to the PVW output.

CMD
j75

Example:

```
void EnquirePreview()
{
    remote_send("j75");
}
```

The response command includes the preview source:

CMD	Param_1
j75	%02x: Preview source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 input 0x04 = Voice-over 1 mix 0x05 = Program output 0x06 = Silence 0x07 = Test tone

0x08 = Passthrough
0xFF = Embedded input (Imagestore 750 only)
<u>Imagestore 2/2U/3</u>
0x00 = A input
0x01 = B input
0x02 = PGM A/B mix
0x03 = Voice-over 1 (pair 1)
0x04 = Voice-over 1 (pair 2)
0x05 = Voice-over 2 (pair 1)
0x06 = Voice-over 2 (pair 2)
0x07 = Voice-over on
0x08 = PGM copy
0x09 = Silence

j76 Enquire Audio Clean Feed j76

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio tallies instead. This makes use of the 'Audio Output Cleanfeed Multiplexer' block which can be customised with up to 16 'named' sources.

This command enquires the source for the audio clean feed (jh) which is sent to the CLN output.

CMD
j76

Example:

```
void EnquireCleanFeed()
{
    remote_send("j76");
}
```

The response command includes the clean feed source:

CMD	Param_1
j76	%02x: Clean feed source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix

0x03 = Voice-over 1 input
0x04 = Voice-over 1 mix
0x05 = Program output
0x06 = Silence
0x07 = Test tone
0x08 = Passthrough
0xFF = Embedded input

j77 Enquire Audio Monitor j77

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'jn0' advanced audio tallies instead. This makes use of the 'Audio Output Monitor Multiplexer' block which can be customised with up to 16 'named' sources.

This command enquires the source for the audio monitor (ji) which is sent to the MON output.

CMD
j77

Example:

```
void EnquireMonitor()
{
    remote_send("j77");
}
```

The response command includes the monitor source:

CMD	Param_1
j77	%02x: Monitor source 0x00 = A input 0x01 = B input 0x02 = PGM A/B mix 0x03 = Voice-over 1 input 0x04 = Voice-over 1 mix 0x05 = Program output 0x06 = Silence

0x07 = Test tone
0x08 = Passthrough
0xFF = Embedded input

j78 Enquire Audio AES Status j78

This command enquires the locked status of the specified input AES pair.

CMD	Param_1
j78	%02: Pair

Example:

```
void EnquireAESStatus(int Pair)
{
    remote_send("j78%02x", Pair);
}
```

The response command has the following format:

CMD	Param_1	Param_2
j78	%02: Pair	%01: Locked 0x0 = No 0x1 = Yes

j79 Enquire Output Gain j79

This command enquires for the current output gain.

CMD	Param_1
j79	%1x: Audio output 0x0 = PGM

Example:

```
void EnquireOutputGainPGM()
{
    remote_send("j790");
}
```

The response command has the following format:

CMD	Param_1	Param_2
-----	---------	---------

j79	%1x: Audio output 0x0 = PGM	%02x: Gain level dB value 0x00 - 0xFF
-----	--------------------------------	---

Commands (A/B Mixer)

The following commands are used for controlling audio A/B mixes.

ja Audio A/B Position ja

This command sets the position of the audio A/B mixer.

This is not particularly recommended since transitions must usually transition to fully-at-A or fully-at-B. It is used for T-bar control of the A/B mixer from PresPanel.

CMD	Param_1
ja	%03x: Audio A/B mix position 0x000 – 0x200 (0x000 = Fader at B) (0x200 = Fader at A)

jb Audio A/B Mix Mode jb

This command switches between audio X-fades ($0.5 \times (A+B)$) at the mid-point and audio V-fades (silence at the mid-point).

CMD	Param_1
jb	%1x: Mix mode 0x0 = X-fade 0x1 = V-fade (through silence)

jc Audio A/B Asymmetric Transition jc

This command performs an asymmetric audio transition such as a cut-fade or fade-cut.

Rate 1 defines the duration of the first half of the fade (time take to reach the midpoint), and rate 2 is the time from the midpoint to completion. Both rates are measured in fields (interlaced) or frames (progressive).

CMD	Param_1	Param_2	Param_3
jc	%1x: Direction 0x0 = Transition to A	%03x: Rate 1	%03x: Rate 2

	0x1 = Transition to B	
--	-----------------------	--

jd Audio A/B Fade to Position jd

This command fades to the specified position over the defined duration in fields (interlaced) or frames (progressive).

This is similar to the audio A/B position command ('ja') but provides a fade to the specified position instead of a cut.

CMD	Param_1	Param_2
jd	%03x: Audio A/B mix position 0x000 – 0x200 (0x000 = Fader at A) (0x200 = Fader at B)	%03x: Duration

je Set Audio A-Fade Position je

This command allows one half (A) of the audio A/B mixer to be faded independently of the other half (B). The duration is defined in fields (interlaced) or frames (progressive).

Instead of treating the A fader and B fader as two linked faders, where one is the inverse of the other, this command allows the A fader to be manipulated independently. This can be used to perform effects such as voiceovers in the AB mixer. You might have the voiceover source presented on the B input, and the program audio coming from A. In this case, assuming the A/B mixer is cut to A, you would simply fade the B up to the required level and duck the A down.

CMD	Param_1	Param_2
je	%03x: A-fader position 0x000 - 0x200 (0x000 = Fully off) (0x200 = Fully on)	%03x: Duration

Note: The fader position is a linear gain value, rather than being expressed in dB.

jf Set Audio B-Fade Position jf

This command allows one half (B) of the audio A/B mixer to be faded independently of the other half (A). The duration is defined in fields (interlaced) or frames (progressive).

The same comments made in the 'je' command also apply.

CMD	Param_1	Param_2
jf	%03x: B-fader position 0x000 - 0x200 (0x000 = Fully off) (0x200 = Fully on)	%03x: Duration

Note: The fader position is a linear gain value, rather than being expressed in dB.

jj Set A/B Transition Switch Point jj

This command sets the transition switchpoint for the A/B mixer cut and data types.

CMD	Param_1
jj	%01x: Switch point 0x00 = Transition start 0x01 = Transition midpoint 0x02 = Transition end 0x03 = Fully at A 0x04 = Fully at B

Commands (Delays)

jl Set Audio Delay jl

This command sets a delay in samples (at 48KHz) for the specified audio delay bank.

This is used to compensate for any video delays within the system so that lip-sync is maintained. For example, DVE processing incurs video delay so PGM audio must be delayed to match.

Please refer to individual product manuals to see how many delay banks are supported. This command relies on the audio configuration (mixfile or 'Audio Graph') being set up correctly.

CMD	Param_1	Param_2	Param_3
jl	%1x: Delay bank	%1x: Enable delay 0x0 = No 0x1 = Yes	%06x: Audio delay (samples at 48KHz)

Commands (Miscellaneous)

j8 Set Audio FX Send j8

This command sets up the audio FX send and return paths, and any gain applied to the FX send path.

CMD	Param_1	Param_2	Param_3
j8	%1x: Enable 0x0 = No 0x1 = Yes	%02x: FX send mode (see below)	%02x: FX return pair 0x04 = Pair 1 0x06 = Pair 2

Param_4
%02x: FX send gain dB value 0x00 - 0xFF

For AB Mix mode, values for 'FX send mode' are as follows:

FX Send mode	Value
A Input Pair 1	0x01
A Input Pair 2	0x02
Voice-over 1 Pair 1	0x04
B Input Pair 1	0x08
B Input Pair 2	0x10
AB Mix Pair 1	0x09
AB Mix Pair 2	0x12
AB Mix Pair 1 + VO1	0x0D
AB Mix Pair 2 + VO1	0x16

And for dual voice-over mode, the following values are allowed:

FX Send mode	Value
BGND Pair 1	0x01
BGND Pair 2	0x02
Voice-over 1 Pair 1	0x04
Voice-over 2 Pair 1	0x08
Voice-over 2 Pair 2	0x10
BGND + VO1 Pair 1	0x05

FX Send mode	Value
BGND + VO2 Pair 1	0x0D
BGND + VO2 Pair 2	0x1D

Example

```
void Set_FX_Send(bool Enable, int SendMode,
                int ReturnPair, float SendGain)
{
    remote_send("j8%1x%02x%02x%02x", Enable,
                SendMode, ReturnPair, SendGain);
}
```

j9 Set Audio Data Path j9

This command specifies that a particular audio pair contains a data-stream rather than PCM audio data. This is used to enable a data 'pass-through' for systems such as Dolby E and Dolby Digital. When using this, the audio data is guaranteed to be passed unmodified, and cuts (rather than fades) are always applied to the data values.

CMD	Param_1	Param_2	Param_3
j9	%1x: Data path enable 0x0 = No 0x1 = Yes	%02x: Input pair	%02x: Output pair

Unsolicited Tallies

Y0 Audio Metering Tally Y0

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 should use the 'js' advanced audio metering tallies instead. This makes use of the 'Audio Meter' block which can be configured with up to 48 meter inputs.

This tally is enabled with the 'Enable Audio Metering Tallies' (Y01) command. Once enabled, it returns information about audio levels. The frequency of the tally depends on the metering mode set.

The format of the tally is:

CMD	Param_1
Y0	%8c: 8 channels of metering values.

All values have the most significant bit set. To convert a value to dB, use the following code:

```
unsigned char inputchar;
int dB;
dB = inputchar - (255);
```

The values seen are the dB value added to 255. This allows a metering range of -254dB (1) to 0dB (255) in 1 dB steps.

Y0 Enable Audio Metering Tallies Y0

Note: Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 can use the 'YC1js' advanced audio command to enable metering tallies. This allows metering of wide audio; not just the first 8 channels.

This command enables Y0 audio metering tallies for 8 channels and also sets the metering tally frequency.

CMD	Param_1
Y0	%1x: Metering mode <u>Imagestore 750, DSK-3901, LGK-3901, ISM-3901</u> 0x0 = Off

<p>0x1 – 0xF = Number of frames between metering tallies <u>Imagestore 2/2U/3, Imagestore 300[+], Imagestore HD-TV</u> 0x0 = Off 0x1 = Tally once every 2 fields/frames 0x2 = Tally once every 1 field/frame</p>
--

Y8 Audio Tally Y8

This tally is enabled with the 'Enable Audio Tallies' (Y81) command.

Once enabled, it returns information about audio A/B mix, fade-to-silence, voiceover 1 and voiceover 2 as these states change whilst under the control of automation.

On registration, a tally is returned so that automation can record the initial audio states.

The format of the tally is:

CMD	Param_1	Param_2
Y8	%1x: Audio A/B mix position 0x0 = A 0x1 = B 0x2 = In between	%1x: Audio fade-to-silence position 0x0 = On 0x1 = Off 0x2 = In between

Param_3	Param_4
%1x: Voiceover 1 position 0x0 = Off 0x1 = On 0x2 = In between	%1x: Voiceover 2 position 0x0 = Off 0x1 = On 0x2 = In between

Note: For Imagestore 2/2U/3, parameters 3 and 4 are “Voice Over 1 Pair 1” and “Voice Over 1 Pair 2” respectively.

Y8 Enable Audio Tallies Y8

This command enables unsolicited audio tallies for the connection on which the command was received.

Tallies relating to 'Advanced Audio' (page 226) for Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 (which have up to 8 voiceovers) require use of the 'YC' command (page 250).

CMD	Param_1
Y8	%x: Tally enable 0x0 = No 0x1 = Yes

Example:

```
void EnableAudioTallies(bool Enable)
{
    remote_send("Y8%x", Enable);
}
```

Note: Audio tallies are also enabled using the Set Passive Mode (Ya) command (page 303).

Y8 Enquire Audio Tallies Y8

This command enquires the status of the Y8 tallies.

CMD
Y8

The response is:

CMD	Param_1
Y8	%x: Tally enabled 0x0 = No 0x1 = Yes

j7 Audio Tallies j7

These tallies are enabled with the 'Enable Passive Mode Tallies' (Ya1) command.

Once enabled, they return information about a wide variety of different states used within the audio model.

Note: Use 'YC1' to enable tallies for the advanced audio model.

The following list shows all 'j7 audio tallies' which are enabled in passive mode:

- j70 Audio Input Tally
- j71 Audio Voiceover Tally
- j72 Audio Output Tally
- j73 Audio Rates Tally
- j74 Audio Follow Video Tally
- j75 Audio Preview Tally
- j76 Audio Clean Feed Tally
- j77 Audio Monitor Tally

As 'set' commands are sent by automation or panels to modify audio settings, the corresponding tally commands are sent back by the device. The format of each response exactly matches the corresponding set/enquire command, and so they are not listed individually in this section of the document.

All of the tallies listed above are also sent in bulk on registration of passive mode tallies so that automation can get an accurate representation of the initial system state.

Advanced Audio

These 'wide' audio commands are supported by the following Oxtel products provided that they are running software version 3.x (or higher) and they are equipped with an appropriate audio option.

- Imagestore 750
- DSK-3901
- LGK-3901
- ISM-3901

They supersede many of the audio commands (page 172) and provide a range of additional features based on a "wide audio" mixing model.

The main advanced audio features provide support for:

- 48 channel audio A/B mixing, shuffles, gain adjustments and multiplexor selection.
- 8 voiceovers on both PGM and PVW.
- Gain blocks to allow the gain across a number of channels to be changed as one, or for channels to be trimmed individually. Gain blocks also provide per-channel muting and phase-inversion.
- Shuffle blocks of up to 48x48 for all audio feeds, voiceovers, and outputs. These can make or break cross-points individually or in bulk using presets.
- Audio multiplexors for dynamic selection of audio preview points and metering. Each source to a multiplexor is named so that panels can represent the current selection. (Preview metering modes are no longer "fixed").
- Configurable data/cut switch point (via System Settings > Audio > AB Mixer > transition switchpoint) on the Audio A/B mixer.
- Fade-To-Silent blocks
- Automatic "Audio Type" setting in AES channel status data for AES outputs and embedded audio. This automatically sets the type as "Audio" or "Non-Audio" depending on what audio is connected.
- Multi-channel Easyplay 2 audio clip player, with simultaneous playout of up to 4 clips sharing 16 channels (Imagestore 750) or 2 clips sharing 8 channels (LGK-3901, ISM-3901).

Note: Dolby/UpMix and metadata commands for Imagestore 750 are covered on pages 278 and 280.

The Imagestore Configurator's 'Audio Graph' is always used for configuring audio routings when these commands are used. Automation commands will

only take effect if the appropriate audio block is present, so it is important to cross-reference the automation schedule against the audio configuration.

Bus Types

Advanced audio commands often specify a 'bus type' and 'bus index'. This allows one command to apply the same function to many different parts of the audio engine. For example, audio shuffles can be applied to input feeds, voiceovers or outputs.

The following table lists the different bus types supported:

%1x: Bus type	%02x: Bus index
0x0 = Input feed	0x00 = Feed 1 0x01 = Feed 2 0x02 = Feed 3 0x03 = Feed 4
0x1 = Voiceover	0x00 = Voiceover 1 (PGM) 0x01 = Voiceover 2 (PGM) 0x02 = Voiceover 3 (PGM) 0x03 = Voiceover 4 (PGM) 0x04 = Voiceover 5 (PGM) 0x05 = Voiceover 6 (PGM) 0x06 = Voiceover 7 (PGM) 0x07 = Voiceover 8 (PGM)
0x2 = Output	0x00 = PGM output 0x01 = PVW output 0x02 = CLN output 0x03 = MON output

Deprecated Audio Commands

The following table lists all of the audio commands (page 172), showing which ones are deprecated and which advanced audio command replaces it:

CMD	Deprecated	Replaced By
j0_0	Yes	jo0, jo1
j0_1	Yes	jp0
j0_2	Yes	jp1
j0_3	Yes	jp2
j0_4	Yes	jp3
j0_5	Yes	Not required
j1_0	Yes	jp3
j1_3	Yes	jp4
j1_1	Yes	jp0
j1_2	Yes	Not required
j2_0	Yes	jq0
j2_1	Yes	jq1
j2_2	Yes	jq3
j30	No	-
j31	No	-
j32	Yes	jq2
j33	Yes	jq2
j34	Yes	jq2
j35	Yes	jq2
j40	No	-
j41	No	-
j42	No	-
j43	No	-
j441	Yes	jq4
j442	Yes	jq4
j461	Yes	jq5
j462	Yes	jq5
j50	No	-
j51	No	-
j52	Yes	k9
j53	Yes	k8, k9
j6	Yes	jn0

j70	Yes	jo0, jp0, jp1, jp2, jp3, jp4
j71	Yes	jq0, jq1, jq3
j72	Yes	jp0, jp3, jp4
j73	Yes	j73, jq2
j74	No	-
j75	Yes	jn0
j76	Yes	jn0
j77	Yes	jn0
j79	Yes	jp4
j8	Yes	Not required
j9	Yes	Not required
ja	No	-
jb	No	-
jc	No	-
jd	No	-
je	No	-
jf	No	-
yg	Yes	jo0, jo1
jh	Yes	jr
ji	Yes	jn0
jj	No	-
jl	No	-
Y0	Yes	YCljs

End of Audio Command Deprecation Table

Commands and Responses (Multiplexer)

These commands perform operations on multiplexer blocks defined within the 'Audio Graph'.

The output of each multiplexer block is configured as 1 to 48 audio channels wide (usually it is 16 channels wide to match embedded SDI audio).

There are up to 16 different 'sources' configured for each multiplexer block, each the same width as the output. Each source can be custom named.

jn0 Set Multiplexer Source jn0

This command selects the source for the specified input or output multiplexer.

The source selected should not be greater than the 'source count' which is returned by the 'jn1' command.

CMD	Param_1	Param_2	Param_3
jn0	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%02x: Source 0x00 - 0x0F

jn0 Enquire Multiplexer Source jn0

This command enquires the source for the specified input or output multiplexer.

CMD	Param_1	Param_2	Param_3
jn0	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

--	--	--	--

The response includes the current source:

CMD	Param_1	Param_2	Param_3
jn0	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%02x: Source 0x00 - 0x0F

jn1 Enquire Multiplexer Source Count jn1

This command enquires the number of sources defined for the specified input or output multiplexer.

CMD	Param_1	Param_2	Param_3
jn1	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response includes the source count:

CMD	Param_1	Param_2	Param_3
jn1	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%02x: Source count 0x00 - 0x0F

jn2 Enquire Multiplexer Source Name jn2

This command enquires the name of a source for the specified input or output multiplexer.

CMD	Param_1	Param_2	Param_3
jn2	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response includes the source name:

CMD	Param_1	Param_2	Param_3
jn2	%1x: Bus type 0x0 = Input stream 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%s: Source name

Commands and Responses (Shuffles)

These commands perform a range of wide shuffling functions on shuffle blocks that can be positioned in various parts of the audio graph (input streams, voiceovers and outputs).

Shuffle blocks and shuffle presets can be defined with up to 48x48 crosspoints, either square or rectangular. Shuffles are often defined 16x16 to match the number of embedded channels in an SDI input feed.

Note: Refer to individual product manuals to see how shuffle blocks and shuffle presets are configured..

Individual crosspoints can be made or broken, or all crosspoints can be changed in bulk by applying a shuffle preset. Changing audio levels are always ramped over a frame to avoid clicks or noise.

jo0 Set Shuffle Crosspoint jo0

This command makes or breaks the specified shuffle crosspoint for an input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jo0	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4	Param_5	Param_6
%02x: Crosspoint in 0x00 - 0x2F	%02x: Crosspoint out 0x00 - 0x2F	%1x: Make crosspoint 0x0 = Break 0x1 = Make

Note: There is no enquire shuffle crosspoint command. Please use shuffle crosspoint tallies (YC1jo0) instead.

jo1 Load Shuffle Preset jo1

This command loads a named shuffle preset into a shuffle block associated with the specified input stream, voiceover or output to change crosspoints in bulk (rather than individually).

When the shuffle preset does not match the size of the shuffle block, the crosspoints that overlap from the preset are used, and the remaining crosspoints in the block are broken.

CMD	Param_1	Param_2	Param_3
jo1	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%s: Shuffle preset filename

jo1 Enquire Shuffle Preset jo1

This command enquires for the shuffle preset filename associated with the specified input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jo1	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response has the following format:

CMD	Param_1	Param_2	Param_3
jo1	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4

%s: Shuffle preset filename

jo2 Save Shuffle Preset jo2

This command saves the current crosspoints from a shuffle block into a named shuffle preset file.

In this way, shuffles that need to be reused frequently can be saved later use.

CMD	Param_1	Param_2	Param_3
jo2	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%s: Shuffle preset filename

jo3 Enquire Shuffle Preset List jo3

This command enquires for a list of shuffle presets that are available to the Imagestore device.

CMD
jo3

The response contains a list of shuffle preset filenames which are comma-separated using the ',' character.

CMD	Param_1	Param_2	Param_3...Param_n
jo3	%03x: Sequence 0x000 = 1 response 0x001 = 2 responses ...	%03x: File count	%s,: Filename 1 %s,: Filename 2 %s,: Filename 3 ...

The 'sequence' parameter is 0x000 when all of the shuffle preset filenames will fit into a single Oxtel response command (limited to 2048 characters). If more than one response command is required (because there is a very long

list of shuffle presets), then the sequence number counts down with each successive response until all shuffle preset names have been returned.

jo4 Enquire Shuffle Block Size jo4

This command enquires for the size (number of inputs and outputs) of the specified shuffle block.

CMD	Param_1	Param_2	Param_3
jo4	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response command returns the size as follows:

CMD	Param_1	Param_2	Param_3
jo4	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%02x: Size 0x01 - 0x30

Commands and Responses (Gains)

These commands perform audio gain, mute and phase adjustments on gain blocks that can be positioned in various parts of the audio graph (input streams, voiceovers and outputs).

Gain blocks and gain presets can be defined up to 48 channels wide. Gains are often defined 16 channels wide to match the number of embedded channels in an SDI input feed.

Note: Refer to individual product manuals to see how many gain blocks and gain presets are supported.

Individual audio channels can be adjusted, or all channels can be changed in bulk by applying a gain preset. Changing audio levels are always ramped over a frame to avoid clicks or noise.

jp0 Set Mute jp0

This command mutes or un-mutes the specified channel of an input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jp0	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4	Param_5
%02x: Channel 0x00 - 0x2F	%1x: Mute 0x0 = No (un-mute) 0x1 = Yes (mute)

jp1 Set Trim jp1

This command trims the level of the specified channel for an input stream, voiceover or output.

The channel trim adjustment is applied after the master level and the channel level.

CMD	Param_1	Param_2	Param_3
jp1	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4	Param_5
%02x: Channel 0x00 - 0x2F	%+07.2f: Trim level (dB)

jp2 Set Phase Invert jp2

This command phase inverts the specified channel of an input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jp2	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4	Param_5
%02x: Channel 0x00 - 0x2F	%1x: Invert 0x0 = No 0x1 = Yes

jp3 Set Level jp3

This command sets the level of the specified channel of an input stream, voiceover or output.

The channel level adjustment is applied after the master level and before the channel trim.

CMD	Param_1	Param_2	Param_3
jp3	%1x: Bus type	%02x: Bus index	%02x: Stem

	0x0 = Input stream 0x1 = Voiceover 0x2 = Output	(See page 227)	0x00 (unused)
--	---	----------------	------------------

Param_4	Param_5
%02x: Channel 0x00 - 0x2F	%+07.2f: Level (dB)

jp4 Set Master Gain jp4

This command sets the master gain level of the specified input stream, voiceover or output.

The master gain is applied before the channel level and channel trim adjustments.

CMD	Param_1	Param_2	Param_3
jp4	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%+07.2f: Master gain (dB)

jp5 Load Gain Preset jp5

This command loads a named gain preset into a the specified input stream, voiceover or output to change gains in bulk (rather than individually). When the gain preset does not match the size of the audio block, the gains that overlap from the preset are used, and the remaining gains in the block are unity.

CMD	Param_1	Param_2	Param_3
jp5	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

	0x2 = Output		
--	--------------	--	--

Param_4
%s: Gain preset filename

jp5 Enquire Gain Preset jp5

This command enquires for the gain preset filename associated with the specified input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jp5	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response has the following format:

CMD	Param_1	Param_2	Param_3
jp5	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%s: Gain preset filename

jp6 Save Gain Preset jp6

This command saves gain settings from the specified input stream, voiceover or output to a named gain preset file.

CMD	Param_1	Param_2	Param_3
jp6	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%s: Gain preset filename

jp7 Enquire Gain Preset List jp7

This command enquires for a list of gain presets that are available to the Imagestore device.

CMD
jp7

The response contains a list of gain preset filenames which are comma-separated using the ',' character.

CMD	Param_1	Param_2	Param_3...Param_n
jp7	%03x: Sequence 0x000 = 1 response 0x001 = 2 responses ...	%03x: File count	%s,: Filename 1 %s,: Filename 2 %s,: Filename 3 ...

The 'sequence' parameter is 0x000 when all of the gain preset filenames will fit into a single Oxtel response command (limited to 2048 characters). If more than one response command is required (because there is a very long list of gain presets), then the sequence number counts down with each successive response until all gain preset names have been returned.

jp8 Enquire Gain Block Size jp8

This command enquires the gain block size (audio channel width) for the specified input stream, voiceover or output.

CMD	Param_1	Param_2	Param_3
jp8	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

The response includes the block size as the final parameter:

CMD	Param_1	Param_2	Param_3
jp8	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output	%02x: Bus index (See page 227)	%02x: Stem 0x00 (unused)

Param_4
%02x: Size 0x00 - 0x30

Commands and Responses (Voiceovers)

Up to 8 voice-over blocks can be defined on both PGM and PST, and these blocks can be up to 48 channels wide. Blocks are often defined 16 channels wide to match the number of embedded channels in an SDI input feed.

Note: Refer to individual product manuals to see how many voiceover blocks are supported.

The PGM voice-over commands perform adjustments including duck level, preset level, voiceover fade rates and triggering voiceover transitions (cuts and fades).

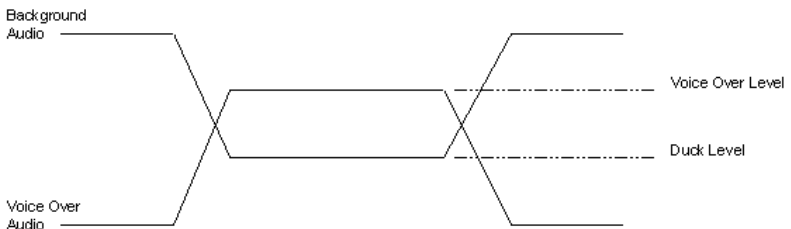
The PST voice-overs are used for previewing the corresponding PGM voice-overs in an arm/take set up. There is no direct control over PST voice-overs using automation commands. Instead they are automatically cut (never faded) up and down as the corresponding PGM voice-over is armed or disarmed to cut or fade up/down. PST voiceovers take duck and preset levels from the corresponding PGM voice-over to ensure that the preview is accurate.

Each PGM voice-over block has two parameters that need to be considered.

Firstly the duck level is the amount by which the background audio is attenuated by when a voice-over becomes active. This level is measured in dB. For example, when the voice-over is on, the background 'ducks' by -12dB.

The second parameter is the voice-over preset level. This represents the level of the voice-over audio when the voice-over becomes active. For example, when the voice-over is on, the voice-over audio comes up to 0dB.

These levels are represented by the following diagram:



PGM voice-overs also have fade rates which define how quickly the voice-over transition is performed.

Note: The commands that follow are all intended for the PGM voice-over unless otherwise stated.

jq0 Set Voice-over Duck Level jq0

This command sets the attenuation of the background audio (duck) when the specified PGM voice-over is on.

CMD	Param_1	Param_2	Param_3
jq0	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%+07.2f: Duck level (dB)

jq0 Enquire Voice-over Duck Level jq0

This command enquires the attenuation of the background audio (duck) for the specified PGM voice-over.

CMD	Param_1	Param_2
jq0	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)

The response includes the requested duck level:

CMD	Param_1	Param_2	Param_3
jq0	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%+07.2f: Duck level (dB)

jq1 Set Voice-over Preset Level jq1

This command sets the preset level of the voice-over when the specified PGM voice-over is on.

CMD	Param_1	Param_2	Param_3
jq1	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%+07.2f: Preset level (dB)

jq1 Enquire Voice-over Preset Level jq1

This command enquires the preset level of the voice-over when the specified PGM voice-over is on.

CMD	Param_1	Param_2
jq1	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)

The response includes the requested preset level:

CMD	Param_1	Param_2	Param_3
jq1	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%+07.2f: Preset level (dB)

jq2 Set Voice-over Fade Rate jq2

This command sets the fade duration of the specified PGM voice-over. Durations are defined in fields (interlaced) or frames (progressive).

CMD	Param_1	Param_2	Param_3
jq2	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%03x: Duration 0x000 - 0x3E7

jq2 Enquire Voice-over Fade Rate jq2

This command enquires the fade duration of the specified PGM voice-over.

CMD	Param_1	Param_2
jq2	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)

The response command returns the PGM voice-over fade duration:

CMD	Param_1	Param_2	Param_3
jq2	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%03x: Duration 0x000 - 0x3E7

jq3 Set Voice-over to Position jq3

This command sets a PGM voice-over to the specified position (or fader angle).

A position of 0x000 cuts down the voice-over completely, and a position of 0x200 cuts up the voice-over completely (using the duck level and preset level).

Normally the voice-over is controlled by the fade/cut voice-over transition commands.

CMD	Param_1	Param_2	Param_3
jq3	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%03x: Position 0x000 - 0x200

jq4 Fade Voice-over jq4

This command fades up (or down) the specified PGM voice-over using the duration specified by the 'jq2' command.

CMD	Param_1	Param_2	Param_3
jq4	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%1x: Direction 0x0 = Down 0x1 = Up

jq5 Cut Voice-over jq5

This command cuts up (or down) the specified PGM voice-over.

CMD	Param_1	Param_2	Param_3
jq5	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%1x: Direction 0x0 = Down 0x1 = Up

jq5 Enquire Voice-over Position jq5

This command enquires the current position of the specified PGM voice-over.

Note that voice-over positions can be tracked using 'jq5' non-solicited tallies, which are enabled with 'YC1'.

CMD	Param_1	Param_2
jq5	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)

The response returns the current PGM voiceover position:

CMD	Param_1	Param_2	Param_3
jq5	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%1x: Position 0x0 = Down 0x1 = Up 0x2 = In between

jq6 Set Voice-over Fade Rate (Panel) jq6

This command sets the fade duration of the specified PGM voice-over for one of the following panel fade modes:

- Automation (equivalent to jq2)
- Slow
- Medium
- Fast

Durations are defined in fields (interlaced) or frames (progressive).

CMD	Param_1	Param_2	Param_3
jq6	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%02x: Panel fade mode 0x00 = Automation 0x01 = Slow 0x02 = Medium 0x03 = Fast

Param_4
%03x: Duration

0x000 - 0x3E7

jq6 Enquire Voice-over Fade Rate (Panel) jq6

This command enquires the fade duration of the specified PGM voice-over for one of the following panel fade modes:

- Automation (equivalent to jq2)
- Slow
- Medium
- Fast

CMD	Param_1	Param_2	Param_3
jq6	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%02x: Panel fade mode 0x00 = Automation 0x01 = Slow 0x02 = Medium 0x03 = Fast

The response command returns the fade duration.

CMD	Param_1	Param_2	Param_3
jq6	%02x: Voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%02x: Panel fade mode 0x00 = Automation 0x01 = Slow 0x02 = Medium 0x03 = Fast

Param_4
%03x: Duration 0x000 - 0x3E7

jq7 Enquire PST Voice-over Position jq7

This command enquires the current position of the specified PST voice-over.

Note that voice-over positions can be tracked using 'jq7' non-solicited tallies, which are enabled with 'YC1'.

CMD	Param_1	Param_2
jq7	%02x: PST voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)

The response returns the current PST voiceover position:

CMD	Param_1	Param_2	Param_3
jq7	%02x: PST voiceover 0x00 - 0x07	%02x: Stem 0x00 (unused)	%1x: Position 0x0 = Down 0x1 = Up

jt0 Enquire Audio Meter Width jt0

This command enquires the number of pins that are connected to the 'Audio Meters' block in the 'Audio Graph'. This helps automation to determine the expected length of 'Audio Meter Tallies' (js) once they are enabled.

CMD
jt0

The response command has the following format:

CMD	Param_1
jt0	%02x: Width 0x00 - 0x30

Unsolicited Tallies

YC Enable Advanced Audio Tallies YC

This command enables tallies for a wide variety of different states used within the advanced audio model.

CMD	Param_1
YC	%1x: Tally enable 0x0 = No 0x1 = Yes

The following list shows all audio tallies which are enabled by the YC1 command:

- j30 Silence Fade Rate Tally
- j31 A/B Mixer Fade Rate Tally
- j42 Cut to/from Silence Tally
- j50 Silence Follow Fade-to-Black Tally
- j51 Audio A/B Follow Video A/B Tally
- j75 Audio Preview Tally
- j76 Audio Clean Feed Tally
- j77 Audio Monitor Tally
- ja Audio A/B Position Tally
- jb Audio A/B Mix Mode Tally
- jj A/B Transition Switch Point Tally
- j1 Audio Delay Tally
- jn0 Multiplexor Source Tally
- jo1 Shuffle Preset Tally
- jp0 Mute Tally
- jp4 Master Gain Tally
- jp5 Gain Preset Tally
- jq0 Voice-over Duck Level Tally
- jq1 Voice-over Preset Level Tally
- jq2 Voice-over Fade Rate Tally
- jq3 Voice-over to Position Tally
- jq5 Voice-over Tally
- jq7 PST Voice-over Position Tally

As 'set' commands are sent by automation or panels to modify audio settings, the corresponding tally commands are sent back by the device. The format of each response exactly matches the corresponding set/enquire command, and so they are not listed individually in this section of the document.

All of the tallies listed above are also sent in bulk on registration of advanced audio tallies so that automation can get an accurate representation of the initial system state.

YC Enquire Advanced Audio Tallies YC

This command enquires the status of the advanced audio tallies.

CMD
YC

The format of the response command is:

CMD	Param_1
YC	%1x: Tally enable 0x0 = No 0x1 = Yes

YC_jo0 Enable Shuffle Crosspoint Tallies YC_jo0

This command enables shuffle crosspoint (jo0) tallies within the advanced audio model. Tallies can be enabled for selected inputs, voiceovers and outputs, or all audio blocks that support shuffles.

Note: Shuffle crosspoint tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' has many inputs, voiceovers and outputs with shuffle matrices.

Note: Shuffle crosspoint tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire shuffle crosspoint tallies' command.

CMD	Param_1	CMD	Param_2	Param_3
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jo0	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output (optional)	%02x: Bus index (See page 227) (optional)

When parameters 2 and 3 are defined, only the specified audio block will return crosspoint shuffle tallies.

When parameter 3 is not defined, all audio blocks of the specified 'bus type' will return crosspoint shuffle tallies.

If parameters 2 and 3 are not defined, all audio input/voiceover/output blocks with crosspoint shuffles will return crosspoint shuffle tallies.

As automation and panels send 'set shuffle crosspoint' (jo0) or 'load shuffle preset' (jo1) commands, shuffle crosspoint (jo0) tally commands are sent back from the device for the enabled audio blocks. The format of each tally response exactly matches the 'jo0' set/enquire command.

Shuffle crosspoint tallies (jo0) are sent on initial registration by the 'YC1jo0' command so that automation can get an accurate representation of the initial system state.

The example below is for an 8x8 shuffle matrix on voiceover 1, and assumes that 'advanced audio tallies' (YC1) has already been enabled. On registration, the initial state (one-to-one mapping) is bulk tallied. Then when a stereo-swap shuffle preset is loaded, 16 tallies are reported showing 8 crosspoint disconnections and 8 crosspoint connections:

```

YC1jo0100 # Enable shuffle crosspoint tallies (V01)
-> jo01000000001 # Input 1 connected to Output 1
-> jo01000001011 # Input 2 connected to Output 2
-> jo01000002021 # Input 3 connected to Output 3
-> jo01000003031 # Input 4 connected to Output 4
-> jo01000004041 # Input 5 connected to Output 5
-> jo01000005051 # Input 6 connected to Output 6
-> jo01000006061 # Input 7 connected to Output 7
-> jo01000007071 # Input 8 connected to Output 8

jo1100008x8_stereo_swap # Load 8x8 stereo swap preset
-> jo01000000000 # Input 1 disconnected from Output 1
-> jo01000001001 # Input 2 connected to Output 1
-> jo01000000011 # Input 1 connected to Output 2
-> jo01000001010 # Input 2 disconnected from Output 2
-> jo01000002020 # Input 3 disconnected from Output 3
-> jo01000003021 # Input 4 connected to Output 3

```

```

-> jo01000002031 # Input 3 connected to Output 4
-> jo01000003030 # Input 4 disconnected from Output 4
-> jo01000004040 # Input 5 disconnected from Output 5
-> jo01000005041 # Input 6 connected to Output 5
-> jo01000004051 # Input 5 connected to Output 6
-> jo01000005050 # Input 6 disconnected from Output 6
-> jo01000006060 # Input 7 disconnected from Output 7
-> jo01000007061 # Input 8 connected to Output 7
-> jo01000006071 # Input 7 connected to Output 8
-> jo01000007070 # Input 8 disconnected from Output 8
-> jo11000008x8_stereo_swap # Shuffle preset loaded

```

YC_jp0 Enable Audio Mute Tallies YC_jp0

This command enables audio mute (jp0) tallies within the advanced audio model. Tallies can be enabled for selected inputs, voiceovers and outputs, or all audio blocks that support gain control.

Note: Audio mute tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' has many inputs, voiceovers and outputs supporting gains.

Note: Audio mute tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire audio mute tallies' command.

CMD	Param_1	CMD	Param_2	Param_3
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jp0	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output (optional)	%02x: Bus index (See page 227) (optional)

When parameters 2 and 3 are defined, only the specified audio block will return audio mute tallies.

When parameter 3 is not defined, all audio blocks of the specified 'bus type' will return audio mute tallies.

If parameters 2 and 3 are not defined, all audio input/voiceover/output blocks with gain control will return audio mute tallies.

As automation and panels send 'set mute' (jp0) or 'load gain preset' (jp5) commands, audio mute (jp0) tally commands are sent back from the device for the enabled audio blocks. The format of each tally response exactly matches the 'jp0' set/enquire command.

Audio mute tallies (jp0) are sent on initial registration by the 'YC1jp0' command so that automation can get an accurate representation of the initial system state.

YC_jp1 Enable Audio Trim Tallies YC_jp1

This command enables audio trim (jp1) tallies within the advanced audio model. Tallies can be enabled for selected inputs, voiceovers and outputs, or all audio blocks that support gain control.

Note: Audio trim tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' has many inputs, voiceovers and outputs supporting gains.

Note: Audio trim tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire audio trim tallies' command.

CMD	Param_1	CMD	Param_2	Param_3
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jp1	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output (optional)	%02x: Bus index (See page 227) (optional)

When parameters 2 and 3 are defined, only the specified audio block will return audio trim tallies.

When parameter 3 is not defined, all audio blocks of the specified 'bus type' will return audio trim tallies.

If parameters 2 and 3 are not defined, all audio input/voiceover/output blocks with gain control will return audio trim tallies.

As automation and panels send 'set trim' (jp1) or 'load gain preset' (jp5) commands, audio trim (jp1) tally commands are sent back from the device for

the enabled audio blocks. The format of each tally response exactly matches the 'jp1' set/enquire command.

Audio trim tallies (jp1) are sent on initial registration by the 'YC1jp1' command so that automation can get an accurate representation of the initial system state.

YC_jp2 Enable Audio Invert Tallies YC_jp2

This command enables audio invert (jp2) tallies within the advanced audio model. Tallies can be enabled for selected inputs, voiceovers and outputs, or all audio blocks that support gain control.

Note: Audio invert tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' has many inputs, voiceovers and outputs supporting gains.

Note: Audio invert tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire audio invert tallies' command.

CMD	Param_1	CMD	Param_2	Param_3
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jp2	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output (optional)	%02x: Bus index (See page 227) (optional)

When parameters 2 and 3 are defined, only the specified audio block will return audio invert tallies.

When parameter 3 is not defined, all audio blocks of the specified 'bus type' will return audio invert tallies.

If parameters 2 and 3 are not defined, all audio input/voiceover/output blocks with gain control will return audio invert tallies.

As automation and panels send 'set invert' (jp2) or 'load gain preset' (jp5) commands, audio trim (jp2) tally commands are sent back from the device for the enabled audio blocks. The format of each tally response exactly matches the 'jp2' set/enquire command.

Audio invert tallies (jp2) are sent on initial registration by the 'YC1jp2' command so that automation can get an accurate representation of the initial system state.

YC_jp3 Enable Audio Level Tallies YC_jp3

This command enables audio level (jp3) tallies within the advanced audio model. Tallies can be enabled for selected inputs, voiceovers and outputs, or all audio blocks that support gain control.

Note: Audio level tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' has many inputs, voiceovers and outputs supporting gains.

Note: Audio level tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire audio level tallies' command.

CMD	Param_1	CMD	Param_2	Param_3
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jp3	%1x: Bus type 0x0 = Input stream 0x1 = Voiceover 0x2 = Output (optional)	%02x: Bus index (See page 227) (optional)

When parameters 2 and 3 are defined, only the specified audio block will return audio level tallies.

When parameter 3 is not defined, all audio blocks of the specified 'bus type' will return audio level tallies.

If parameters 2 and 3 are not defined, all audio input/voiceover/output blocks with gain control will return audio level tallies.

As automation and panels send 'set level' (jp3) or 'load gain preset' (jp5) commands, audio level (jp3) tally commands are sent back from the device for the enabled audio blocks. The format of each tally response exactly matches the 'jp3' set/enquire command.

Audio level tallies (jp3) are sent on initial registration by the 'YC1jp3' command so that automation can get an accurate representation of the initial system state.

YC_jq6 Enable Voice-over Fade Rate Tallies

YC_jq6

This command enables voice-over fade rate (jq6) tallies within the advanced audio model.

Note: Voice-over fade rate tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic when the 'Audio Graph' supports may voice-overs, with each having four different fade rates (automation, slow, medium and fast).

Note: Voice-over fade rate tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

Note: There is no 'enquire voice-over fade rate tallies' command.

CMD	Param_1	CMD
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	jq6

As automation and panels send 'set voice-over fade rate' (jq6) commands, voice-over fade rate (jq6) tally commands are sent back from the device for the enabled audio blocks. The format of each tally response exactly matches the 'jq6' set/enquire command.

Voice-over fade rate tallies (jq6) are sent on initial registration by the 'YC1jq6' command so that automation can get an accurate representation of the initial system state.

js Audio Meter Tally js

This tally is enabled with the 'Enable Audio Meter Tallies' (YC1js) command.

Once enabled, it returns metering information about the audio currently feeding the 'Audio Meters' block. The size of the tally depends on how many connections are made to input pins on the 'Audio Meters' block. The number of pins connected to the 'Audio Meters' block is determined with the 'jt0' command.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
js	%02x: Meter type 0x00 = Peak 0x01 = Phase 0x02 = Loudness 0x03 = VU	%c: Meter 1 0 - 255	%c%c%c ... Meter 2 – Meter N

YC_js Enable Audio Meter Tallies YC_js

This command enables audio meter (js) tallies within the advanced audio model. Tallies can be enabled for peak, phase, loudness and VU meters, individually or in combination.

Note:	Audio meter tallies are enabled separately from YC1 to allow a granular (or targeted) approach. This prevents excessive amounts of automation traffic at critical times such as when an automation system (or panel) is setting up a channel. Audio meter tallies can be disabled or “throttled back” until the control system is able to receive audio meters at the normal operating rate.
Note:	Audio meter tallies can only be enabled if advanced audio tallies (YC1) are enabled first.

CMD	Param_1	CMD	Param_2
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	js	%02x: Meter type 0x00 = Peak 0x01 = Phase 0x02 = Loudness 0x03 = VU

T

Param_3	Param_4
%02x: Send meters 0x00 = Never 0x01 = Once 0x02 = On change 0x03 = With frequency	%02: Frequency 0x00 – 0xFF

As audio feeding the 'Audio Meters' block changes, 'audio meters' (js) tallies are sent out using the specified meter type and frequency.

YC2js Enquire Audio Meter Tallies YC2js

This command enquires the status of advanced audio meter tallies for the specified meter type.

CMD	Param_1	CMD	Param_2
YC	%1x: Flag 0x2 = Enquire	js	%02x: Meter type 0x00 = Peak 0x01 = Phase 0x02 = Loudness 0x03 = VU

The format of the response command is:

CMD	Param_1	CMD	Param_2
YC	%1x: Tally enable 0x0 = No 0x1 = Yes	js	%02x: Meter type 0x00 = Peak 0x01 = Phase 0x02 = Loudness 0x03 = VU

Param_3	Param_4
%02x: Send meters 0x00 = Never 0x01 = Once 0x02 = On change 0x03 = With frequency	%02: Frequency 0x00 – 0xFF

Easyplay

Note: The Easyplay commands are only available if an Imagestore unit has the Easysound audio mixer and Easyplay options installed.

Commands and Responses

k0 Load Audio Clip (Stream 1) k0

This command loads (or pre-rolls) an audio clip into the selected queue of stream 1 in preparation for play out.

Note: Imagestore 750, LGK-3901 and ISM-3901 version 3.1 (and above) adds support for more than one Easyplay stream. They should use the 'ka' command instead of 'k0'.

CMD	Param_1	Param_2
k0	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5	%s: Audio file name

The number of queues is equal to the number of keying layers:

- Imagestore 2/2U/3 2 queues supported (0x0 - 0x1)
- Imagestore 300[+] 2 queues supported (0x0 - 0x1)
- Imagestore HD-TV 2 queues supported (0x0 - 0x1)
- Imagestore 750 4 queues supported (0x0 - 0x3)
- LGK-3901, ISM-3901 5 queues supported (0x0 - 0x4)

Please see page 38 for per-product layer descriptions

The reason for queues is to automatically associate audio clips to store media by filename. When a keying layer is cut up, the associated audio clip will play out automatically.

Example:

```
void load_audio(char * filename, int layer)
{
    char temp[100];
    put_remote_str(filename, temp);
    remote_send("k0%1x%s", layer, temp);
}
```

k1 Start Audio Playback (Stream 1) k1

This command starts playing out the audio clip loaded into the selected queue of stream 1.

Note: Imagestore 750, LGK-3901 and ISM-3901 version 3.1 (and above) adds support for more than one Easyplay stream. They should use the 'kb' command instead of 'k1'.

CMD	Param_1
k1	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

Note: Easyplay stream 1 has multiple queues that can be loaded concurrently. However, only one queue can ever play out at a time.

k2 Stop Audio Playback (Stream 1) k2

This command stops playing out the audio clip for the selected queue of stream 1.

Note: Imagestore 750, LGK-3901 and ISM-3901 version 3.1 (and above) adds support for more than one Easyplay stream. They should use the 'kc' command instead of 'k2'.

CMD	Param_1
k2	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

k3 Unload Audio Clip (Stream 1) k3

This command unloads an audio clip from the selected queue for stream 1. This prevents the audio clip from being played out automatically when its associated keyer is cut up.

Note: Imagestore 750, LGK-3901 and ISM-3901 version 3.1 (and above) adds support for more than one Easyplay stream. They should use the 'kd' command instead of 'k3'.

CMD	Param_1
k3	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

k4 Enquire Audio Clip (Stream 1) k4

This command enquires the filename of the audio clip loaded the selected queue for stream 1.

Note: Imagestore 750, LGK-3901 and ISM-3901 version 3.1 (and above) adds support for more than one Easyplay stream. They should use the 'ka' command instead of 'k4'.

CMD	Param_1
k4	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

The response has the following format:

CMD	Param_1
k4	%s: Audio file name

R8 Enquire Audio File Information R8

This command enquires for information on whether the specified audio clip file (WAV, OXE or OXW) exists on the device's internal disk.

Note: Use the R3 command to query the existence of image files.

Note: OXE files are not supported by Imagestore 2/2U/3.

CMD	Param_1
R8	%s: Filename

Example:

```
void EnquireAudioFile ()
{
    remote_send("R8%s", "song.wav");
}
```

The information returned is:

CMD	Param_1	Param_2
R8	%1x: File exists 0x0 = No 0x1 = Yes	%s: Filename

RB Get Audio File Duration RB

This enquire command is used to find the duration of an audio clip. This allows automation to determine how long an Easyplay file will play out for ahead of time so that the associated voiceover can be controlled accordingly. When an audio clip is set to loop, then the duration returned is for the first loop.

Audio clips should have a valid Easyplay file extension (.WAV, .OXW or .OXE) and can be any valid width (i.e. 2-channel stereo, 5.1 surround, 16-channel). Please check individual product documentation to see which file formats are widths supported.

CMD	Param_1
RB	%s: Filename

The information returned for a valid audio file is:

Format	Field	Description
%c%c	CMD	'RB'
%s	Filename	Audio clip filename
%02d	Hours	Number of hours
%02d	Minutes	Number of minutes
%02d	Seconds	Number of seconds
%02d	Frames	Number of frames

Easyplay 2

Easyplay 2 audio clip playout commands are only supported by Imagestore 750, LGK-3901 and ISM-3901 version 3.1 and above.

Note: DSK-3901 does not support Easyplay 2 audio clip playout.

These enhanced Easyplay commands add support for playing out multiple audio streams concurrently, which can be used for multiple-language applications.

Easyplay 2 also increases the total number of individual audio channels that can be played out. Please check individual product manuals for details.

The older Easyplay commands (k0, k1, k2, k3, k4) are still supported, however they are limited to control of just one stream. These commands are now superseded by 'ka', 'kb', 'kc' and 'kd'.

Streams and Queues

The number of streams supported by version 3.1 depends on the product:

- Imagestore 750 4 streams supported (0x00 - 0x03)
- LGK-3901, ISM-3901 2 streams supported (0x00 - 0x01)

The number of queues (per stream) is equal to the number of keying layers:

- Imagestore 750 4 queues supported (0x00 - 0x03)
- LGK-3901, ISM-3901 5 queues supported (0x00 - 0x04)

Please see page 38 for per-product layer descriptions

The reason for queues is to automatically associate audio clips to store media by filename. When a keying layer is cut up, the associated audio clip will play out automatically.

Although each stream has multiple queues that can be loaded concurrently, only one queue per stream can play out at a time. When one queue is playing out, starting a different queue on the same stream will automatically stop the first queue.

Commands and Responses

k6 Enquire Easyplay Stream Count k6

This command enquires the maximum possible number of Easyplay streams that can be supported by this device.

This does not give the actual number of streams defined within the 'Audio Graph' configuration. This has to be established using the 'ke' command.

CMD
k6

The response is formatted as follows:

CMD	Param_1
k6	%02x: Stream count 0x02 = LGK-3901, ISM-3901 0x04 = Imagestore 750

k7 Enquire Easyplay Queue Count k7

This command enquires the number of Easyplay queues that are supported by the selected stream.

This does not tell you whether the stream is active or not within the 'Audio Graph', which has to be established using the 'ke' command.

CMD	Param_1
k7	%02x: Stream

The response is formatted as follows::

CMD	Param_1	Param_2
k7	%02x: Stream	%02x: Queue count 0x04 = Imagestore 750 0x05 = LGK-3901, ISM-3901

k8 Set Easyplay Associated Voiceover k8

Note: This command is closely related to 'k9' and 'kf'.

This command sets the PGM voiceover or PST voiceover which is associated with an Easyplay stream.

Normally this association will automatically drive the associated voiceover up (or down) as the Easyplay stream plays (or stops). However, when the 'stream play follow mode' (k9) is set to one of the 'Associated VO' settings, it is the Easyplay stream/queue that is driven by the voiceover position.

CMD	Param_1	Param_2	Param_3
k8	%02x: Stream	%02x: Voiceover 0x00 – 0x07	%1x: Voiceover type 0x0 = PGM 0x1 = PST

k8 Enquire Easyplay Associated Voiceover k8

This command enquires the PGM voiceover or PST voiceover which is currently associated with an Easyplay stream.

CMD	Param_1
k8	%02x: Stream

The response is:

CMD	Param_1	Param_2	Param_3
k8	%02x: Stream	%02x: Voiceover 0x00 – 0x07	%1x: Voiceover type 0x0 = PGM 0x1 = PST

k9 Set Easyplay Play Follow Mode k9

Note: This command is closely related to 'k8' and 'kf'.

This command sets how Easyplay streams are driven based on other events.

CMD	Param_1	Param_2
k9	%02x: Stream	%02x: Play follow mode 0x00 = Off

	0x01 = PGM keyers 0x02 = PST keyers 0x03 = Associated voiceover queue 1 0x04 = Associated voiceover queue 2 0x05 = Associated voiceover queue 3 0x06 = Associated voiceover queue 4 0x07 = Associated voiceover queue 5
--	---

When set 'Off' the Easyplay stream is driven independently by automation. When set to 'PGM keyers' or 'PST keyers' the queue of the Easyplay stream that is played/stopped corresponds to a keying layer being cut up/down. The Easyplay stream may still be driven independently by automation, but it is best not to mix different schemes. Alternatively a queue of the Easyplay stream may follow the associated voiceover position, in which case the Easyplay stream no longer drives the associated voiceover. This is useful for control panels which give the user control of voiceovers.

k9 Enquire Easyplay Play Follow Mode k9

This command enquires how Easyplay streams are driven based on other events.

CMD	Param_1
k9	%02x: Stream

The response is:

CMD	Param_1	Param_2
k9	%02x: Stream	%02x: Play follow mode 0x00 = Off 0x01 = PGM keyers 0x02 = PST keyers 0x03 = Associated voiceover queue 1 0x04 = Associated voiceover queue 2 0x05 = Associated voiceover queue 3 0x06 = Associated voiceover queue 4 0x07 = Associated voiceover queue 5

ka Load Audio Clip ka

This command loads (or pre-rolls) an audio clip into the selected stream and queue in preparation for playing it out.

Note: This command supersedes the 'k0' command.

CMD	Param_1	Param_1	Param_2
ka	%02x: Stream	%02x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5	%s: Audio file name

ka Enquire Audio Clip ka

This command enquires the filename of the audio clip loaded the selected stream and queue.

Note: This command supersedes the 'k4' command.

CMD	Param_1	Param_2
ka	%02x: Stream	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

The response has the following format:

CMD	Param_1	Param_1	Param_2
ka	%02x: Stream	%02x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5	%s: Audio file name

kb Play Audio Clip kb

This command starts playing out the audio clip that is loaded into the selected stream and queue.

Note: This command supersedes the 'k1' command.

CMD	Param_1	Param_2
kb	%02x: Stream	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

Note: Each Easyplay stream has multiple queues that can be loaded concurrently. However, only one queue of each stream can ever play out at a time.

kc Stop Audio Clip kc

This command stops playing out the audio clip for the selected stream and queue.

Note: This command supersedes the 'k2' command.

CMD	Param_1	Param_2
kc	%02x: Stream	%02x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

kd Unload Audio Clip kd

This command unloads an audio clip from the selected stream and queue. This prevents the audio clip from being played out automatically when its associated keyer is cut up.

Note: This command supersedes the 'k3' command.

CMD	Param_1	Param_2
kd	%02x: Stream	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

ke Enquire Easyplay Stream Valid ke

This command enquires whether the selected Easyplay stream is configured as a block within the 'Audio Graph' of the Imagestore Configurator.

CMD	Param_1
ke	%02x: Stream

The response shows whether the Easyplay stream is present or not:

CMD	Param_1	Param_2
ke	%02x: Stream	%1x: Stream present 0x0 = No 0x1 = Yes

Note: This command should be used in conjunction with the 'k6' command. When enumerating streams, please note that there may be gaps depending on the 'Audio Graph' configuration. For example, only streams 1 and 4 may be valid.

kf Set Easyplay Load Follow Mode kf

Note: This command is closely related to 'k8' and 'k9'.

This command sets whether the selected Easyplay stream is configured for 'stream load follow mode'.

When enabled, this allows queues of the Easyplay stream to be loaded automatically with audio clips following the loading of video media files, provided that an audio clip with the same filename (different extension) exists. The queue that is loaded always matches the keying layer of the video media load.

When the video layer is cut up (on PGM or PVW), the associated audio clip can be played out automatically using the 'k9' command.

CMD	Param_1	Param_2
kf	%02x: Stream	%02x: Load-follow mode 0x00 = Off 0x01 = On (PGM keyers)

kf Enquire Easyplay Load Follow Mode kf

This command enquires whether the selected Easyplay stream is configured for 'stream load-follow mode'.

CMD	Param_1
kf	%02x: Stream

The response shows whether the load follow mode is enabled or not:

CMD	Param_1	Param_2
kf	%02x: Stream	%02x: Load-follow mode 0x00 = Off 0x01 = On (PGM keyers)

Unsolicited Tallies

k8 Easyplay Associated Voiceover Tally k8

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about changes to the PGM voiceover or PST voiceover which is currently associated with an Easyplay stream (k8).

On registration, tallies for all Easyplay streams are returned so that automation can record the initial association states.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
k8	%02x: Stream	%02x: Voiceover 0x00 – 0x07	%1x: Voiceover type 0x0 = PGM 0x1 = PST

k9 Easyplay Play Follow Mode Tally k9

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about changes to how Easyplay streams are driven based on other events (k9).

On registration, tallies for all Easyplay streams are returned so that automation can record the initial states.

The format of the tally is:

CMD	Param_1	Param_2
k9	%02x: Stream	%02x: Play follow mode 0x00 = Off 0x01 = PGM keyers 0x02 = PST keyers 0x03 = Associated voiceover queue 1 0x04 = Associated voiceover queue 2 0x05 = Associated voiceover queue 3 0x06 = Associated voiceover queue 4

	0x07 = Associated voiceover queue 5
--	-------------------------------------

ka Load Audio Clip Tally ka

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about loads (and unloads) of audio clips into streams and queues.

On registration, tallies for all Easyplay streams and queues are returned so that automation can record the initial load states. If no Easyplay file is loaded, then the filename will be blank.

The format of the tally is:

CMD	Param_1	Param_1	Param_2
ka	%02x: Stream	%02x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5	%s: Audio file name

kb Play Audio Clip Tally kb

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about the playing of audio clips which are loaded into streams and queues.

The format of the tally is:

CMD	Param_1	Param_2
kb	%02x: Stream	%1x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

Note: Each Easyplay stream has multiple queues that can be loaded concurrently. However, only one queue of each stream can ever play out at a time.

kc Stop Audio Clip Tally kc

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about the stopping of audio clips which are loaded into streams and queues.

The format of the tally is:

CMD	Param_1	Param_2
kc	%02x: Stream	%02x: Queue 0x0 = Associated with DSK1 0x1 = Associated with DSK2 0x2 = Associated with DSK3 0x3 = Associated with DSK4 0x4 = Associated with DSK5

kf Easyplay Load Follow Mode Tally kf

This tally is enabled with the 'Enable Easyplay Tallies' (YE1) command.

Once enabled, it returns information about changes to how Easyplay loads for a stream are configured to be driven based on other events (kf).

On registration, tallies for all Easyplay streams are returned so that automation can record the initial states.

The format of the tally is:

CMD	Param_1	Param_2
kf	%02x: Stream	%02x: Load-follow mode 0x00 = Off 0x01 = On (PGM keyers)

YE Enable Easyplay Tallies YE

This command enables or disables Easyplay tallies k8, k9, ka, kb, kc and kf.

CMD	Param_1
YUI	%1x: Tally enable 0x0 = No 0x1 = Yes

Example with one stream enabled (S = stream, Q = queue):

```

YE1
-> k800000      # S1 is associated with VO1-PGM
-> k90003      # S1/Q1 is driven by associated VO(1-PGM)
-> kf0000      # S1 loads are explicit
-> ka0000clip.wav # S1/Q1 is loaded with clip.wav
-> kc0000      # S1/Q1 is stopped
-> ka0001      # S1/Q2 is unloaded
-> kc0001      # S1/Q2 is stopped
-> ka0002      # S1/Q3 is unloaded
-> kc0002      # S1/Q3 is stopped
-> ka0003      # S1/Q4 is unloaded
-> kc0003      # S1/Q4 is stopped

```

YE Enquire Easyplay Tallies YE

This command enquires the status of the Easyplay (k8, k9, ka, kb, kc, kf) tallies:

CMD
YE

The format of the response command is:

CMD	Param_1
YE	%1x: Tally enable 0x0 = No 0x1 = Yes

Oxtel v16-001 PROVISIONAL

Dolby and Up-Mix

Note: Dolby and up-mix commands are applicable to Imagestore 750 v2.0 or above with the relevant Dolby or up-mix modules fitted. See the Imagestore 750 user manual for information.

Commands

q00 Set Up-Mix Mode q00

This command sets the operating mode of the specified up-mix module.

CMD	Param_1	Param_2
q00	%1x: Module 0x0 – 0x3	%1x: Mode 0x0 = Passthrough 0x1 = Up-mix 0x2 = From metadata 0x3 = AutoMAX™

q10 Set Dolby E Encode Mode q10

This command sets the operating mode of the specified Dolby E encoder module.

CMD	Param_1	Param_2
q10	%1x: Module 0x0 – 0x3	%1x: Mode 0x0 = Encode 0x1 = Passthrough 0x2 = From decoder status

q20 Set Dolby Digital (AC3) Encode Mode

q20

This command sets the operating mode of the specified Dolby Digital (AC3) encoder module.

CMD	Param_1	Param_2
q20	%1x: Module 0x0 – 0x3	%1x: Mode 0x0 = Encode 0x1 = Passthrough 0x2 = From decoder status

q30 Set Dolby Decode Mode

q30

This command sets the operating mode of the specified Dolby decoder module.

CMD	Param_1	Param_2
q30	%1x: Module 0x0 – 0x3	%1x: Mode 0x0 = Automatic 0x1 = Dolby E 0x2 = Dolby Digital (AC3) 0x3 = Mute 0x4 = Force backup

Metadata

Note: Metadata commands apply to Imagestore 750 v2.0 or above. See the Imagestore 750 user manual for more information.

The following metadata commands provide granular control over individual metadata parameters. However, it is often preferable to make use of predefined 'metadata presets' which have all parameters set up in advance.

Please see Imagestore 750 product documentation for more details on setting up metadata presets. You can use the 'oT' command to switch between different 'metadata presets'.

Commands and Responses (Bulk)

oT Load Metadata Preset oT

This command loads a metadata preset for the specified bank. This allows all metadata settings to be changed in bulk, provided that metadata presets have been set up in advance.

Note: Imagestore 750 product documentation gives details for configuring metadata presets.

CMD	Param_1	Param_2
oT	%1x: Bank	%s: Metadata preset name

oT Enquire Metadata Preset oT

This command enquires the metadata preset associated with the specified bank.

CMD	Param_1
oT	%1x: Bank

The response command format matches the corresponding load command.

Commands and Responses (Granular)

o0 Set Pitch Shift Code o0

This command sets the pitch shift code for the specified bank number.

This defines the amount of pitch shift between the original and current playback speeds of the audio frame. See the Dolby documentation for how to calculate the pitch shift code.

CMD	Param_1	Param_2	Param_3
o0	%1x: Bank	%02x: Flags 0x00 (unused)	%08x: Pitch shift code

o0 Enquire Pitch Shift Code o0

This command enquires for the current pitch shift code for the specified bank number.

CMD	Param_1
o0	%1x: Bank

The response command format matches the corresponding set command.

o1 Set Program Description o1

This command sets the program description for the specified bank and program using any ASCII characters between 0x20 and 0x7E. This command is not applicable to user metadata.

CMD	Param_1	Param_2	Param_3
o1	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%s: Program description

o1 Enquire Program Description o1

This command enquires a program description for the specified bank and program.

CMD	Param_1	Param_2
o1	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o2 Set Program Bitstream Mode o2

This command sets the program bitstream mode for the specified bank and program to indicate the program service type.

CMD	Param_1	Param_2	Param_3
o2	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02x: Bitstream mode 0x00 = Complete main 0x01 = Music and effects 0x02 = Visually impaired 0x03 = Hearing impaired 0x04 = Dialogue 0x05 = Commentary 0x06 = Emergency 0x07 = Voiceover 0x08 = Karaoke

o2 Enquire Program Bitstream Mode o2

This command enquires the program bitstream mode for the specified bank and program.

CMD	Param_1	Param_2
o2	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o3 Set Program Audio Coding Mode o3

This command sets the program audio coding mode for the specified bank and program to indicate the main service channels in use.

CMD	Param_1	Param_2	Param_3
o3	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02x: Audio coding mode 0x00 = Two independent channels 0x01 = Centre channel 0x02 = Left and right channels 0x03 = Left, centre and right channels 0x04 = Left, right and surround channels 0x05 = Left, centre, right and surround channels 0x06 = Left, right, surround left and surround right channels 0x07 = Left, centre, right, surround left and surround right channels

o3 Enquire Program Audio Coding Mode o3

This command enquires the program audio coding mode for the specified bank and program.

CMD	Param_1	Param_2
o3	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o4 Set Program Centre Down-Mix Level o4

This command sets the program nominal down-mix level of the centre channel compared to the left and right channels.

CMD	Param_1	Param_2	Param_3
o4	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Centre down-mix level
0x00 = -3dB
0x01 = -4.5dB
0x02 = -6dB

o4 Enquire Program Centre Down-Mix Level

o4

This command enquires the program nominal down-mix level of the centre channel compared to the left and right channels.

CMD	Param_1	Param_2
o4	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o5 Set Program Surround Down-Mix Level

o5

This command sets the program nominal down-mix level of the surround channels.

CMD	Param_1	Param_2	Param_3
o5	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Surround down-mix level
0x00 = -3dB
0x01 = -4.5dB
0x02 = -6dB

o5 Enquire Program Surround Down-Mix Level o5

This command enquires the program nominal down-mix level of the surround channels.

CMD	Param_1	Param_2
o5	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o6 Set Program Dolby Surround Mode o6

This command sets the program Dolby surround mode for the specified bank and program.

CMD	Param_1	Param_2	Param_3
o6	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Dolby Surround Mode 0x00 = Not indicated 0x01 = Dolby surround not encoded 0x02 = Dolby surrounded encoded

o6 Enquire Program Dolby Surround Mode o6

This command enquires the program Dolby surround mode for the specified bank and program.

CMD	Param_1	Param_2
o6	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o7 Set Program LFE Enable o7

This command sets the program Low Frequency Effects (LFE) enable for the specified bank and program.

CMD	Param_1	Param_2	Param_3
o7	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: LFE enable 0x0 = No 0x1 = Yes

o7 Enquire Program LFE Enable o7

This command enquires the program Low Frequency Effects (LFE) enable for the specified bank and program.

CMD	Param_1	Param_2
o7	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o8 Set Program Dialogue Normalisation o8

This command sets the program dialogue normalisation value for the specified bank and program. This compensates for dialogue of different volumes. The normalisation value is the mean level for the dialogue in a program relative to 0dB full-scale digital level.

CMD	Param_1	Param_2	Param_3
o8	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Dialogue normalisation 0x00 - 0x1F

o8 Enquire Program Dialogue Normalisation

o8

This command enquires the program dialogue normalisation value for the specified bank and program.

CMD	Param_1	Param_2
o8	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

o9 Set Program Production Mix Level

o9

This command sets the program production mix level for the specified bank and program.

CMD	Param_1	Param_2	Param_3
o9	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Mix level 0x00 - 0x1F

o9 Enquire Program Production Mix Level

o9

This command enquires the program production mix level for the specified bank and program.

CMD	Param_1	Param_2
o9	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oA Set Program Production Room Type

oA

This command sets the program production room type used for the final audio mixing.

CMD	Param_1	Param_2	Param_3
oA	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Room type 0x00 = Not Indicated 0x01 = Large, X curve monitor 0x02 = Small, flat monitor

oA Enquire Program Production Room Type

oA

This command enquires the program production room type used for the final audio mixing.

CMD	Param_1	Param_2
oA	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oB Set Program Copyright Flag

oB

This command sets the program copyright flag for the information contained in the bitstream.

CMD	Param_1	Param_2	Param_3
oB	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: Copyright flag 0x0 = Not copyright 0x1 = Copyright

oB Enquire Program Copyright Flag oB

This command enquires the program copyright flag for the information contained in the bitstream.

CMD	Param_1	Param_2
oB	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oC Set Program Original Bitstream Format oC

This command sets whether the program is an original bitstream, or a copy of another bitstream.

CMD	Param_1	Param_2	Param_3
oC	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: Bitstream flag 0x0 = Not original bitstream 0x1 = Original bitstream

oC Enquire Program Original Bitstream Format oC

This command enquires whether the program is an original bitstream, or a copy of another bitstream.

CMD	Param_1	Param_2
oC	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oD Set Program Down-Mix Mode oD

This command sets the type of stereo program down-mix mode preferred by the master engineer. Note that this may be used, overridden or ignored.

CMD	Param_1	Param_2	Param_3
oD	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Preferred down-mix mode 0x00 = Not indicated 0x01 = Left total/Right total (Lt/Rt) preferred 0x02 = Left only/Right only (Lo/Ro) preferred

oD Enquire Program Down-Mix Mode oD

This command enquires the type of stereo program down-mix mode preferred by the master engineer.

CMD	Param_1	Param_2
oD	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oE Set Program Down-Mix Lt/Rt Centre Mix Level oE

This command sets the nominal program down-mix level of the centre channels with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix.

CMD	Param_1	Param_2	Param_3
oE	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Centre mix level 0x00 = +3dB

0x01 = +1.5dB
0x02 = 0dB
0x03 = -1.5dB
0x04 = -3dB
0x05 = -4.5dB
0x06 = -6dB
0x07 = -infinity dB

oE Enquire Program Down-Mix Lt/Rt Centre Mix Level oE

This command enquires the nominal program down-mix level of the centre channels with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix.

CMD	Param_1	Param_2
oE	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oF Set Program Down-Mix Lt/Rt Surround Mix Level oF

This command sets the nominal program down-mix level of the surround channel with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix.

CMD	Param_1	Param_2	Param_3
oF	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Surround mix level
0x00 = -1.5dB
0x01 = -3dB
0x02 = -4.5dB
0x03 = -6dB
0x04 = -infinity dB

oF Enquire Program Down-Mix Lt/Rt Surround Mix Level oF

This command enquires the nominal program down-mix level of the surround channel with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix.

CMD	Param_1	Param_2
oF	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oG Set Program Down-Mix Lo/Ro Centre Mix oG

This command sets the program down-mix level of the centre channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix.

CMD	Param_1	Param_2	Param_3
oG	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Centre mix level 0x00 = -1.5dB 0x01 = -3dB 0x02 = -4.5dB 0x03 = -6dB 0x04 = -infinity dB

oG Enquire Program Down-Mix Lo/Ro Centre Mix oG

This command enquires the program down-mix level of the centre channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix.

CMD	Param_1	Param_2
oG	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oH Set Program Down-Mix Lo/Ro Surround Mix Level oH

This command sets the program down-mix level of the surround channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix.

CMD	Param_1	Param_2	Param_3
oH	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Centre mix level 0x00 = -1.5dB 0x01 = -3dB 0x02 = -4.5dB 0x03 = -6dB 0x04 = -infinity dB

oH Enquire Program Down-Mix Lo/Ro Surround Mix Level oH

This command enquires the program down-mix level of the surround channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix.

CMD	Param_1	Param_2
oH	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oI Set Program Dolby Surround EX™ Mode oI

This command (lower case “o” upper case “I”) sets whether the program has been executed in Dolby Surround EX™ mode. This command is only applicable to audio coding modes of 2/2 or 3/2.

CMD	Param_1	Param_2	Param_3
ol	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Dolby surround EX™ mode 0x00 = Not indicated 0x01 = Not encoded 0x02 = Encoded

ol Enquire Program Dolby Surround EX™ Mode ol

This command enquires whether the program has been executed in Dolby Surround EX™ mode. This command is only applicable to audio coding modes of 2/2 or 3/2.

CMD	Param_1	Param_2
ol	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oJ Set Program Dolby Headphone Mode oJ

This command sets whether the program has been Dolby headphone encoded. This command is only applicable to audio coding mode 2/0.

CMD	Param_1	Param_2	Param_3
oJ	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: Headphone mode 0x00 = Not indicated 0x01 = Not encoded 0x02 = Encoded

oJ Enquire Program Dolby Headphone Mode oJ

This command enquires whether the program has been Dolby headphone encoded. This command is only applicable to audio coding mode 2/0.

CMD	Param_1	Param_2
oJ	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oK Set Program A/D Converter Type oK

This command sets the program analogue-to-digital (A/D) converter type used to capture PCM audio.

CMD	Param_1	Param_2	Param_3
oK	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%02: A/D convertor type 0x00 = Standard (or unknown) 0x01 = HDCD

oK Enquire Program A/D Converter Type oK

This command enquires the program analogue-to-digital (A/D) converter type used to capture PCM audio.

CMD	Param_1	Param_2
oK	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oL Set Program High Pass Filter Enable oL

This command sets the program DC blocking High Pass Filter (HPF) enable for the main input channels of the Dolby Digital encoder.

CMD	Param_1	Param_2	Param_3
oL	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: HPF enable 0x0 = No 0x1 = Yes

oL Enquire Program High Pass Filter Enable oL

This command enquires the program DC blocking High Pass Filter (HPF) enable for the main input channels of the Dolby Digital encoder.

CMD	Param_1	Param_2
oL	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oM Set Program Low Pass Filter Enable oM

This command sets the program Low Pass Filter (LPF) enable for the main input channels of the Dolby Digital encoder.

CMD	Param_1	Param_2	Param_3
oM	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: LPF enable 0x0 = No 0x1 = Yes

oM Enquire Program Low Pass Filter Enable

oM

This command enquires the program Low Pass Filter (LPF) enable for the main input channels of the Dolby Digital encoder.

CMD	Param_1	Param_2
oM	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oN Set Program LFE LPF Enable

oN

This command sets the program Low Frequency Effects (LFE) Low Pass Filter (LPF) enable for the LFE channel input of the Dolby Digital encoder.

CMD	Param_1	Param_2	Param_3
oN	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: LFE LPE enable 0x0 = No 0x1 = Yes

oN Enquire Program LFE LPF Enable

oN

This command enquires the program Low Frequency Effects (LFE) Low Pass Filter (LPF) enable for the LFE channel input of the Dolby Digital encoder.

CMD	Param_1	Param_2
oN	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oO Set Program Surround 90 Degree Phase-Shift

oO

CMD	Param_1	Param_2	Param_3
-----	---------	---------	---------

oO	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)
----	-----------	--------------	------------------------------

Param_4
%1: 90 degrees phase shift enable 0x0 = No 0x1 = Yes

oO Enquire Program Surround 90 Degree Phase-Shift oO

CMD	Param_1	Param_2
oO	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oP Set Program Surround 3dB Attenuation oP

This command sets the program surround channel 3dB attenuation enable. This command is used to compensate surround levels prior to encoding.

CMD	Param_1	Param_2	Param_3
oP	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: 3dB attenuation enable 0x0 = No 0x1 = Yes

oP Enquire Program Surround 3dB Attenuation oP

This command enquires the program surround channel 3dB attenuation enable.

CMD	Param_1	Param_2
oP	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oQ Set Program RF Pre-emphasis oQ

Note: This command is deprecated and is unsupported from v3.1 software onwards.

This command sets the program radio frequency (RF) pre-emphasis for RF modulated Dolby Digital bitstreams.

CMD	Param_1	Param_2	Param_3
oQ	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4
%1: RF pre-emphasis enable 0x0 = No 0x1 = Yes

oQ Enquire Program RF Pre-emphasis oQ

Note: This command is deprecated and is unsupported from v3.1 software onwards.

This command enquires the program radio frequency (RF) pre-emphasis for RF modulated Dolby Digital bitstreams.

CMD	Param_1	Param_2
oQ	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oR Set Program Compression Profile/Word

oR

This command sets the program RF compression profile/word. A flag is set to indicate whether the compression is a profile or a word, and sets that actual profile or word.

CMD	Param_1	Param_2	Param_3
oR	%1x: Bank	%1x: Program	%02x: Flags 0x00 (unused)

Param_4	Param_5
%1: Format 0x0 = Profile 0x1 = Word	%02: Value <u>Compression profile</u> 0x00 = None 0x01 = Film-standard 0x02 = Film-light 0x03 = Music-standard 0x04 = Speech <u>Word</u> 0x00 - 0xFF

oR Enquire Program Compression Profile/Word

oR

This command enquires the program RF compression profile/word.

CMD	Param_1	Param_2
oR	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

oS Set Program Dynamic Range Value oS

This command sets the program dynamic range compression profile. The value depends on whether the word is a word or profile.

CMD	Param_1	Param_2	Param_3	Param_4
oS	%1x: Bank	%1x: Program	%02x: Word 0x00 - 0x07	%02x: Flags 0x00 (unused)

Param_5
%02: Value <u>Compression profile</u> 0x00 = None 0x01 = Film-standard 0x02 = Film-light 0x03 = Music-standard 0x04 = Speech <u>Word</u> 0x00 - 0xFF

oS Enquire Program Dynamic Range Value oS

This command enquire the program dynamic range compression profile. The value depends on whether the word is a word or profile.

CMD	Param_1	Param_2	Param_3
oS	%1x: Bank	%1x: Program	%02x: Word 0x00 - 0x07

The response command format matches the corresponding set command.

oU Set Program Dynamic Range Type oU

This command sets the program dynamic range type for the specified bank and program.

CMD	Param_1	Param_2	Param_3
oU	%1x: Bank	%1x: Program	%02x: Flags

			0x00 (unused)
--	--	--	------------------

Param_4
%01: Dynamic range type 0x0 = Profile 0x1 = Word

oU Enquire Program Dynamic Range Type

oU

This command enquires the program dynamic range type for the specified bank and program.

CMD	Param_1	Param_2
oU	%1x: Bank	%1x: Program

The response command format matches the corresponding set command.

Passive Mode

In 'passive mode' the device transmits unsolicited tallies whenever its internal states change. The amount of polling commands required by automation reduces since listening for tallies keeps automation's representation of the device status up-to-date without any time delays.

Essentially passive mode is a way of enabling several different tallies at the same time. Generally speaking it is better only to enable the tallies that you are interested and so use of passive mode is not recommended. It is mostly of use for Presmaster.

Unsolicited Tallies

Ya Enable Passive Mode Tallies Ya

This command enables or disable passive mode for this automation connection. When enabled, various video and audio tallies are enabled depending on the specified passive mode option.

If the old audio model (page 172) is being used by automation, then mode 0x1 should be used to give 'j7' audio tallies and 'Y8' meter tallies.

If the advanced audio model (page 226) is being used by automation, then mode 0x2 should be used to give 'YC1' 'individual' audio tallies, although normally this will be enabled independently without using 'Ya'.

CMD	Param_1
Ya	%1x: Passive mode <u>Imagestore 750, DSK-3901, LGK-3901, ISM-3901</u> 0x0 = Off 0x1 = Video (Y61) + audio (j7) + audio meters (Y8) 0x2 = Video (Y61) + audio (YC1) 0x3 = Video (Y61) + audio (YC1, j7) + audio meters (Y8) <u>Imagestore 2/2U/3, Imagestore 300[+], Imagestore HD-TV</u> 0x0 = Off 0x1 = On - (tallies as above)

Yb Set Passive Mode Transmission Interval

Yb

This command sets the interval (in milliseconds) that the device transmits tallies.

CMD	Param_1
Yb	%04x: Transmission interval (ms)

The default value is 500ms (0x1F4). Once changed this value is saved in the Imagestore state file for future use.

Note: This command is only valid for Imagestore 2/2U/3. Other Imagestore products always transmit state tallies every field (interlaced) or frame (progressive).

Ye Enquire Passive Mode

Ye

This command enquires what the current passive mode is.

CMD
Ye

The response has the same format as the Ya command:

CMD	Param_1
Ye	%1x: Passive mode <u>Imagestore 750, DSK-3901, LGK-3901, ISM-3901</u> 0x0 = Off 0x1 = Video (Y61) + audio (j7) + audio meters (Y8) 0x2 = Video (Y61) + audio (YC1) 0x3 = Video (Y61) + audio (YC1, j7) + audio meters (Y8) <u>Imagestore 2/2U/3, Imagestore 300[+], Imagestore HD-TV</u> 0x0 = Off 0x1 = On - (tallies as above)

System and Status

Commands and Responses

M Enquire System Status M

This command returns different information about the system.

It is of limited use, but can be used by automation to poll the system periodically to make sure that the Oxtel device is still alive.

CMD
M

Example:

```
void GetSystemStatus(void)
{
    remote_send(CmdBuffer, "M");
}
```

The information returned is:

Format	Field	Description
%c	CMD	'M'
%1d	System mode	Imagestore 2/2U/3: 0 = Keyer, 1 = Mixer Otherwise: 0 = Cascade, 1 = Swap Preview
%03x	Version high	First integer in the software version
%03x	Version low	Second integer in the software version
%1d	Video standard	0 = PAL, 1 = NTSC, 2 = 1080i@59.94Hz, 3 = 1080i@50Hz, 4 = 720p@59.94Hz, 5 = 720p@50Hz, 6 = 1080p@59.94Hz, 7 = 1080p@50Hz
%03x	Preview source	(Intuition: always 0)
%03x	Fade rate DSK1	
%03x	Fade rate DSK2	
%03x	FTB rate DSK1	
%03x	FTB rate DSK2	
%1x	System not	

accessed

Note: The 'Xb' command (see page 319) returns the full software version. This is preferred when it is supported.

N Enquire Video Layer Status N

This command returns status information about the video layers and A/B mixer.

CMD

N

Example:

```
void EnquireVideoLayerStatus()
{
    remote_send("N");
}
```

Information returned for Imagestores 2/2U/3, HD-TV and 300[+] is:

Format	Field
%c	CMD 'N'
%03x	DSK1 fader angle (0x000 - 0x200)
%03x	DSK1 FTB angle (0x000 - 0x200)
%03x	DSK2 fader angle (0x000 - 0x200)
%03x	DSK2 FTB angle (0x000 - 0x200)
%02x	A/B mixer transition type: 0x01 = V-Fade, 0x03 = X-Fade, 0x05 = Cut, 0x06 = Wipe-LR, 0x07 = Wipe-TB, 0x10 = Wipe-RL, 0x11 = Wipe-BT

The information returned for Imagestore 750 is:

Format	Field
%c	CMD 'N'
%03x	DSK1 fader angle (0x000 - 0x200)
%03x	DSK1 FTB angle (0x000 - 0x200)
%03x	DSK2 fader angle (0x000 - 0x200)
%03x	DSK2 FTB angle (0x000 - 0x200)
%02x	A/B mixer transition type: 0x01 = V-Fade, 0x02 = Fade-Cut, 0x03 = X-Fade,

	0x04 = Cut-Fade, 0x05 = Cut, 0x1A = U-fade 0x1B = V-fade preset 1, 0x1C = V-fade preset 2
%03x	DSK3 fader angle (0x000 - 0x200)
%03x	DSK3 FTB angle (0x000 - 0x200)
%03x	DSK4 fader angle (0x000 - 0x200)
%03x	DSK4 FTB angle (0x000 - 0x200)

The information returned for DSK-3901 is:

Format	Field
%c	CMD 'N'
%03x	DSK1 fader angle (0x000 - 0x200)
%03x	DSK1 FTB angle (0x000 - 0x200)
%03x	DSK2 fader angle (0x000 - 0x200)
%03x	DSK2 FTB angle (0x000 - 0x200)
%02x	A/B mixer transition type: 0x01 = V-Fade, 0x02 = Fade-Cut, 0x03 = X-Fade, 0x04 = Cut-Fade, 0x05 = Cut, 0x1A = U-fade 0x1B = V-fade preset 1, 0x1C = V-fade preset 2

The information returned for LGK-3901 is:

Format	Field
%c	CMD 'N'
%03x	DSK1 fader angle (0x000 - 0x200)
%03x	DSK1 FTB angle (0x000 - 0x200)
%03x	DSK2 fader angle (0x000 - 0x200)
%03x	DSK2 FTB angle (0x000 - 0x200)
%02x	A/B mixer transition type: 0x01 = V-Fade, 0x02 = Fade-Cut, 0x03 = X-Fade, 0x04 = Cut-Fade, 0x05 = Cut, 0x1A = U-fade 0x1B = V-fade preset 1, 0x1C = V-fade preset 2
%03x	DSK3 fader angle (0x000 - 0x200)
%03x	DSK3 FTB angle (0x000 - 0x200)
%03x	DSK4 fader angle (0x000 - 0x200)
%03x	DSK4 FTB angle (0x000 - 0x200)
%03x	DSK5 fader angle (0x000 - 0x200)
%03x	DSK5 FTB angle (0x000 - 0x200)

Intuition SD/HD[+] requires an extra “layer number” parameter and returns the status of a single layer.

CMD	Param_1
N	%x: Layer

Intuition SD/HD[+] example:

```
void EnquireVideoLayerStatus(int Layer)
{
    remote_send("N%x", Layer);
}
```

The information returned for Imagestore Intuition is:

Format	Field
%c	CMD 'N'
%03x	Layer fader angle (0x000 - 0x200)
%03x	Layer FTB angle (0x000 - 0x200)
%03x	0x000 (unused)
%03x	0x000 (unused)
%02x	0x00 (unused)

Fader Angle = 0x000 equates to “Keyer Off” or “FTB On”.

Fader Angle = 0x200 equates to “Keyer On” or “FTB Off”.

Note: Transition types supported are not consistent between different products, so please check individual user manuals for details.

Note: This command is of little use to automation because of the inconsistent responses given for different products. Where possible use layer-specific enquires or unsolicited tallies for determining keyer and A/B mixer states.

X1 Enquire Licence X1

This command returns whether a unit has the specified licence (or option) installed, and also its value when the option has different variants.

You can use this to detect if a device is fitted with options such as Easysound, A/B-Mix, DVE, etc. Please refer to individual product manuals to see which licence options are available.

CMD	Param_1
X1	%c%c%c%c: Licence

Example:

```
void EnquireEthernet()
{
    remote_send("X1ENET");
}
```

The information returned is:

CMD	Param_1	Param_2	Param_3
X1	%c%c%c%c:c: Licence	%1x: Exists 0x0 = No 0x1 = Yes	%04x: Value

Note: DSK-3901, LGK-3901 and ISM-3901 options cannot be enquired using the X1 command, and will return "X100000" to all enquiries. Use iControl to view the licences.

X2 Insert Log Message X2

This command adds a user-defined string into the device's log message file. This can be used by automation to tag specific events and is helpful when recreating and diagnosing faults.

CMD	Param_1
X2	%s: Message string

Example:

```
void InformStartup()
{
    remote_send("X2%s", "Automation System rebooted");
}
```

X3 Enquire Command Availability X3

Note: For Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 version 3.1.x software onwards, the X3 command returns true for any command that can be supported by the combined code base. LGK-3901 will therefore appear to support Dolby commands; but if you send one this will fail. In that sense X3 is of little use.

This command is used to determine if a particular Oxtel command is currently available on the device for automation to make use of.

Different products support different sets of automation commands (see the Command Validity Table on page 373). However, the availability of certain commands also depends on licence options and hardware being present. For example, if DVE option is not present or DVE hardware fails to initialise on start up then DVE commands will not be available. This can be determined using the X3 command.

CMD	Param_1
X3	%c: Command (1 byte) %c%c: Command (2 bytes) %c%c%c: Command (3 bytes)

The information returned is:

CMD	Param_1	Param_2
X3	%c_: Command (1 byte) + space %c%c: Command (2 bytes) %c%c%c: Command (3 bytes)	%1x: Supported 0x0 = No 0x1 = Yes

Note: There is only a space in the response when the command is one byte long.

X4 Set Device Bypass X4

This command is used to control the device's mechanical video bypass relay so that the A-input is passed directly to the PGM output. This causes all processing of video and audio which is normally performed by the Imagestore to be entirely bypassed.

CMD	Param_1
X4	%1x: Bypass Imagestore device 0x0 = No (Imagestore processes video/audio) 0x1 = Yes (passthrough of video/audio)

Note: The bypass relay is enabled automatically whenever the software stops, such as during a system restart. During normal operation this command is not recommended since it is likely to cause undesirable output.

X4 Enquire Device Bypass X4

This command enquires whether the device's mechanical video bypass relay has been enabled, or not.

CMD
X4

The response returned is:

CMD	Param_1
X4	%1x: Bypass Imagestore device 0x0 = No (Imagestore processes video/audio) 0x1 = Yes (passthrough of video/audio)

X5 Enquire Input Colour Field X5

This command returns RGB information for the input colour fields. Values returned are in the range 0x000000 to 0xFFFFFFFF.

CMD
X5

The information returned for Imagestore 300[+] and Imagestore HD-TV is:

Format	Field	Description
%c%c	CMD	X5
%1x	A source	0x0 = Use SDI 0x1 = Use colour field
%06x	A colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Fill-1 source	0x0 = Use SDI 0x1 = Use colour field
%06x	Fill-1 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Key-1 source	0x0 = Use SDI 0x1 = Use colour field
%06x	Key-1 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	B source	0x0 = Use SDI 0x1 = Use colour field
%06x	B colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Fill-2 source	0x0 = Use SDI

		0x1 = Use colour field
%06x	Fill-2 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Key-2 source	0x0 = Use SDI 0x1 = Use colour field
%06x	Key-2 colour	RGB values (0x000000 - 0xFFFFFFFF)

If there are no Fill-2/Key-2 inputs, then they are treated as colour-field disabled and the colour is returned as black.

For Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 the command is extended to include extra SDI inputs (Fill-3 and Key-3). Also there are three colour fields (shared, not dedicated) each of which may be associated with any physical input(s). The RGB information returned applies to the colour field currently associated with the SDI input (if any).

Format	Field	Description
%c%c	CMD	X5
%1x	A source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	A colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Fill-1 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	Fill-1 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Key-1 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	Key-1 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	B source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	B colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Fill-2 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	Fill-2 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	Key-2 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	Key-2 colour	RGB values (0x000000 - 0xFFFFFFFF)
%1x	C/Fill-3 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	C/Fill-3 colour	RGB values (0x000000 - 0xFFFFFFFF)

%1x	D/Key-3 source	0x0 = Use SDI or test pattern 0x1 = Use colour field
%06x	D/Key-3 colour	RGB values (0x000000 - 0xFFFFFFFF)

XA Enquire Slave Layer Status XA

This command returns overall layer status for the primary (Imagestore) and slave (Intuition) devices when a command forwarding configuration is used.

The Imagestore provides the first 2 keying layers and the Intuition provides the next 14 layers (8 for Intuition XG).

Note: This command is largely deprecated and has little meaning if the primary device has more than 2 keying layers. Automation should instead use Keyer Position Tallies (3) for obtaining layer status.

The Intuition layer status is based on commands that the Imagestore has forwarded to the Intuition. This means if the Intuition is controlled directly (either by automation or the front panel), the layer changes will not be reflected in this command.

CMD
XA

Example:

```
void EnquireExtLayerStatus(void)
{
    remote_send("XA");
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	XA
%1x	Layer state 0x0 = Down 0x1 = Up	Imagestore DSK1 (0x0)
%1x	Layer state	Imagestore DSK2 (0x1)
%1x	Layer state	Intuition Layer 1 (0x0)
%1x	Layer state	Intuition Layer 2 (0x1)
%1x	Layer state	Intuition Layer 3 (0x2)
%1x	Layer state	Intuition Layer 4 (0x3)

Format	Field	Description
%1x	Layer state	Intuition Layer 5 (0x4)
%1x	Layer state	Intuition Layer 6 (0x5)
%1x	Layer state	Intuition Layer 7 (0x6)
%1x	Layer state	Intuition Layer 8 (0x7)
%1x	Layer state	Intuition Layer 9 (0x8)
%1x	Layer state	Intuition Layer 10 (0x9)
%1x	Layer state	Intuition Layer 11 (0xA)
%1x	Layer state	Intuition Layer 12 (0xB)
%1x	Layer state	Intuition Layer 13 (0xC)
%1x	Layer state	Intuition Layer 14 (0xD)

Example of returned XA is: "XA0110000000000000"

Representing, the DSK2 of the Imagestore is cut up, and the first layer of the Intuition is cut up.

XC Enquire Serial Baud Rate XC

This command returns to current baud rate for the specified serial port.

CMD	Param_1
XC	%1x: Serial port number

Example:

```
void GetSerialBaudRate(int Port)
{
    remote_send("XC%1x", Port);
}
```

The response command has the following format:

CMD	Param_1	Param_2
XC	%1x: Serial port number	%d: Baud rate

XI Enquire Reference XI

This command enquires the current reference mode.

CMD
XI

The response command has the following format:

CMD	Param_1
XI	%1x: Reference 0x0 = Analogue (REF1) 0x1 = Analogue (REF2) 0x2 = SDI background 0x3 = Current video genlock standard

For Intuition[+] the response command has the following format:

CMD	Param_1
XI	%1x: Reference 0x3 = Video genlock is current standard 0x4 = Invalid reference signal 0x5 = Video genlock is valid but not the current standard

XJ Enquire Clock Offset XJ

This command enquires the current clock offset related to the specified reference mode.

CMD	Param_1
XJ	%1x: Reference 0x0 = Analogue (REF1) 0x1 = Analogue (REF2) 0x2 = SDI background 0x3 = Current video genlock standard

The response returned has the following format:

CMD	Param_1	Param_2
XJ	%1x: Reference 0x0 = Analogue (REF1)	%06x: Clock Offset (samples)

0x1 = Analogue (REF2)
0x2 = SDI background
0x3 = Current video genlock standard

Param_3	Param_4
%06x: Minimum accepted value (0xFFFFFFFF indicates no minimum)	%06x: Maximum accepted value (0xFFFFFFFF indicates no maximum)

XK Enquire Serial Number XK

This command enquires for the 8-digit serial number of a unit.

CMD
XK

The response returns the serial number as follows:

CMD	Param_1
XK	%08d: Serial number

The highest four digits of the serial number define the product code, and the lowest four digits define the unit identifier number.

The exception is Imagestore 750, DSK-3901, LGK-3901 and ISM-3901 where the four digit product code is followed by a zero then a three digit unit ID. The full serial number can be obtained using the XU command (see page 318).

Product codes are as follows:

- 0629 Imagestore 2
- 0675 Imagestore HD-TV
- 0730 Imagestore 3
- 0765 Imagestore 300
- 0781 Intuition SD
- 0811 Intuition HD
- 0814 Imagestore 300+ 1RU
- 0832 Imagestore 300+ 2RU
- 0838 Intuition+ SD
- 0839 Intuition+ HD
- 0872 Imagestore 750
- 0887 DSK-3901, LGK-3901, ISM-3901

XN Enquire Serial Protocol XN

This command enquires the current protocol for the specified serial port.

CMD	Param_1
XN	%1x: Serial port number

The serial protocol is returned in the response command:

CMD	Param_1	Param_2
XN	%1x: Serial port number	%1x: Protocol 0x0: Oxtel automation 0x1: Oxtel to PCS 0x2: Easysound Standalone 0x3: Oxtel Slave (i.e. Intuition) 0x4: Sage ENDEC Model 1822 0x5: TFT EAS 911T 0x6: RCP Classic 0x7: ST61XX Temperature Sensor 0x8: Presmaster Protocol

Example:

```
void GetSerialProtocol(int Port)
{
    remote_send("XN%1x", Port);
}
```

XO Enquire Serial Type XO

This command enquires the current serial port type (RS-232 or RS-422) for the specified serial port.

CMD	Param_1
XO	%1x: Serial port number

The serial port type is returned in the response command:

CMD	Param_1	Param_2
XO	%1x: Serial port number	%1x: Serial type 0x0: RS-232 0x1: RS-422

Example:

```
void GetSerialPortType(int Port)
{
    remote_send("XO%lx", Port);
}
```

XR Enquire Number of Fill/Key XR

This command enquires how many Fill/Key input pairs are supported by the device.

CMD
XR

The fill/key count is returned in the response command:

CMD	Param_1
XR	%02x: Fill/Key count

XS Enquire Preview Count XS

This command enquires how many preview-type outputs (i.e. preview, clean feed, monitor) are supported by the device.

CMD
XS

The preview count is returned in the response command:

CMD	Param_1
XS	%02x: Preview count

XU Enquire Extended Serial Number XU

This command enquires for the extended (fifteen digit) serial number of the device.

CMD
XU

The response includes the extended serial number:

CMD	Param_1
XU	%15s: Extended serial number

The extended serial number has the format 'PPPPEE-WWWWSSSS', where:

- P: Project number (4 digits) – (see command XK on page 316)
- E: Final assembly element number (2 digits)
- W: Works order number (5 digits)
- S: Actual serial number (3 digits)

XX Enquire Input Status XX

This command enquires the current input status for the selected input.

CMD	Param_1
XX	%1x: Input (see page 119)

The response command includes whether the input is locked or unlocked, and the video standard of the input source. See individual product manuals for the different video standards that are supported.

CMD	Param_1	Param_2	Param_3
XX	%1x: Input (see page 119)	%1x: Status 0x0 = Unlocked 0x1 = Locked	%1x: Video standard 0x0 = PAL (625) 0x1 = NTSC (525) 0x2 = 1080i 59.94Hz 0x3 = 1080i 50Hz 0x4 = 720p 59.94Hz 0x5 = 720p 50Hz 0x6 = 1080p 59.94Hz (level A) 0x7 = 1080p 50Hz (level A) 0x8 = 1080p 59.94Hz (level B) 0x9 = 1080p 50Hz (level B) 0xF = Unknown (no input)

Xb Enquire Full Version Number Xb

This command enquires the full software version number as a string.

CMD
Xb

The full software version is included in the response:

CMD	Param_1
Xb	%s: Software version

The response string includes:

- High version
- Low version
- Point numbers (if present)
- Build number (if supported this is after a dash “-”)
- Development state: “a” = alpha, “b” = beta, “rc” = release candidate, no letters = production release

This differs from the M command (see page 305) which returns only the high and low integers of the software version.

Xc Pass Close Caption Xc

This command enables or disables close caption data in the video signal.

When enabled, the close caption data is extracted at the input of the Imagestore device, and re-inserted at the output. This prevents images, animations, text or DVE from overwriting the close caption data when it is located within the active picture. This applies only to NTSC (525) where close captions are present on lines 20 and 21.

CMD	Param_1
Xc	%1x: Pass close caption 0x0 = No 0x1 = Yes

Example:

```
void PassCloseCaption(bool Enable)
{
    remote_send("Xc%1x", Enable);
}
```


Xc Enquire Close Caption Xc

This command enquires for the current pass close caption state.

CMD
Xc

The response command has the following format:

CMD	Param_1
Xc	%1x: Pass close caption 0x0 = No 0x1 = Yes

Xe Set Date and Time Xe

This command sets the date and time of the device.

CMD	Param_1	Param_2	Param_3	Param_4
Xe	%04d: Year	%02d: Month	%02d: Day	%02d: Hours
Param_5	Param_6			
%02d: Minutes	%02d: Seconds			

For example, 'Xe20110302230010' Sets the date to be 02-March-2011 and the time to be 23:00:10.

Note: When a valid time code is present or an NTP server connected, then the system time will follow this.

Xn Enquire Product Name Xn

This command enquires the full product name for the device.

It is predominantly used to distinguish between DSK-3901, LGK-3901 and ISM-3901 products which have the same product code.

CMD
Xn

The response command has the following format:

CMD	Param_1
Xn	%s: Product name "Imagestore 750", "DSK-3901", "LGK-3901", or "ISM-3901"

Xu Enquire AES Audio Signal Type Xu

This command enquires the audio signal type and locked status for a given AES pair. Please refer to individual product documentation for how many pairs of AES input channels are supported.

CMD	Param_1
Xu	%02x: AES input pair

The information returned in the response command is:

CMD	Param_1	Param_2
Xu	%02x: AES input pair	%1x: Audio type 0x0 = PCM audio 0x1 = Non-PCM audio 0x2 = Dolby E data 0x3 = Dolby Digital (AC3) data

Param_4
%1x: Status 0x0 = Unlocked 0x1 = Locked

Xv Enquire Embedded Audio Signal Type Xv

This command enquires the audio signal type and locked status for a given channel of an embedded audio feed.

Note that audio feeds may be assigned to any available SDI input. Please refer to individual product documentation for the number of feeds supported.

CMD	Param_1	Param_2
Xv	%02x: Audio feed 0x00 - 0x03	%02x: Channel 0x00 - 0x0F

The information returned in the response commands is:

CMD	Param_1	Param_2	Param_3
Xv	%02x: Audio feed 0x00 - 0x03	%02x: Channel 0x00 - 0x0F	%1x: Audio type 0x0 = PCM audio 0x1 = Non-PCM audio 0x2 = Dolby E data 0x3 = Dolby Digital (AC3) data

Param_4
%1x: Status 0x0 = Unlocked 0x1 = Locked

DSK-3901, LGK-3901 and ISM-3901 can not detect Dolby E or Dolby Digital (AC3) audio types. They will be shown as "non-PCM".

Xz Enquire URS Reference Xz

This command enquires whether the system reference will use the Densité reference if present.

CMD
Xz

The response command gives the current URS setting:

CMD	Param_1
Xz	%1x: Enable URS 0x0 = No

	0x1 = Yes
--	-----------

Oxtel v16-001 PROVISIONAL

Unsolicited Tallies

Y7 Wake-up Tally Y7

This tally is always enabled.

It returns information about the device on serial connections (not on network connections) during a restart.

The format of the tally is:

CMD	Param_1	Param_2
Y7	%04x: Serial Number	%08x: IP Address

Note: The Y7 command is normally only used by Presmaster.

Y7 Request Wake-up Packet Y7

This command enquires for a solicited wake-up packet, which is then returned by the device on the appropriate connection.

CMD
Y7

The response provides information about IP address and serial number of the unit:

CMD	Param_1	Param_2
Y7	%04x: Serial Number	%08x: IP Address

YD Heartbeat Tally YD

This tally is enabled with the 'Enable Heartbeat Tallies' (YD1) command.

Once enabled, it returns as a heartbeat with a predefined frequency. This helps automation to know that an Oxtel device is 'alive' or communicating correctly without having to explicitly poll the device periodically.

The format of the tally is:

CMD	Param_1	Param_2
YD	%1x: Enable tally	%04x: Frequency (frames)

	0x1 = Enable	0x0001 - 0xFFFF
--	--------------	-----------------

YD Set Heartbeat Tallies YD

This command enables heartbeat tallies for the connection on which the command was received.

It also sets the number of frames between each heartbeat tally that is sent.

CMD	Param_1	Param_2
YD	%1x: Enable tally 0x0 = No 0x1 = Yes	%04x: Frequency (frames) 0x0001 - 0xFFFF

YD Enquire Heartbeat Tallies YD

This command enquires the status of the heartbeat tally for the connection.

CMD
YD

The format of the command response is:

CMD	Param_1	Param_2
YD	%1x: Enable tally 0x0 = No 0x1 = Yes	%04x: Frequency (frames) 0x0001 - 0xFFFF

Health

Commands and Responses

X0 Enquire Temperature X0

This command returns information about the ambient temperature of the device. Please refer to individual product manuals for safe temperature limits.

CMD
X0

The information returned is:

CMD	Param_1
X0	%5f: Temperature (°C)

Note: This is a representative temperature for the device as a whole and is not accurate for all zones. The 'Xht' command offers advanced temperature status in some products.

X6 Enquire Voltages X6

This command returns information (space separated) about various voltages within the device, plus some other health information.

CMD
X6

The information returned by Imagestore 2/2U/3 and Imagestore 300[+] is:

Format	Field	Description
%c%c	CMD	X6
%05.1f	Temperature	Ambient in °C
%03.2f	-12V	
%03.2f	-5V	
%03.2f	+3.3V	
%03.2f	+5V	

%03.2f	+5V	Analogue supply
%03.2f	+12V	
%03.2f	PLL Voltage	

The information returned by Imagestore HD-TV is:

Format	Field	Description
%c%c	CMD	X6
%05.1f	Temperature	Ambient in °C
%03.2f	-12V	
%03.2f	+1.8V	
%03.2f	+3.3V	
%03.2f	+5V	
%03.2f	+2.5V	
%03.2f	+12V	
%03.2f	0.00	Unused field

The information returned by Imagestore 750 is:

Format	Field	Description
%c%c	CMD	X6
%05.1f	Temperature	Ambient in °C
%03.2f	12V PSU A	0.00 = PSU not fitted or failed 1.00 = PSU present and working
%03.2f	12V PSU B	0.00 = PSU not fitted or failed 1.00 = PSU present and working
%03.2f	12V	Worst case measurement
%03.2f	Video voltages	0.00 = Failed 1.00 = Video voltages all within tolerances
%03.2f	DVE voltages	0.00 = Failed 1.00 = DVE voltages all within tolerances
%03.2f	0.00	Unused field
%05.2f	Fan speeds	Worst case fan (1-5) speed in RPM

Voltage tolerances are +/-10% for the 12V supplies and +/-5% for all other voltages.

Note: DSK-3901, LGK-3901 and ISM-3901 do not support the X6 command. The 'Xhv' command should be used instead.

XV Enquire System Health XV

This command enquires for health status of an Imagestore 750. This is an extended version of the X6 command, but largely been deprecated by the Xh commands, which also gives allowable health ranges.

CMD
XV

The values included in the response are separated by spaces.

Format	Field	Description
%c%c	CMD	XV
%05.1f	Ambient temperature	Worst case temperature in °C
%05.1f	CPU temperature	Temperature in °C
%05.1f	KID temperature	Temperature in °C
%05.1f	CPU FPGA temperature	Temperature in °C
%05.1f	SID temperature	Temperature in °C
%05.1f	SOD temperature	Temperature in °C
%05.1f	DVE1 temperature	Temperature in °C (when fitted)
%05.1f	DVE2 temperature	Temperature in °C (when fitted)
%03.2f	12V PSU A	Voltage
%03.2f	12V PSU B	Voltage
%03.2f	5V	Voltage
%03.2f	3.3V	Voltage
%03.2f	2.5V	Voltage
%03.2f	SID 1.8V	Voltage
%03.2f	1.2V	Voltage
%03.2f	DVE 12V	Voltage (when fitted)
%03.2f	DVE 2.5V	Voltage (when fitted)
%03.2f	DVE 1.8V	Voltage (when fitted)
%03.2f	DVE 1.2V	Voltage (when fitted)
%1x	PSU A status	0x0 = Failed, 0x1 = OK
%1x	PSU A converted status	0x0 = Failed, 0x1 = OK
%1x	PSU B status	0x0 = Failed, 0x1 = OK
%1x	PSU B converted status	0x0 = Failed, 0x1 = OK
%05.2f	Fan 1 speed	In RPM
%05.2f	Fan 2 speed	In RPM
%05.2f	Fan 3 speed	In RPM

%05.2f	Fan 4 speed	In RPM
%05.2f	Fan 5 speed	In RPM

The XV command is related to the enquire voltages command (X6). Presmaster and IMM actually utilise X6 for sounding alarms, but XV should then be used to find the real reasons for the alarm.

Xh Enquire Extended Health Commands Xh

The following commands (Xhf, Xhp, Xht and Xhv) provide extended information about system health for:

- Fans
- Power supply units (PSU)
- Temperatures
- Voltages

When supported these commands should be used in preference to the other health commands listed above. However, please note that a periodic polling mechanism is still required to check values, which implies a fairly high bandwidth of automation commands.

A superior alternative is to utilise unsolicited health alarm traps via SNMP (Imagestore 750) or Densité protocol (DSK-3901, LGK-3901, ISM-3901). These give a much larger range of health information than is available via Oxtel automation including validity of SDI video sources, references, audio (embedded or AES), slave device state, router state, network links and disk usage. Please see individual product manuals for further information.

Xhf Enquire Extended Fan Health Xhf

This command enquires for extended health information for the system fans.

Firstly, automation must enquire to see how many different fans are supported by the device. Not only is the fan count dependent on the product type, but it is also dependent on the hardware options present.

CMD
Xhf

The response command returns the number of fans supported:

CMD	Param_1
Xhf	%02x: Fans count

Next automation must enumerate through each fan to ensure that values are within range of the thresholds.

CMD	Param_1
Xhf	%02x: Fan ID

The response command returns information which is space-separated. The 'expected' flag indicates whether the value is expected to be good. If not then there is no need to generate an alarm if the value is not between the 'low range' and 'high range' values.

CMD	Param_1	Param_2	Param_3
Xhf	%02x: Fan ID	%1x: Expected 0x0 = No 0x1 = Yes	%06.2f: Low range

Param_4	Param_5	Param_6
%06.2f: High range	%06.2f: Fan speed	%s: Description

Xhp Enquire Extended PSU Health Xhp

This command enquires for extended health information for the system power supply units (PSU).

Firstly, automation must enquire to see how many different PSUs are supported by the device. Not only is the PSU count dependent on the product type, but it is also dependent on the hardware options present.

CMD
Xhp

The response command returns the number of PSUs supported:

CMD	Param_1
Xhp	%02x: PSU count

Next automation must enumerate through each PSU to ensure that they are working correctly.

CMD	Param_1
Xhp	%02x: PSU ID

The response command returns information which is space-separated. The 'expected' flag indicates whether the value is expected to be good. If not then there is no need to generate an alarm.

CMD	Param_1	Param_2	Param_3	
Xhp	%02x: PSU ID	%1x: Expected 0x0 = No 0x1 = Yes	%06.2f: PSU good 0.00 = No 1.00 = Yes	

Param_4
%s: Description

Xht Enquire Extended Temperature Health Xht

This command enquires for extended health information for the system temperatures.

Firstly, automation must enquire to see how many different temperature readings are supported by the device. Not only is the temperature count dependent on the product type, but it is also dependent on the hardware options present.

CMD
Xht

The response command returns the number of temperatures supported:

CMD	Param_1
Xht	%02x: Temperature count

Next automation must enumerate through each temperature to ensure that values are within range of the thresholds.

CMD	Param_1
Xht	%02x: Temperature ID

The response command returns information which is space-separated. The 'expected' flag indicates whether the value is expected to be good. If not then

there is no need to generate an alarm if the value is not between the 'low range' and 'high range' values.

CMD	Param_1	Param_2	Param_3
Xht	%02x: Temperature ID	%1x: Expected 0x0 = No 0x1 = Yes	%06.2f: Low range

Param_4	Param_5	Param_6
%06.2f: High range	%06.2f: Temperature	%s: Description

Xhv Enquire Extended Voltage Health Xhv

This command enquires for extended health information for the system voltage.

Firstly, automation must enquire to see how many different voltages are supported by the device. Not only is the voltage count dependent on the product type, but it is also dependent on the hardware options present.

CMD
Xhv

The response command returns the number of voltages supported:

CMD	Param_1
Xhv	%02x: Voltage count

Next automation must enumerate through each voltage to ensure that values are within range of the thresholds.

CMD	Param_1
Xhv	%02x: Voltage ID

The response command returns information which is space-separated. The 'expected' flag indicates whether the value is expected to be good. If not then there is no need to generate an alarm if the value is not between the 'low range' and 'high range' values.

CMD	Param_1	Param_2	Param_3
Xhv	%02x: Voltage ID	%1x: Expected 0x0 = No 0x1 = Yes	%06.2f: Low range

Param_4	Param_5	Param_6
%06.2f: High range	%06.2f: Voltage	%s: Description

Oxtel v16-001 PROVISIONAL

Scheduler

Scheduled commands allow other automation commands to be executed at the specified time in hours, minutes, seconds and frames.

They are primarily used to achieve frame-accurate transitions or state changes when this can not be guaranteed by automation; for example under network control.

Note: If no timecode source (LTC, VITC) is present the scheduler may not operate correctly.

Commands can be scheduled up to 23 hours in the future, and midnight crossing is supported. However scheduled commands are normally issued just a few seconds ahead of when they are due to prevent command conflicts.

Scheduled commands are volatile and so will powering off or restarting the Imagestore software will clear the scheduled list.

Commands and Responses

i0 Add Scheduled Command i0

This command schedules an automation command at the specified time in hours, minutes, seconds and frames.

CMD	Param_1	Param_2
i0	%08d: Timecode (HHMMSSFF)	%s: Automation command

i1 Delete Scheduled Command i1

This command deletes all commands that were previously scheduled at the specified time in hours, minutes, seconds and frames.

CMD	Param_1
i1	%08d: Timecode (HHMMSSFF)

i2 Delete All Scheduled Commands i2

This command deletes all scheduled commands.

CMD
i2

i3 Enquire Next Scheduled Command i3

This command enquires the time of the next scheduled command, or the time remaining until the next command.

CMD	Param_1
i3	%1x: Flag 0x0 = Timecode of next scheduled event 0x1 = Time until the next scheduled event 0x2 = Current time

The response command has the following format:

CMD	Param_1	Param_2
i1	%1x: Flag	%08d: Timecode (HHMMSSFF)

General Purpose Interface

Commands and Responses

X7 Virtual GPI X7

This command triggers the GPI macro associated with a GPI number.

It is called a “virtual GPI” because it does not affect the physical GPI lines in any way. You can use the X9 command to set physical GPIs.

CMD	Param_1	Param_2
X7	%1x: GPI direction 0x0 = Off 0x1 = On	%02x: GPI number

X8 Enquire GPI Status X8

Note: For Imagestore 300[+] or Imagestore HD-TV units with GPI lines on video and audio cards, video GPIs are numbered 0 to 8 and the audio GPIs are numbered 9 to 17. For the Imagestore 750 only video GPIs are supported; numbered 0 to 15. For the DSK-3901, LGK-3901 and ISM-3901 only video GPIs are supported; numbered 0 to 7. Please check individual user manuals for details.

This command enquires for the status of a GPI line. The GPI number is the (zero-based) GPI line number.

CMD	Param_1	Param_2
X8	%1x: GPI type 0x0 = Video 0x1 = Audio (if supported)	%02x: GPI number

The information returned is:

CMD	Param_1	Param_2
X8	%02x: GPI number	%1x: GPI state 0x0 = GPI off (level high) 0x1 = GPI on (level low)

Example for Imagestore 300[+] and Imagestore HD-TV:

```

INT rem_remote_gpi_read(Premote_rem, INT8 cmd, char * str)
{
    BOOL video_gpi[MAX_GPI], audio_gpi[MAX_GPI];
    int result = -1, vid_aud = -1, number = -1;

    if (sscanf(str, "%1x%02x", &vid_aud, &number) == 2)
    {
        if (number < MAX_GPI)
        {
            get_gpi_state(video_gpi, audio_gpi);

            if (vid_aud == 0)
            {
                result = video_gpi[number];
            }
            else
            {
                result = (audio_fitted ? audio_gpi[number] : -1);
            }

            if (result != -1)
            {
                rem_send_status(rem, "X8%02x%1x", number, result);
            }
            else
            {
                // 99 indicates something has gone wrong
                rem_send_status(rem, "X8%02x%1x", 99, 99);
            }
        }
    }
}

```

X9 GPI Special X9

This command can perform two functions. One of the functions is to return a status command containing a bit mask of the GPI input status. The other function is to set the physical output status of a GPI.

CMD	Param_1	Param_2
X9	%1x: GPI function 0x0 = GPI off 0x1 = GPI on 0x2 = Status request	%02x: GPI number (Not required for status request)

Note: For Imagestore 300[+] or Imagestore HD-TV units with GPI lines on video and audio cards, video GPIs are numbered 0 to 8 and the audio GPIs are numbered 9 to 17. For the Imagestore 750 only video GPIs are supported; numbered 0 to 15. For the DSK-3901, LGK-3901 and ISM-3901 only video GPIs are supported; numbered 0 to 7. Please check individual user manuals for details.

If a 'status request' is used, the GPI status information for all GPI lines is returned in the form of a bitwise flag:

CMD	Param_1
X9	%06x: GPI status (bitwise) 0x000001 = GPI 0 On 0x000002 = GPI 1 On 0x000004 = GPI 2 On 0x000008 = GPI 3 On 0x000010 = GPI 4 On ... 0x008000 = GPI 15 On

XB GPI Output Status XB

This command sets the output state of the specified GPI number.

Note: For Imagestore 300[+] or Imagestore HD-TV units with GPI lines on video and audio cards, video GPIs are numbered 0 to 8 and the audio GPIs are numbered 9 to 17. For the Imagestore 750 only video GPIs are supported; numbered 0 to 15. For the DSK-3901, LGK-3901 and ISM-3901 only video GPIs are supported; numbered 0 to 7. Please check individual user manuals for details.

CMD	Param_1	Param_2
XB	%1x: GPI number	%1x: GPI state 0x0 = Off 0x1 = On

Example:

```
void GPIOutputStatus(int GPI, bool Status)
```

```
{  
    remote_send("XB%1x %1x", GPI, Status);  
}
```

Oxtel v16-001 PROVISIONAL

Macros

The macros commands allow up to 65536 different macros to be triggered by automation or panels. These macros are set up using the Imagestore Configurator (Macro Mode) and each includes a list of automation commands that will be executed when the macro is triggered.

Macro commands are not coupled to a particular GPI line as with 'GPI macros' but are instead triggered using automation commands. That said, a 'GPI macro' can include a 'trigger macro' (x0) command.

Commands and Responses

x0 Trigger Macro x0

This command triggers the specified macro. This then executes a predefined list of Oxtel automation commands.

CMD	Param_1
x0	%04x: Macro ID 0x0000 - 0xFFFF

Note: Macros are defined using 'Macro Mode' within the Imagestore Configurator. Delays are inserted between commands using the 'Xd' command. Please see individual product manuals for how to configure macros.

x1 Enquire Macro Count x1

This command enquires for the total number of macros defined on the system. Strictly speaking the 'count' value is the highest macro ID plus 1.

Note: There may be missing macros IDs below the 'count' value, and so valid macros have to be enumerated using the 'x2' or 'x3' commands.

CMD
x1

The response command returns the macro count:

CMD	Param_1
x1	%04x: Macro count

x2 Enquire Macro By ID x2

This command enquires for the 'macro name' of the specified 'macro ID'.

CMD	Param_1
x2	%04x: Macro ID 0x0000 - 0xFFFF

The response command returns the 'macro name'. If this is an empty string then the specified 'macro ID' is missing:

CMD	Param_1	Param_2
x2	%04x: Macro ID 0x0000 - 0xFFFF	%s: Macro name

x3 Enquire Macro By Index x3

This command enquires for the 'macro ID' and 'macro name' of the specified 'macro index' (a notional list including all valid macros ordered by increasing 'macro ID').

This command makes enumerating macros easier than with the 'x2' command because missing macro IDs are skipped. The 'x3' command should be called with increasing 'macro index' until the 'macro ID' returned is the 'macro count' minus 1.

CMD	Param_1
x3	%04x: Macro index 0x0000 - 0xFFFF

The response command returns the corresponding macro ID and its name:

CMD	Param_1	Param_2	Param_3
x2	%04x: Macro index 0x0000 - 0xFFFF	%04x: Macro ID 0x0000 - 0xFFFF	%s: Macro name

Note: The 'macro index' and 'macro ID' do not necessarily match.

Master Control

The sections listed below define the Oxtel automation commands which are designed for Imagestore Master Control broadcast systems that include a Miranda NV9000 router controller, and Miranda iMC-Panel for manual override of automation:

- Arm Take Commands (page 343)
- Routing Control Commands (page 351)

The commands apply to the following products when the Master Control option has been purchased:

- Imagestore 750 4.2
- ISM-3901 4.3

Note: These commands do not apply to Imagestore devices under the control of Presmaster/PCS with PresPanel or PresStation panels. Imagestore/PCS systems and Imagestore Master Control systems are incompatible with one another.

Arm and Take

The arm/disarm and take mechanism allows one or more commands to be primed, and then all commands executed together as a batch job when a 'take' command is issued by automation or the operator of an iMC-Panel.

The transition that follows the 'take' command can encompass multiple functions such as switching the source video, cutting up keying layers, playing DVE moves, audio voiceovers and macros.

A major benefit of arming commands in advance of the take is that the end of the overall transition can be previewed beforehand on the PVW output.

The following commands can be armed and disarmed:

- 1 Fade Keyer Up / Fade Keyer Down
- 3 Cut Keyer Up / Cut Keyer Down
- UA Asymmetric Transition
- jq4 Fade Voiceover
- jq5 Cut Voiceover
- S0 Start Animation
- S1 Stop Animation
- S4 Restart Animation
- Wb DVE Sequence Play

- x0 Macro Trigger
- X7 Virtual GP
- kb Easyplay Play
- kc Easyplay Stop

Oxtel v16-001 PROVISIONAL

Commands

p1 Arm p1

This command arms a command so that is executed when the next 'take' command is issued by automation or iMC-Panel.

Where different arm commands conflict with one another before a 'take' – (for example, arm DSK1 to cut up (p130 1) followed by arm DSK1 to fade up (p110 1)) – the last arm command always replaces the previous arm, which is implicitly disarmed.

CMD	Param_1
p1	Command to arm (%s)

The commands that can be armed are shown on page 343.

p0 Disarm p0

This command disarms a command that is currently armed so that it is not executed when the next 'take' command is issued by automation or iMC-Panel.

CMD	Param_1
p0	Command to disarm (%s)

The commands that can be disarmed are shown on page 343.

If no parameter is passed in the command then all currently armed commands are disarmed, and the next 'take' has no effect.

CMD
p0

p4 Take p4

This command causes any commands that are currently armed to be executed immediately.

Take commands from automation normally have no parameter. The resulting transitions always uses the “automation” transition rates configured for A/B mixing, keyer fades and voiceovers.

CMD
p4

Take commands from an iMC-Panel pass a single parameter with value 0x1. The resulting transitions use rates which depend on the current “channel transition mode”, which can be:

- Automation
- Slow
- Medium
- Fast

CMD	Param_1
p4	Take from (%1x) 0x0 = Automation 0x1 = Panel

Note: The “channel transition mode” only has meaning to iMC-Panels and so is not documented in this manual.

Unsolicited Tallies

Yip1 Arm Tally Yip1

This tally is enabled with the 'Enable Arm/Take Tallies' (Yi1) command.

Once enabled, it returns information whenever a valid arm command is received.

On registration, tallies for all existing 'arms' are returned so that automation can record the initial arm state.

The format of the tally is:

CMD	Param_1
Yip1	%s: Arm command

For example:

```
p110 1 // Arm DSK 1 to fade up
-> Yip110 1 // The tally echoes the arm
```

Yip0 Disarm Tally Yip0

This tally is enabled with the 'Enable Arm/Take Tallies' (Yi1) command.

Once enabled, it returns information whenever a valid disarm command is received.

The format of the tally is:

CMD	Param_1
Yip0	%s: Disarm command

For example:

```
p010 1 // Disarm DSK 1 to fade up
-> Yip010 1 // Tally echoes the arm
```

Note that disarms can occur as the side-effect of another action, such as a conflicting arm, or a take command:

```
p110 1          // Arm DSK 1 to fade up
-> Yip110 1     // Tally echoes the arm

p130 1          // Arm DSK 1 to cut up (conflicts with fade up)
-> Yip010 1    // Tally echoes implicit disarm of fade up
-> Yip130 1     // Tally echoes the second arm
```

Note: The unsolicited tally response includes an unnecessary “Yi” at the start of it, so does not exactly echo the arm/disarm/take command. This is contrary to the standard for most other unsolicited tallies.

Yip4 Take-Start Tally Yip4

This tally is enabled with the 'Enable Arm/Take Tallies' (Yi1) command.

Once enabled, it returns information whenever a valid take command is received and a take-transition starts.

Armed transitions may take some frames to complete if they include fades. This command therefore signifies that a take transition has just started. The 'take end' tally (Yip5) signifies when a transition ends.

The format of the tally is:

CMD
Yip4

Note: The unsolicited tally response includes an unnecessary “Yi” at the start of it, so does not exactly echo the arm/disarm/take command. This is contrary to the standard for most other unsolicited tallies.

Yip5 Take-End Tally Yip5

This tally is enabled with the 'Enable Arm/Take Tallies' (Yi1) command.

Once enabled, it returns information whenever a valid take-transition ends.

The format of the tally is:

CMD

Yip5

For example:

```
p110 1 // Arm DSK 1 to fade up
-> Yip110 1 // Tally echoes the arm

p1UA1 // Arm A/B mixer to transition to B
-> Yip1UA1 // Tally echoes the arm

p4 // Take
-> Yip0UA1 // Implicit disarm of A/B mixer due to take
-> Yip010 1 // Implicit disarm of DSK 1 due to take
-> Yip4 // Take start tally
-> Yip5 // Take end tally (follows longest transition)
```

Note: The unsolicited tally response includes an unnecessary “Yi” at the start of it, so does not exactly echo the arm/disarm/take command. This is contrary to the standard for most other unsolicited tallies.

Yi Enable Arm/Take Tallies Yi

This command enables or disables tallies of arm, disarm and take actions for the specified connection.

When tallies are enabled, arms, disarms and takes are echoed so that state can be shared between automation and any panels that own or share the channel.

CMD	Param_1
Yi	%1x: Enable arm/disarm/take tallies 0x0 = No 0x1 = Yes

Yi Enquire Arm/Take Tallies Yi

This command enquires whether arm/disarm/take tallies are currently enabled on this connection.

CMD

Yi

The response command shows the tally enable status:

CMD	Param_1
Yi	%1x: Enable arm/disarm/take tallies 0x0 = No 0x1 = Yes

Routing Control

Note: Please refer to individual product documentation for details on configuring router sources and destinations.

Source Type and Source Index

The following 'source types' are supported by the routing control commands:

- 0x0 = Channel source
- 0x1 = NV9000 source
- 0x2 = Source being fed by

The 'source index' has different meanings for each 'source type':

Channel sources and their IDs are defined within the Master Control configuration for each Imagestore channel. They typically represent "known" sources of video/audio from servers, tape machines, live feeds, satellite feeds, etc. These sources and IDs are often shared between different channels, but they "belong" to each channel.

NV9000 sources and their IDs are defined within the SE-Utilities Configurator for the NV9000 router controller. They are never used by automation and other source types are always preferred.

'Sources being fed by' indices are defined within the Imagestore Configurator's 'System Settings > Master Control > Router Sources'. They represent "known" router inputs either being fed by outputs of the Imagestore channel, or by special user-defined sources (such as a outputs from a backup channel or black/silence). They are often used by panels for monitoring purposes, and are usually specific to this Imagestore channel. They can take the following values:

- 0x00000000 = PGM Output
- 0x00000001 = PST Output
- 0x00000002 = CLN Output
- 0x00000003 = MON Output
- 0x00000004 = AUX
- 0x00000005 = BLACK/SILENCE
- 0x00000006 = USER_1
- 0x00000007 = USER_2
- 0x00000008 = USER_3
- 0x00000009 = USER_4
- ...
- 0x00000019 = USER_20

Destination Type and Destination Index

The following 'destination types' are supported by the routing control commands:

- 0x0 = Destination feeding bus
- 0x1 = NV9000 destination

The 'destination index' has different meanings for each 'destination type':

'Destination feeding bus' and their indices are defined within the Imagestore Configurator's 'System Settings > Master Control > Router Destinations'. They represent router destinations feeding the PGM and PST inputs of the Imagestore channel, and an AUX feed which is sent to a local monitor.

- 0x00000000 = PGM/A-Input
- 0x00000002 = PST/B-Input
- 0x00000004 = AUX

NV9000 destinations and their IDs are defined within the SE-Utilities Configurator for the NV9000 router controller. They are used by panels for monitoring purposes.

Commands and Responses

rs0 Make Source Selection rs0

This command instructs the Imagestore device to make a router switch to the specified destination from the specified source.

CMD	Param_1	Param_2
rs0	%1x: Source type 0x0 = Channel source 0x2 = Source being fed by	%08x: Source index (see page 351)

Param_3	Param_4
%1x: Destination type 0x0 = Destination feeding bus 0x1 = NV9000 destination	%08x: Destination index (see page 352)

rs1 Enquire Source Selection rs1

This command enquires which source is currently selected for the specified destination.

CMD	Param_1	Param_2
rs1	%1x: Destination type 0x0 = Destination feeding bus 0x1 = NV9000 destination	%08x: Destination index (see page 352)

The response command has the following format:

CMD	Param_1	Param_2
rs0	%1x: Source type 0x0 = Channel source 0x1 = NV9000 source 0x2 = Source being fed by	%08x: Source index (see page 351)

Param_3	Param_4
%1x: Destination type 0x0 = Destination feeding bus 0x1 = NV9000 destination	%08x: Destination index (see page 352)

rs2 Enquire Channel Source Count rs2

This command enquires how many channel sources can be enumerated by automation.

CMD
rs2

The response command includes the channel source count:

CMD	Param_1
rs2	%04x: Count

Channel source IDs may be numbered from zero upwards. The number of sources that can be enumerated will always equal the highest channel source ID plus 1. For example, when the defined channel source IDs range from 0-31 this gives a 'count' value of 32. Channel source IDs may be missing from the list. For example, if channel source IDs 0, 5, 18-19 are missing (i.e. channel source IDs 1-4, 6-17, 20-31 are defined), this 'count' value returned is still 32.

rs3 Enquire Source Information rs3

This command enquires for the name and status of the specified source.

CMD	Param_1	Param_2
rs3	%1x: Source type 0x0 = Channel source 0x2 = Source being fed by	%08x: Source index (see page 351)

The response command includes the following information:

CMD	Param_1	Param_2
rs3	%1x: Source type 0x0 = Channel source 0x2 = Source being fed by	%08x: Source index (see page 351)

Param_3	Param_4
%1x: Enabled 0x0 = No 0x1 = Yes	%s: Name

If a source is disabled (parameter 3 = 0x0), it should not be shown on panels since it may represent a deleted source.

It is also possible to specify up to 16 enquiries in a single automation command with the argument being the same format, but separated by a “|” character.

The response command will be the same as above with the parameters separated by “|” character as well. So an example would be:

```
rs3000000000
-> rs3000000000Dummy Src 1

rs3000000000|000000010
-> rs3000000000Dummy Src 1|0000000100Dummy Src 17
```

rm0 Enquire Salvo Count rm0

This command enquires how many salvos can be enumerated by automation.

Salvos and their IDs are defined within the Master Control configuration for each Imagestore channel. The numbering of salvo IDs follows the same rules as for channel sources.

CMD
rm0

The response command includes the salvo count:

CMD	Param_1
rm0	%08x: Count

rm1 Enquire Salvo Information rm1

This command enquires for the name and status of the specified salvo.

CMD	Param_1
rm1	%08x: Salvo ID

The response command includes the following information:

CMD	Param_1	Param_2	Param_3
rm1	%08x: Salvo ID 0x0 = Channel source 0x2 = Source being fed by	%1x: Enabled 0x0 = No 0x1 = Yes	%s: Name

If a salvo is disabled (parameter 3 = 0x0), it should not be shown on panels since it may represent a deleted source.

rm2 Execute Salvo rm2

This command executes the specified salvo.

CMD	Param_1
rm2	%08x: Salvo ID

Unsolicited Tallies

YF Enable Router Tallies YF

This command enables various tallies (rs0, rs3, rm1, rm2) related to router switches and state.

Once enabled, it returns information whenever a source selection changes (rs0), source information changes (rs3), salvo information changes (rm1) or a salvo is fired (rm2).

The format of the command is:

CMD	Param_1
YF	%1x: Enable tallies 0x0 = No 0x1 = Yes

YF Enquire Router Tallies YF

This command enquires whether router tallies are currently enabled, or not.

CMD
YF

The response command has the format:

CMD	Param_1
YF	%1x: Enable tallies 0x0 = No 0x1 = Yes

rs0 Source Selection Tally rs0

This tally is enabled with the 'Enable Router Tallies' (YF1) command.

Once enabled, it returns information about router switches as they occur. This is limited to destinations which feed a specified Imagestore bus to avoid excessive amounts of unwanted router switch tallies.

On registration, tallies for all destinations (feeding buses) known by the configuration are returned so that automation can record the initial router state.

The format of the tally is:

CMD	Param_1	Param_2
rs0	%1x: Source type 0x0 = Channel source 0x1 = NV9000 source 0x2 = Source being fed by	%08x: Source index (see page 351)

Param_3	Param_4
%1x: Destination type 0x0 = Destination feeding bus	%08x: Destination index (see page 352)

rs3 Source Information Tally rs3

This tally is enabled with the 'Enable Router Tallies' (YF1) command.

Once enabled, it returns information about changes to the name and status of sources as they occur.

On registration, tallies for all sources known by the configuration are returned so that automation can record the initial router state.

The format of the tally is:

CMD	Param_1	Param_2
rs3	%1x: Source type 0x0 = Channel source 0x2 = Source being fed by	%08x: Source index (see page 351)

Param_3	Param_4
%1x: Enabled 0x0 = No 0x1 = Yes	%s: Name

rm1 Salvo Information Tally rm1

This tally is enabled with the 'Enable Router Tallies' (YF1) command.

Once enabled, it returns information about changes to the name and status of salvos as they occur.

On registration, tallies for all salvos known by the configuration are returned so that automation can record the initial router state.

The format of the tally is:

CMD	Param_1	Param_2	Param_3
rm1	%08x: Salvo ID 0x0 = Channel source 0x2 = Source being fed by	%1x: Enabled 0x0 = No 0x1 = Yes	%s: Name

rm2 Salvo Tally rm2

This tally is enabled with the 'Enable Router Tallies' (YF1) command.

Once enabled, it returns information about the execution of salvos as they occur.

The format of the tally is:

CMD	Param_1
rm2	%08x: Salvo ID

Intuition XG

The following commands only apply to the Intuition XG.

Note: A space separates the command from the first parameter. The “|” character is used to separate each parameter from the next parameter in the command.

Commands and Responses

Cue V0

This command cues a template onto an Intuition XG keyer.

```
V0 %s|%x|%s[|%s,%s,...]
```

CMD	Param_1	Param_2
V0	%s Keyer number, or name in Xplay	%x Template ID type 0 = Recall ID 1 = Name 2 = Embedded page

Param_3	Param_4
%s Template ID	%s,%s,... (optional) Cue parameters

The first parameter is the keyer to cue on. The parameter may be the keyer number, or the keyer name in Xplay.

The second parameter is the type of the ID used to identify the template to cue.

The third parameter is the ID identifying the template. Since this parameter may be xml, the “:” and other Oxtel Protocol reserved characters (see page 36) must be escaped. Also since the xml may have international characters in it, the parameter must be encoded in UTF-8.

The fourth parameter is an optional comma limited delimited list of cue parameters (available in 4.6 and up only). The parameters must be encoded in UTF-8. Since commas are separators, commas contained within parameters, that are not within quotes, must be escaped by replacing the

comma with its escaped version: "\2C". Comma parameters are referenced in the template actions as Input:P1, Input:P2 and so on. Parameters may be valid expressions currently supported in the VertigoXmedia suite however Data: variables are not supported.

Note that in the case of names, the name may be a "short" name, that is the name of the template or page itself without any category information. Additionally a name may be specified as a fully qualified name, such as "XMS:Pages.Demo:PG_Lower3rd". Note that since ':' is a reserved character, the name would have to be written using escaped characters as follows:

```
V0 1|1|XMS\3APages.Demo\3APG_Lower3rd
```

Note that the tallies should include both short and full names.

If the template ID is specified 'Embedded Page', the embedded page xml must be specified. This will contain characters that need to be escaped.

You can specify a page by name, but a page is not necessarily an embedded page. If using regular shared pages the Intuition XG V commands should be just:

- 1) Regular Page Comamnds: 'V0 0|1|PageName' (this page is called a page and lives on XMS). This is more common.
- 2) Embedded Page Commands: 'V0 0|2|<Embedded Page XML>'. This page is based on a template that lives in XMS but the page itself is specified through xml and basically applies an override to the template. For instance IF I have 1 template inside of XMS with a full screen image. The embedded page xml may specify a difference image be applied to the page.

Take V1

This command causes items cued on Intuition XG keyers to be taken.

```
V1 %s,%s,...[|%x]
```

CMD	Param_1	Param_2
V1	%s,%s,... Keyer numbers or names in Xplay Comma-separated	%x (optional) Take number -1 = Do all takes (default) 0 = Do first take 1 = Do second take ...

The first parameter is a comma separated list of keyer numbers or keyer names to take.

The second parameter is the take number. This parameter is optional. The default is -1.

Clear V2

This command causes items that are on-air on Intuition XG keyers to be cleared.

V2 %s,%s,... [|%x]

CMD	Param_1	Param_2
V2	%s,%s,... Keyer numbers or names in Xplay Comma-separated	%x (optional) Output to clear 0 = Preview 1 = On air (default)

The first parameter is a comma separated list of keyer numbers or keyer names to clear.

The second parameter is the output to clear. This parameter is optional. The default is 1.

Keyer On V3

This command cuts or fades up Intuition XG keyers.

V3 %s,%s,... |%x [|%x]

CMD	Param_1	Param_2	Param_3
V3	%s,%s,... Keyer numbers or names in Xplay Comma-separated	%x Fade rate in mS 0 = Cut	%x (optional) Output 0 = Preview 1 = On air (default)

The first parameter is a comma separated list of keyer numbers or keyer names to turn the keyer on.

The second parameter is the duration it will take to fade the keyer up. A duration of 0 means cut. The parameter is in milliseconds.

The third parameter specifies the output. This parameter is optional.

Keyer Off V4

This command cuts or fades down Intuition XG keys.

V4 %s,%s,... |%x [|%x]

CMD	Param_1	Param_2	Param_3
V4	%s,%s,... Keyer numbers or names in Xplay Comma-separated	%x Fade rate in mS 0 = Cut	%x (optional) Output 0 = Preview 1 = On air (default)

The first parameter is a comma separated list of keyer numbers or keyer names to turn the keyer off.

The second parameter is the duration it will take to fade the keyer down. A duration of 0 means cut. The parameter is in milliseconds.

The third parameter specifies the output. This parameter is optional.

Set Property V5

This command sets properties on Intuition XG. For example the text or image to be used within one or more template on one or more keyer can be set. The property can be set to be changed on the preview output or on air. The default is for the property to change to be seen on air.

V5 %s,%s,... |%s,%s,... |%s,%s,... [|%x] [|%s]

CMD	Param_1	Param_2
V5	%s,%s,... Objects and properties Comma-separated	%s,%s,... Values Comma-separated

Param_3	Param_4
%s,%s,... Keyer numbers or names in Xplay	%x (optional) Output 0 – Preview 1 – On Air (default)

The first parameter is a comma delimited list of objects and properties. The format is ObjectName.PropertyName. So an example would be:

```
Textbox1.Text, Image1.ImageName
```

The second parameter is a comma delimited list of values. For example:

```
"Hello World!","XMS:Images.Headshots:Obama"
```

The third parameter is a comma delimited list of keyers.

The fourth parameter specifies whether to target the preview or on-air buffer. This parameter is optional and by default is 1.

The fifth parameter specifies the output. This parameter is optional and is currently not used.

Multiple properties can be set by separating each 'Objects and Properties', 'Values' with the ',' character. The 'Objects and Properties' to be set are a delimited list in the form "ObjectName.PropertyName". The 'Values' indicate, for example, the text or image to be used for that object and name. The keyers can be identified using the keyer number, or the keyer name used in Xplay. If a coma ',' is needed within a string parameter it can be escaped as "\2C". For example:

To set the following on keyers 1 and 2, with output = 'On Air'

- Object: Textbox1
- Property: Text
- Value: Hello World!

- Object: Image1
- Property: ImageName
- Value: XMS:Images.Headshots:Obama

```
V5 Textbox1.Text,Image1.ImageName|Hello World!,
XMS\3AImages.Headshots\3AObama|1,2|1
```

Cue Clip V9

This command cues a clip onto an Intuition XG keyer.

```
V9 %s|%x|%s|[%s]|[%s]|[%s]|
```

CMD	Param_1	Param_2	Param_3
V9	%s Keyer number, or name in Xplay	%x Template ID type 0 = Recall ID 1 = Name	%s Clip ID

Param_4	Param_5	Param_6
%s (optional)	%s (optional)	%s,%s,... (optional)

Mark in value	Mark out value	Cue parameters
---------------	----------------	----------------

The first parameter is the keyer to cue on. The parameter may be the keyer number or the keyer name in Xplay.

The second parameter is the type of the ID used to identify the clip to cue.

The third parameter is the ID identifying the clip. The "." and other Oxtel Protocol reserved characters must be escaped. Also the parameter must be encoded in UTF-8.

The fourth parameter is optional and specifies the 'mark in' value in timecode without punctuation (e.g. 4:26 would be 426).

The fifth parameter is optional and specifies the 'mark out' value in timecode without punctuation (e.g. 1:10:10 would be 11010).

The sixth parameter is an optional comma limited delimited list of cue parameters. The parameters must be encoded in UTF-8. Since commas are separators, commas contained within parameters, that are not within quotes, must be escaped by replacing the comma with the escaped version: '\2C'. Comma parameters are referenced in the template actions as Input:P1, Input:P2 and so on. Parameters may be valid expressions currently supported in the VertigoXmedia suite however Data: variables are not supported.

Note that in the case of names, the name may be a "short" name, that is the name of the template or page itself without any category information. Additionally a name may be specified as a fully qualified name, such as "XMS:Clips.Demo:Promo1". Note that since ':' is a reserved character, the name would have to be written using escaped characters as follows:

```
XMS\3A Clips.Demo\3A Promo1
```

The tallies will look the same as the Cue (V0) tallies except they will include the 'mark in' and/or 'mark out' values if they were specified in the original command. Here are some examples showing a command followed by a tally:

```
V9 0|1|ColorBars|426
V0 0|RecallId=100,ShortName=ColorBars.mpeg,FullName=XMS\3A Clips\3A ColorBars.mpeg|markIn=00000426
```

```
V9 0|1|ColorBars|426
V0 0|RecallId=100,ShortName=ColorBars.mpeg,FullName=XMS\3A Clips\3A ColorBars.mpeg|markIn=00000426
```

```
V9 0|1|ColorBars||1010
V0 0|RecallId=100,ShortName=ColorBars.mpeg,FullName=XMS\3A Clips\3A ColorBars.mpeg|markIn=00000426
```

```
S\3aClips\3aColorBars.mpeg|markOut=00001010
```

```
V9 0|1|ColorBars|426|1010
```

```
V0 0|RecallId=100,ShortName=ColorBars.mpeg,FullName=XM
S\3aClips\3aColorBars.mpeg|markIn=00000426,markOut=000
01010
```

```
V9 0|1|ColorBars|426|1010|A,B,C
```

```
V9 0|RecallId=100,ShortName=ColorBars.mpeg,FullName=XM
S\3aClips\3aColorBars.mpeg|markIn=00000426,markOut=000
01010|Params=A,B,C
```

Enable External Command Tallies **Va**

This command enabled external command tallies.

Va %x

CMD	Param_1
Va	%x Enable tallies 0 = No 1 = Yes

The first parameter sets whether external command tallies are disabled (0) or enabled (1).

An external command is a request to execute an Oxtel protocol command and is generated by a device that has an associated external keyer. This command is reported back to the application and is forwarded to the associated external keyer device if there is one. If external command tallies are enabled, then the command is also tallied back.

Tallies received will be of the form: Va %s - The parameter is the command text as specified in the animation. If the command text contains multiple commands they are separated by a ';' character.

Unsolicited Tallies

All Vertigo tallies are enabled using the V8 command. They report the system state back to automation as it changes.

Enable Vertigo Tallies V8

This command enables or disables Vertigo (V) tallies. These are specified on page 366.

V8 %d

CMD	Param_1
V8	%d Enable tallies 0 = No 1 = Yes

The first parameter disables or enables Vertigo tallies.

If 'V8 1' is sent to enable tallies, then the tallies returned from Xplay are different from those returned when Y61 (enable video tallies) is used. The Vertigo tallies monitor:

- The state of each keyer
- The item that is cued on each keyer
- The item that is on air on each keyer

Cue Tally V0

This tally is reported when an item is cued.

V0 %x|%s[|%s,%s,...]

CMD	Param_1	Param_2	Param_3
V0	%x Keyer number	%s Item specification (see page 370)	%s,%s,... Cue parameters

The first parameter specifies the keyer on which the item was cued.

The second parameter is the specification of the item that was cued.

The third parameter is the parameters that the were specified when the item was cued. If no parameters are specified on the cue then none will be present in the tally.

For example if "purple" is cued, you will get the following tally returned:

```
V0 1|RecallId=135,ShortName=purple.tem,FullName=XMS\3aTemplat
es.Naomi\3apurple.tem
```

If the recall id field was blank as is the default in XMS then the tally would look like this:

```
V0 1|RecallId=,ShortName=purple.tem,FullName=XMS\3aTemplates.Naomi\3apurple.tem
```

If not using embedded pages, you can cue using the fully qualified name or the short name. In the tally response both names will be returned as is shown in the examples above. Both the fully qualified name and the short name are present in the tally.

The Embedded page is a special case where we specify the xml of a page the we want to cue. In this case the tally will simply specify the embedded page XML. For example:

```
V0 1|Embedded=<Page assetID="9722728b-add4-416e-a619-1e64d45a a246" name="PG_CROSS" templateID="6617c1c7-742e-493f-b74f-e74 8e660991e" templateCatID="445bb9b3-27de-4702-8c3c-fc66cf73ded 9" template="TPL_002" version="1.4" description="" categoryID=""><Description></Description><CreationDate></CreationDate>< ModificationDate></ModificationDate><Author></Author><RecallID></RecallID><TargetDeviceType>D2</TargetDeviceType><Duration >0</Duration><Override><Textbox id="1" name="Textbox1"><Property name="Text">CROSS</Property></Textbox><Texture id="2" name="Texture1"><Property name="ImageLocation">XMS\3a</Property><Property name="ImageCategory">a49d2979-a930-4096-98c9-a127adf07760</Property><Property name="ImageName">6fba512e-9049-4b01-a4cd-147c61327595</Property><Property name="AutoCreateJob">False</Property></Texture></Override><Assets><Asset type="Image" source="Model"><AssetId>6fba512e-9049-4b01-a4cd-147c61327 595</AssetId><CategoryId>a49d2979-a930-4096-98c9-a127adf07760 </CategoryId><Location>XMS\3a</Location><AssetName>Crosses</A ssetName><CategoryName>Images\5cJdTest</CategoryName></Asset ></Assets></Page>
```

Take Tally V1

This tally is reported when an item is taken to air.

```
V1 %x| %s[|Params=%s]
```

CMD	Param_1	Param_2	Param_3
V0	%x Keyer number	%s Item specification (see page 370)	%s,%s,... Cue parameters

The first parameter is the keyer that was taken.

The second parameter is the specification of the item that was taken to air.

The third parameter is the parameters that the were specified when the item was cued. If no parameters are specified on the cue then none will be present in the tally.

Clear Tally V2

This tally is reported when either a cued or on-air item is cleared.

V2 %x | %x

CMD	Param_1	Param_2
V2	%x Keyer number	%x Cleared item was 0 = Cued 1 = On-air

The first parameter is the keyer that was cleared.

The second parameter reports whether a cued item or on-air item was cleared.

Keyer On Tally V3

This tally is reported when a keyer is switched on/up.

V3 %x

CMD	Param_1
V3	%x Keyer number

The parameter is the keyer that was switched on/up.

Keyer Off Tally V4

This tally is reported when a keyer is switched off/down.

V4 %x

CMD	Param_1
-----	---------

V4	%x
	Keyer number

The parameter is the keyer that was switched off/down.

Item Specifications

In V0 or V1 tallies, the 'item specification' contains the following information.

```
RecallId=<recall id>,ShortName=<short name with extension>,
FullName=<full xms name with category and extension>
```

or...

```
Embedded=<embedded page xml>
```

For example:

```
RecallId=3004,ShortName=PG_Style.vpg,FullName=XMS:Pages.JdTest
:PG_Style.vpg
```

```
RecallId=,ShortName=TPL_901.tem,FullName=XMS:Templates.JdTest:
TPL_901.tem
```

```
Embedded=<Page assetID="9722728b-add4-416e-a619-1e64d45aa246"
name="PG_CROSS" templateID="6617c1c7-742e-493f-b74f-e748e6609
91e" templateCatID="445bb9b3-27de-4702-8c3c-fc66cf73ded9" tem
plate="TPL_002" version="1.4" description="" categoryID=""><D
escription></Description><CreationDate></CreationDate><Modifi
cationDate></ModificationDate><Author></Author><RecallID></Re
callID><TargetDeviceType>D2</TargetDeviceType><Duration>0</Du
ration><Override><Textbox id="1" name="Textbox1"><Property na
me="Text">CROSS</Property></Textbox><Texture id="2" name="Tex
ture1"><Property name="ImageLocation">XMS\3a</Property><Prope
rty name="ImageCategory">a49d2979-a930-4096-98c9-a127adf07760
</Property><Property name="ImageName">6fba512e-9049-4b01-a4cd
-147c61327595</Property><Property name="AutoCreateJob">False<
/Property></Texture></Override><Assets><Asset type="Image" so
urce="Model"><AssetId>6fba512e-9049-4b01-a4cd-147c61327595</A
ssetId><CategoryId>a49d2979-a930-4096-98c9-a127adf07760</Cate
goryId><Location>XMS\3a</Location><AssetName>Crosses</AssetNa
me><CategoryId>Images\5cJdTest</CategoryId></Asset></Asse
ts></Page>
```

If any of the information is not available, for instance the item has no recall ID, the item specification is left blank. Note that a V0 or V1 tally is never used to report that nothing is cued or on-air (V2 is used instead). If all of the fields are empty, it means that the information was not available. Also note that the full

name and embedded page xml will likely contain oxtel escaped character sequences that need to be converted back to the original character.

Scene Event Summary

The scene event summary reports, for each keyer:

- The keyer state (on or off) using `V3` or `V4`<tt>
- The state of the cue buffer. Either <tt>`V2` if nothing is cued or `V0` if something is cued.
- The state of the on-air buffer. Either `V2` if nothing is on-air or `V1` if something is on-air.

For example, if there are two keyers (0 and 1). Keyer 0 is on, has nothing cued and 'PG_Style' is on air. Keyer 1 is off, has 'TPL_001' cued and nothing on air. The scene event summary would contain:

```
V3 0
V2 0|0
V1 0|RecallId=3004,ShortName=PG_Style.vpg,FullName=XMS:Pages.
JdTest:PG_Style.vpg
V4 1
V0 1|RecallId=3006,ShortName=TPL_001.tem,FullName=XMS:Templat
es.JdTest:TPL_001.tem
V2 1|1
```

Cue/Live Selection and Tallies in Action

Automation can enable either 'Oxtel video tallies' or 'Vertigo tallies' from the Intuition XG. Which tallies are received depends on whether automation sends a `Y61` (therefore get Oxtel video tallies) or `V81` (therefore get Vertigo tallies) to the Intuition XG last. You cannot register for both sets of tallies at the same time.

Note: The `Y6` and `V` tallies are mutually exclusive. You'll get either a `Y6` or a `V` tally depending on which tallies you enabled last.

When automation uses command forwarding (registered with `Y611`), the Imagestore enables one set of tallies with the Intuition XG. Which one depends on the Imagestore configuration tool option 'System Settings > Devices > Slave > slave type' within 'Configuration Mode'.

- "Intuition-XG" mode means that the Imagestore enables Vertigo tallies (sending `V81` to the slave).

- "Not Intuition-XG" mode means that the Imagestore enables Oxtel video (sending V61 to the slave).

As the Intuition XG receives various store/keyer commands, it sends tallies for each mode as follows:

CMD to Int-XG	Keyer Position	Change seen on Int-XG (X-Play)	Oxtel Tallies	Vertigo Tallies
R0	Down	Loads image to cue	Y9 for PST	V0
R0	Up	Loads image to cue, then live keyer	Y9 for PGM	V0 and V1
3	N/A	Cut keyer up/down, changes whether template is on PGM or PST, and puts '> Empty <' to other "store"	None	
V3	N/A	Cut keyer up	Y9 for PGM	
V4	N/A	Cut keyer down	Y9 for PST	
V0	Down	Loads image to cue	Y9 for PST	
V0	Up	Loads image to cue	None	
V1	Down	Does and Int-XG take	Y9 for PGM	
V1	Up	Does and Int-XG take	Y9 for PST	

Note the following Intuition XG behaviour (for layer "x"), which shows what happens to the cue file, on-air file and the keyer state when different commands are received by Intuition XG:

CMD to Int-XG	Cue File	On-air File	Keyer
Initial state	File A	File B	Down
3x 1	Empty	File A	Up
3x 0	File A	Empty	Down
Initial state	File A	File B	Down
V3 x ...	File A	File B	Up
V4 x ...	File A	File B	Down
Initial state	File A	File B	Down
V1 x ...	File C	File A	Down

Where 'File C' is the next file in the playlist, or Empty if there is none.

Command Validity Table

[to Introduction](#)

In the following table automation commands are grouped by function. An 'X' denotes that the command is supported by a particular product. Where a software version is shown, this is the first release in which the command was supported. An empty cell denotes that there is no support for this command.

Command	Imagestore 22U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	L GK-3901 ISM-3901
Keyers - Commands							
0	X	X	X		1.14.1	2.1	2.1
1	X	X	X	4.0	1.14.1	2.1	2.1
2	X	X	X		1.14.1	2.1	2.1
3	X	X	X	4.0	1.14.1	2.1	2.1
4	X	X			1.14.1	2.1	2.1
5	X	X			1.14.1	2.1	2.1
6	X						
7	X						
B	X	X	X	4.0	1.14.1	2.1	2.1
C	X	X			1.14.1	2.1	2.1
D	X	X			1.14.1	2.1	2.1
E	X	X			1.14.1	2.1	2.1
F	X	X	X		1.14.1	2.1	2.1
Keyers - Tallies							
3			X		1.14.1	2.1	2.1
Yc	X						
Yf					1.14.1	2.1	2.1
Stores - Commands							
R0	X	X	X	4.0	1.14.1		2.1
R3	X	X	X	4.2	1.14.1		2.1
R4	X	X	X	4.0	1.14.1		2.1
R5	X	X	X	4.0	1.14.1		2.1
R6	X	X	X	4.0	1.14.1		2.1

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
R7	X	X			1.14.1		2.1
R9		X			1.14.1		2.1
Ra	X						
Rm		X	X		1.14.1		2.1
RA			4.0	4.2			
RC					X	X	X
A	X	X	X	4.2	1.14.1		2.1
G	X	X			1.14.1		2.1
H	X	X			1.14.1		2.1
O	X	X	X	4.0	1.14.1		2.1
P	X	X			1.14.1		2.1
Q	X	X	X		1.14.1		2.1
Xi							2.1
Stores - Tallies							
Y9	X	X	X		1.14.1		2.1
YA		X			1.14.1		2.1
YB		1.13.8	1.13.8	4.2	1.14.1	2.1	2.1
YG					4.2		4.3
O		1.13.8	1.13.8		2.1	2.1	2.1
YO		1.13.8	1.13.8		1.14.1	2.1	2.1
Xi							2.1
YS							2.1.1
Animations - Commands							
S0	X	X	4.0		1.14.1		2.1
S1	X	X	4.0		1.14.1		2.1
S2	X	X	4.0		1.14.1		2.1
S4		X	4.0		1.14.1		2.1
Timers - Commands							
T0	X	X			1.14.1		2.1
T1	X	X			1.14.1		2.1
T2	X	X			1.14.1		2.1
T3	X	X			1.14.1		2.1

Command	Imagestore 2I/2U/3	Imagestore 300I[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
Easytext - Commands							
Z0	X	X	X		1.14.1		2.1
Z1	X	X	X		1.14.1		2.1
Z2	X	X	X		1.14.1		2.1
Z3	X	X	X		1.14.1		2.1
Z4	X	X	X		1.14.1		2.1
Z5	X	X			1.14.1		2.1
Z6	X	X	X		1.14.1		2.1
Z7	X	X	X		1.14.1		2.1
Z8	X	X	X		1.14.1		2.1
Z9	X	X	X		1.14.1		2.1
ZA	X	X	X		1.14.1		2.1
ZB	X	X	X		1.14.1		2.1
ZC	X	X	X		1.14.1		2.1
ZD	X	X			1.14.1		2.1
ZE	X	X	X		1.14.1		2.1
ZF	X	X	X		1.14.1		2.1
ZH					3.1		3.1
Za	X	X	X		1.14.1		2.1
Zb	X	X	X		1.14.1		2.1
Zc	X	X	X		1.14.1		2.1
Zd	X	X	X		1.14.1		2.1
Ze	X	X	X		1.14.1		2.1
Zf			X				
Zg		X	X		1.14.1		2.1
Zh			X				
Easytext - Tallies							
Yd	X						
Yg			1.13.8				
Yj			1.13.8				
Datasources - Commands							
m0		X	X	4.2	1.14.1		2.1
m1		X	X		1.14.1		2.1

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
m2		1.13.8	1.13.8	4.2	2.0.1		2.1
m3		1.13.8	1.13.8		2.0.1		2.1
m4		1.13.8	1.13.8		2.0.1		2.1
m5		1.13.8	1.13.8		2.0.1		2.1
m6		1.13.8	1.13.8		2.0.1		2.1
m7		1.13.8	1.13.8		2.0.1		2.1
m8		1.13.8	1.13.8		3.1		2.1
Emergency Alert System (EAS) - Commands							
n1	X	X			1.14.1		2.1
n2	X	X			1.14.1		2.1
n3	X	X			1.14.1		2.1
n4	X	X			1.14.1		2.1
SDI Video - Commands							
l					1.14.1	2.1	2.1
XM		X			1.14.1	2.1	2.1
Xq					3.1.2		
Xs		X	X		1.14.1	2.1	2.1
Xt					1.14.1		
A/B Mixer - Commands							
U0	X	X			1.14.1	3.0	3.0
U1	X	X			1.14.1	3.0	3.0
U2	X	X			1.14.1	3.0	3.0
U3	X	X			1.14.1	3.0	3.0
U4	X	X			1.14.1	3.0	3.0
U5	X	X			1.14.1	3.0	3.0
U6	X	X			1.14.1	3.0	3.0
U7	X	X			1.14.1	3.0	3.0
U8	X	X			1.14.1	3.0	3.0
U9	X	X			1.14.1	3.0	3.0
UA	X	X			1.14.1	3.0	3.0
UB	X	X			1.14.1	3.0	3.0
UC	X	X			1.14.1	3.0	3.0
UD	X	X			1.14.1	3.0	3.0

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
UE		Xk			1.14.1		
UF						2.1	2.1
UH					4.1		4.3
UI					4.1		4.3
UJ					4.1		4.3
Ua	X	X			1.14.1	3.0	3.0
Ub		X			1.14.1	3.0	3.0
A/B Mixer - Tallies							
Y6	X	X	X	4.0	1.14.1	2.1	2.1
U6					4.1	4.3	4.3
YU6					4.1	4.3	4.3
UE					4.1	4.3	4.3
UF					4.1	4.3	4.3
UI					4.1	4.3	4.3
YUI					4.1	4.3	4.3
UJ					4.1	4.3	4.3
YUJ					4.1	4.3	4.3
DVE - Commands							
W0	X						
W1	X	X			1.14.1		
W2	X	X			1.14.1		
W7		X			1.14.1		
W8		X			1.14.1		
WP					1.14.1		
WR					1.14.1		
WV					3.1		
WW					3.1		
WX					3.1		
WY					3.1		
WZ					4.1		
Wb					4.1		
Wd					4.1		
DVE - Tallies							

Command	Imagestore 2/2U/3	Imagestore 300I[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
YWb					4.1		
Wb					4.1		
W1					4.1		
WP					4.1		
Video Preview - Commands							
Y1	X	X			1.14.1	2.1	2.1
Y3	X	X			1.14.1	2.1	2.1
Yp					4.1	4.3	4.3
XH		X			1.14.1	2.1	2.1
XP		X			1.14.1		
XT					1.14.1		
Audio - Commands							
j0 0	X	X			1.14.1	3.0	3.0
j0 1	X	X			1.14.1	3.0	3.0
j0 2	X	X			1.14.1	3.0	3.0
j0 3	X	X			1.14.1	3.0	3.0
j0 4	X	X			1.14.1	3.0	3.0
j0 5	X	X			1.14.1	3.0	3.0
j1 0	X	X			1.14.1	3.0	3.0
j1 1	X	X			1.14.1	3.0	3.0
j1 2	X	X					
j1 3		X			1.14.1	3.0	3.0
j2 0	X	X			1.14.1	3.0	3.0
j2 1	X	X			1.14.1	3.0	3.0
j2 2	X	X			1.14.1	3.0	3.0
j30	X	X			1.14.1	3.0	3.0
j31	X	X			1.14.1	3.0	3.0
j32	X	X			1.14.1	3.0	3.0
j33	X	X			1.14.1	3.0	3.0
j34	X						
j35	X						
j40	X	X			1.14.1	3.0	3.0
j41	X	X			1.14.1	3.0	3.0

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
j42	X	X			1.14.1	3.0	3.0
j43	X	X			1.14.1	3.0	3.0
j441	X	X			1.14.1	3.0	3.0
j442	X	X			1.14.1	3.0	3.0
j451	X	X					
j452	X	X					
j461	X	X			1.14.1	3.0	3.0
j462	X	X			1.14.1	3.0	3.0
j471	X	X					
j472	X	X					
j50	X	X			1.14.1	3.0	3.0
j51	X	X			1.14.1	3.0	3.0
j52	X	X			1.14.1		
j53		X			1.14.1		
j6	X	X			1.14.1	3.0	3.0
j70	X	X			1.14.1	3.0	3.0
j71	X	X			1.14.1	3.0	3.0
j72	X	X			1.14.1	3.0	3.0
j73	X	X			1.14.1	3.0	3.0
j74	X	X			1.14.1	3.0	3.0
j75	X	X			1.14.1	3.0	3.0
j76					1.14.1		
j77					1.14.1		
j78					1.14.1	3.0	3.0
j79					1.14.1	3.0	3.0
j8	X						
j9	X						
ja	X	X			1.14.1	3.0	3.0
jb	X	X			1.14.1	3.0	3.0
jc	X	X			1.14.1	3.0	3.0
jd	X	X			1.14.1	3.0	3.0
je	X	X			1.14.1	3.0	3.0
jf	X	X			1.14.1	3.0	3.0

Command	Imagestore 2/2U/3	Imagestore 300I[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
jg	X	X			1.14.1	3.0	3.0
jh					1.14.1		
ji					1.14.1		
jj					1.14.1	3.0	3.0
jl		X			1.14.1	3.0	3.0
Audio - Tallies							
Y0	X	X			1.14.1	3.0	3.0
Y8	X	X			1.14.1	3.0	3.0
j70	X	X			1.14.1	3.0	3.0
j71	X	X			1.14.1	3.0	3.0
j72	X	X			1.14.1	3.0	3.0
j73	X	X			1.14.1	3.0	3.0
j74	X	X			1.14.1	3.0	3.0
j75	X	X			1.14.1	3.0	3.0
j76					1.14.1		
j77					1.14.1		
Advanced Audio - Commands							
jn0					3.1	3.0	3.0
jn1					3.1	3.0	3.0
jn2					3.1	3.0	3.0
jo0					3.1	3.0	3.0
jo1					3.1	3.0	3.0
jo2					3.1	3.0	3.0
jo3					3.1	3.0	3.0
jo4					3.1	3.0	3.0
jpo					3.1	3.0	3.0
jp1					3.1	3.0	3.0
jp2					3.1	3.0	3.0
jp3					3.1	3.0	3.0
jp4					3.1	3.0	3.0
jp5					3.1	3.0	3.0
jp6					3.1	3.0	3.0
jp7					3.1	3.0	3.0

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
jp8					4.1	4.3	4.3
jq0					3.1	3.0	3.0
jq1					3.1	3.0	3.0
jq2					3.1	3.0	3.0
jq3					3.1	3.0	3.0
jq4					3.1	3.0	3.0
jq5					3.1	3.0	3.0
jq6					4.1	4.3	4.3
jq7					4.2	4.3	4.3
jt0					3.1	3.0	3.0
Advanced Audio - Tallies							
YC					3.1	3.0	3.0
YC jo0					3.1	3.0	3.0
YC jp0					3.1	3.0	3.0
YC jp1					3.1	3.0	3.0
YC jp2					3.1	3.0	3.0
YC jp3					3.1	3.0	3.0
YC jq6					4.1	4.3	4.3
js					3.1	3.0	3.0
YC js					3.1	3.0	3.0
Easyplay - Commands							
k0	X	X			1.14.1		3.0
k1	X	X			1.14.1		3.0
k2	X	X			1.14.1		3.0
k3		X			1.14.1		3.0
k4		X			1.14.1		3.0
R8	X	X			1.14.1		3.0
RB	2.27	1.13.8			2.0.1		3.0
Easyplay 2 - Commands							
k6					3.1		3.0
k7					3.1		3.0
k8					3.1		3.0
k9					3.1		3.0

Command	Imagestore 2/2U/3	Imagestore 300I[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
ka					3.1		3.0
kb					3.1		3.0
kc					3.1		3.0
kd					3.1		3.0
ke					3.1		3.0
kf					3.1		3.0
Easyplay 2 - Tallies							
k8					3.1		3.0
k9					3.1		3.0
ka					3.1		3.0
kb					3.1		3.0
kc					3.1		3.0
kf					3.1		3.0
YE					3.1		3.0
Dolby/UpMix - Commands							
q00					2.0.1		
q10					2.0.1		
q20					2.0.1		
q30					2.0.1		
Metadata - Commands							
o0					2.0.1		
o1					2.0.1		
o2					2.0.1		
o3					2.0.1		
o4					2.0.1		
o5					2.0.1		
o6					2.0.1		
o7					2.0.1		
o8					2.0.1		
o9					2.0.1		
oA					2.0.1		
oB					2.0.1		
oC					2.0.1		

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
oD					2.0.1		
oE					2.0.1		
oF					2.0.1		
oG					2.0.1		
oH					2.0.1		
oI					2.0.1		
oJ					2.0.1		
oK					2.0.1		
oL					2.0.1		
oM					2.0.1		
oN					2.0.1		
oO					2.0.1		
oP					2.0.1		
oQ					2.0.1		
oR					2.0.1		
oS					2.0.1		
oT					2.0.1		
oU					2.0.1		
Passive Mode - Tallies							
Ya	X	X	X	4.0	1.14.1	2.1	2.1
Yb	X						
Ye	X	X	X	4.0	1.14.1	2.1	2.1
System and Status - Commands							
M	X	X	X	4.0	1.14.1	2.1	2.1
N	X	X	X	4.2	1.14.1	2.1	2.1
X1	X	X	X		1.14.1		
X2	X	X	X		1.14.1	2.1	2.1
X3	X	X	X		1.14.1	2.1	2.1
X4	X	X			1.14.1		
X5	X	X			1.14.1	2.1	2.1
XA	X	X			1.14.1	2.1	2.1
XC		X	X		1.14.1	2.1	2.1

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
XI		X	Xm		1.14.1	2.1	2.1
XJ		X	X		1.14.1	2.1	2.1
XK	X	X	X	4.0	1.14.1	2.1	2.1
XN		X	X		1.14.1	2.1	2.1
XO					X	X	X
XR		X	X		1.14.1	2.1	2.1
XS		X	X		1.14.1	2.1	2.1
XU					1.14.1	2.1	2.1
XX					3.1	2.1	2.1
Xb		1.13.8.1	1.13.8.1		2.0.2	2.1	2.1
Xc		X			1.14.1	2.1	2.1
Xe					1.14.1	2.1	2.1
Xn					3.0	3.0	3.0
Xu					2.0.1	3.1	3.1
Xv					2.0.1		
Xz						X	X
System and Status - Tallies							
Y7	X	X	X	4.0	1.14.1	2.1	2.1
YD						3.1	3.1
Health - Commands							
X0	X	X			1.14.1	2.1	2.1
X6	X	X			1.14.1		
XV					1.14.1		
Xhf					3.1		
Xhp					3.1	3.1	3.1
Xht					3.1	3.1	3.1
Xhv					3.1	3.1	3.1
Scheduled - Commands							
i0	X	X			1.14.1	2.1	2.1
i1	X	X			1.14.1	2.1	2.1
i2	X	X			1.14.1	2.1	2.1

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
i3	X	X			1.14.1	2.1	2.1
GPI - Commands							
X7	X	X			1.14.1	2.1	2.1
X8	X	X			1.14.1	2.1	2.1
X9	X	X			1.14.1	2.1	2.1
XB		X			1.14.1	2.1	2.1
Macro - Commands							
x0					4.1	4.3	4.3
x1					4.1	4.3	4.3
x2					4.1	4.3	4.3
x3					4.1	4.3	4.3
Arm and Take - Commands							
p0		X	X	X	X	X	X
p1		X	X	X	X	X	X
p4		X	X	X	X	X	X
Arm and Take - Tallies							
Yip0		X	X	X	X	X	X
Yip1		X	X	X	X	X	X
Yip4		X	X	X	X	X	X
Yip5		X	X	X	X	X	X
Yi		X	X	X	X	X	X
Routing Control - Commands							
rs0					4.1	4.3	4.3
rs1					4.1	4.3	4.3
rs2					4.1	4.3	4.3
rs3					4.1	4.3	4.3
rm0					4.1	4.3	4.3
rm1					4.1	4.3	4.3
rm2					4.1	4.3	4.3
Routing Control - Tallies							
YF					4.1	4.3	4.3
rs0					4.1	4.3	4.3
rs3					4.1	4.3	4.3

Command	Imagestore 2/2U/3	Imagestore 300[+] Imagestore HD-TV	Intuition[+]	Intuition XG	Imagestore 750	DSK-3901	LGK-3901 ISM-3901
rm1					4.1	4.3	4.3
rm2					4.1	4.3	4.3
Intuition XG - Commands							
V0				4.2			
V1				4.2			
V2				4.2			
V3				4.2			
V4				4.2			
V5				4.2			
V9				X			
Va				X			
Intuition XG - Tallies							
V8				4.2			
V0				X			
V1				X			
V2				X			
V3				X			
V4				X			

CRC Generation

The following C example demonstrates simple codes to format and send messages to an Imagestore. Imagestore status returns and acknowledges are ignored.

```
// Note these codes bear no relation to the ASCII defined
codes with similar names.

#define STX0                0x002
#define STX1                0x003
#define ACK0                0x004
#define ACK1                0x005
#define NAK                 0x007

char stx = STX0;

const UINT lstab[] =
{
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0,
    0x0280, 0xc241, 0xc601, 0x06c0, 0x0780, 0xc741,
    0x0500, 0xc5c1, 0xc481, 0x0440
};

const UINT mstab[] =
{
    0x0000, 0xcc01, 0xd801, 0x1400, 0xf001, 0x3c00,
    0x2800, 0xe401, 0xa001, 0x6c00, 0x7800, 0xb401,
    0x5000, 0x9c01, 0x8801, 0x4400
};

void do_crc(INT8 ch, UINT16 * crcptr)
{
    UINT tmp;

    tmp = *crcptr ^ ch;
    *crcptr = mstab[(tmp>>4) & 0xf] ^ lstab[tmp&0xf]
                ^ ((*crcptr) >> 8);
}

// Send a single command to an Imagestore, using printf style
formatting.
void remote_send(char * format,...)
{
    UINT16 rem_crc = 0;
    INT ch;
    char message[128];
    char* messageptr = (char*)message;
    va_list argptr;
```

```
va_start(argptr, format);
vsprintf(message, format, argptr);
va_end(argptr);

rem_send_char(stx);

while((ch = *messageptr++) != 0)
{
    rem_send_char(ch);
    do_crc(ch, &rem_crc);
}

rem_send_char(':');
do_crc(':', &rem_crc);

rem_send_char(rem_crc & 0xff);
rem_send_char(rem_crc >> 8);

if (stx == STX0)
    stx = STX1;
else
    stx = STX0;
}
```

Automation Examples

Whilst it is all very well to have a list of automation commands, individual commands are often only useful in conjunction with other related commands. It is the job of automation to sequence commands at appropriate time intervals to achieve the desired on-air effects. This chapter lists some basic sequences of automation commands to illustrate typical video operation.

Please note that automation commands can also be “batched” together into macros.

Keyer Animation Cut/Fade

In the following example, an animation called “Example.oxa” is loaded into DSK 2, cut up, held on-air for a period of time, faded down over 30 fields and then unloaded.

```

R01Example.oxa:    // Load 'Example.oxa' into DSK 2
                  // Wait for image to load
31 1:             // Cut up DSK 2
                  // Wait with image on-air
B1 1 1e:         // Set DSK 2 fade rate 60 fields
11 0:            // Fade down DSK 2
                  // Wait for fade-down to complete
A1:              // Erase DSK 2
  
```

A/B Mixing

The following example begins by performing a cross-fade from A to B over 120 fields. It then performs an asymmetric V-fade back to A via a black colour field – again over 120 fields (40 + 80).

```

U603:           // Set AB transition type: X-fade
U5078:          // Fade AB over 120 fields
                // Wait
UD000000:       // Set V-fade colour to black
U8028050:       // V-fade AB over 40 fields then
                // 80 fields
  
```

DVE Moves

The following example brings the DVE in circuit, and then loads and plays DVE sequence number 10 forwards over its default duration. We assume that this DVE sequence begins with a full-screen keyframe so that there will be no noticeable change on-air when the DVE is loaded. After waiting some time after the DVE sequence completes, the DVE sequence is then played backwards to the first (full-screen) keyframe. On completion of the reverse sequence, the DVE is then taken out of circuit.

```
W104:          // Put DVE in circuit
W20A3E7:       // Load first key frame of seq 10
W7001:         // Play DVE forward
               // Wait some time
W8001:         // Play DVE backward
               // Wait for reverse move complete
W109:         // Set DVE mode to disabled
```

Command Index

[to Introduction](#)

0	Fade Keyer to Black	0-----	42
1	Fade Keyer	1-----	42
2	Cut Keyer to Black	2-----	43
3	Cut Keyer	3-----	44
3	Keyer Position Tally	3-----	51
4	Set Layer Mask	4-----	44
5	Layer Mask Enable	5-----	45
6	Swap PVW and PGM Image using Cut	6-----	45
7	Swap PVW and PGM Image using V-	fade 7-----	46
A	Erase Store	A-----	66
B	Set Transition Duration	B-----	46
C	Set Self Key, Separate Key or No Key	C-----	47
D	Set Key Linear or Full	D-----	48
E	Set Key Normal or Invert	E-----	49
F	Enquire Clip, Gain and Transparency	F-----	50
F	Set Clip, Gain and Transparency	F-----	49
G	Enquire Image Position	G-----	67
G	Set Image Position	G-----	66
H	Enquire Live Video	H-----	69
H	Load Live Video	H-----	68
i0	Add Scheduled Command	i0-----	336
i1	Delete Scheduled Command	i1-----	336
i2	Delete All Scheduled Commands	i2-----	337
i3	Enquire Next Scheduled Command	i3-----	337
j0_0	Set Input Shuffle	j0_0-----	175
j0_1	Set Input Mute	j0_1-----	177
j0_2	Set Input Trim	j0_2-----	178
j0_3	Set Input Phase	j0_3-----	178
j0_4	Set Input Level	j0_4-----	179
j0_5	Set Input Source	j0_5-----	180
j1_0	Set Output Level	j1_0-----	182
j1_1	Set Output Mute	j1_1-----	184
j1_2	Output Embed	j1_2-----	184
j1_3	Set Output Level via Gain Type	j1_3-----	183
j2_0	Set Voice-over Duck	j2_0-----	187
j2_1	Set Voice-over Preset	j2_1-----	188
j2_2	Set Voice-over Position	j2_2-----	188
j30	Set Silence Fade Rate	j30-----	190
j31	Set A/B Mixer Fade Rate	j31-----	190
j32	Set Voice-over 1 Fade Rate	j32-----	191
j33	Set Voice-over 2 Fade Rate	j33-----	191
j34	Set Voice-over 3 Fade Rate	j34-----	192
j35	Set Voice-over 4 Fade Rate	j35-----	192
j40	Cut A/B	j40-----	194
j41	Fade A/B	j41-----	194
j42	Cut To Silence	j42-----	195
j43	Fade To Silence	j43-----	195
j441	Fade Voice-over 1	j441-----	195
j442	Fade Voice-over 2	j442-----	196
j451	Fade Voice-over 3	j451-----	196
j452	Fade Voice-over 4	j452-----	197
j461	Cut Voice-over 1	j461-----	198
j462	Cut Voice-over 2	j462-----	198
j471	Cut Voice-over 3	j471-----	199
j472	Cut Voice-over 4	j472-----	199
j50	Set Silence Follow Fade-to-Black	j50-----	201
j51	Set Audio A/B Follow Video A/B	j51-----	201
j52	Set Easyplay Follow Keyers	j52-----	202
j53	Voice-over Follow Easyplay	j53-----	202
j6	Set Audio Preview	j6-----	204
j7	Audio Tallies	j7-----	224
j70	Enquire Audio Input	j70-----	207
j71	Enquire Voice-over	j71-----	208
j72	Enquire Outputs	j72-----	209
j73	Enquire Rates	j73-----	210
j74	Enquire Audio Follow Video	j74-----	210
j75	Enquire Audio Preview	j75-----	211
j76	Enquire Audio Clean Feed	j76-----	212
j77	Enquire Audio Monitor	j77-----	213
j78	Enquire Audio AES Status	j78-----	214
j79	Enquire Output Gain	j79-----	214
j8	Set Audio FX Send	j8-----	220
j9	Set Audio Data Path	j9-----	221
ja	Audio A/B Position	ja-----	216
jb	Audio A/B Mix Mode	jb-----	216
jc	Audio A/B Asymmetric Transition	jc-----	216
jd	Audio A/B Fade to Position	jd-----	217

je	Set Audio A-Fade Position	je	-----	217
jf	Set Audio B-Fade Position	jf	-----	218
jj	Set Input Shuffle Bitwise	jj	-----	176
jh	Audio Clean Feed Select	jh	-----	205
ji	Audio Monitor Feed Select	ji	-----	206
jj	Set A/B Transition Switch Point	jj		218
jl	Set Audio Delay	jl	-----	219
jn0	Enquire Multiplexer Source	jn0		230
jn0	Set Multiplexer Source	jn0	-----	230
jn1	Enquire Multiplexer Source Count	jn1	-----	231
jn2	Enquire Multiplexer Source Name	jn2	-----	232
jo0	Set Shuffle Crosspoint	jo0	-----	233
jo1	Enquire Shuffle Preset	jo1	-----	234
jo1	Load Shuffle Preset	jo1	-----	234
jo2	Save Shuffle Preset	jo2	-----	235
jo3	Enquire Shuffle Preset List	jo3		235
jo4	Enquire Shuffle Block Size	jo4	--	236
jp0	Set Mute	jp0	-----	237
jp1	Set Trim	jp1	-----	237
jp2	Set Phase Invert	jp2	-----	238
jp3	Set Level	jp3	-----	238
jp4	Set Master Gain	jp4	-----	239
jp5	Enquire Gain Preset	jp5	-----	240
jp5	Load Gain Preset	jp5	-----	239
jp6	Save Gain Preset	jp6	-----	240
jp7	Enquire Gain Preset List	jp7	-----	241
jp8	Enquire Gain Block Size	jp8	-----	241
jq0	Enquire Voice-over Duck Level	jq0	-----	244
jq0	Set Voice-over Duck Level	jq0	--	244
jq1	Enquire Voice-over Preset Level	jq1	-----	245
jq1	Set Voice-over Preset Level	jq1		244
jq2	Enquire Voice-over Fade Rate	jq2	-----	245
jq2	Set Voice-over Fade Rate	jq2	--	245
jq3	Set Voice-over to Position	jq3	--	246
jq4	Fade Voice-over	jq4	-----	246
jq5	Cut Voice-over	jq5	-----	246
jq5	Enquire Voice-over Position	jq5		247
jq6	Enquire Voice-over Fade Rate	jq6	-----	248
jq6	Set Voice-over Fade Rate (Panel)	jq6	-----	247
jq7	Enquire PST Voice-over Position	jq7	-----	249
js	Audio Meter Tally	js	-----	257
jt0	Enquire Audio Meter Width	jt0	-----	249
k0	Load Audio Clip (Stream 1)	k0	---	260
k1	Start Audio Playout (Stream 1)	k1	-----	261
k2	Stop Audio Playout (Stream 1)	k2	-----	261
k3	Unload Audio Clip (Stream 1)	k3		262
k4	Enquire Audio Clip (Stream 1)	k4	-----	262
k6	Enquire Easyplay Stream Count	k6	-----	266
k7	Enquire Easyplay Queue Count	k7	-----	266
k8	Easyplay Associated Voiceover Tally	k8	-----	274
k8	Enquire Easyplay Associated Voiceover	k8	-----	267
k8	Set Easyplay Associated Voiceover	k8	-----	267
k9	Easyplay Play Follow Mode Tally	k9	-----	274
k9	Enquire Easyplay Play Follow Mode	k9	-----	268
k9	Set Easyplay Play Follow Mode	k9	-----	267
ka	Enquire Audio Clip	ka	-----	269
ka	Load Audio Clip	ka	-----	269
ka	Load Audio Clip Tally	ka	-----	275
kb	Play Audio Clip	kb	-----	270
kb	Play Audio Clip Tally	kb	-----	275
kc	Stop Audio Clip	kc	-----	270
kc	Stop Audio Clip Tally	kc	-----	276
kd	Unload Audio Clip	kd	-----	271
ke	Enquire Easyplay Stream Valid	ke	-----	272
kf	Easyplay Load Follow Mode Tally	kf	-----	276
kf	Enquire Easyplay Load Follow Mode	kf	-----	273
kf	Set Easyplay Load Follow Mode	kf	-----	272
1	Enquire Input Masks	1	-----	120
1	Set Input Masks	1	-----	120
M	Enquire System Status	M	-----	306
m0	Set Global Datasource Value	m0	-----	108
m1	Delete Global Datasource	m1	--	109
m2	Set Layer Datasource Value	m2		109
m3	Delete Layer Datasource	m3	---	109
m4	Enquire Global Datasource List	m4	-----	109

m5 Enquire Global Datasource Value	
m5-----	110
m6 Enquire Layer Datasource List	m6
-----	110
m7 Enquire Layer Datasource Value	
m7-----	111
m80 Delete All Datasources	m80---
-----	111
N Enquire Video Layer Status	N----
-----	307
n1 EAS On	n1-----
-----	117
n2 EAS Off	n2-----
-----	118
n3 EAS Set Text	n3-----
-----	118
n4 EAS Set Priority	n4-----
-----	118
O Enquire Image Status	O-----
-----	69
O Image Status Tally	O-----
-----	75
o0 Enquire Pitch Shift Code	o0-----
-----	282
o0 Set Pitch Shift Code	o0-----
-----	282
o1 Enquire Program Description	o1-----
-----	283
o1 Set Program Description	o1-----
-----	282
o2 Enquire Program Bitstream Mode	o2-----
-----	283
o2 Set Program Bitstream Mode	o2-----
-----	283
o3 Enquire Program Audio Coding Mode	o3-----
-----	284
o3 Set Program Audio Coding Mode	o3-----
-----	284
o4 Enquire Program Centre Down-Mix Level	o4-----
-----	285
o4 Set Program Centre Down-Mix Level	o4-----
-----	284
o5 Enquire Program Surround Down-Mix Level	o5-----
-----	286
o5 Set Program Surround Down-Mix Level	o5-----
-----	285
o6 Enquire Program Dolby Surround Mode	o6-----
-----	286
o6 Set Program Dolby Surround Mode	o6-----
-----	286
o7 Enquire Program LFE Enable	o7-----
-----	287
o7 Set Program LFE Enable	o7-----
-----	287
o8 Enquire Program Dialogue Normalisation	o8-----
-----	288
o8 Set Program Dialogue Normalisation	o8-----
-----	287
o9 Enquire Program Production Mix Level	o9-----
-----	288
o9 Set Program Production Mix Level	o9-----
-----	288
oA Enquire Program Production Room Type	oA-----
-----	289
oA Set Program Production Room Type	oA-----
-----	288
oB Enquire Program Copyright Flag	oB-----
-----	290
oB Set Program Copyright Flag	oB-----
-----	289
oC Enquire Program Original Bitstream Format	oC-----
-----	290
oC Set Program Original Bitstream Format	oC-----
-----	290
oD Enquire Program Down-Mix Mode	oD-----
-----	291
oD Set Program Down-Mix Mode	oD-----
-----	291
oE Enquire Program Down-Mix Lt/Rt Centre Mix Level	oE-----
-----	292
oE Set Program Down-Mix Lt/Rt Centre Mix Level	oE-----
-----	291
oF Enquire Program Down-Mix Lt/Rt Surround Mix Level	oF-----
-----	293
oF Set Program Down-Mix Lt/Rt Surround Mix Level	oF-----
-----	292
oG Enquire Program Down-Mix Lo/Ro Centre Mix	oG-----
-----	293
oG Set Program Down-Mix Lo/Ro Centre Mix	oG-----
-----	293
oH Enquire Program Down-Mix Lo/Ro Surround Mix Level	oH-----
-----	294
oH Set Program Down-Mix Lo/Ro Surround Mix Level	oH-----
-----	294
oI Enquire Program Dolby Surround EX™ Mode	oI-----
-----	295
oI Set Program Dolby Surround EX™ Mode	oI-----
-----	294
oJ Enquire Program Dolby Headphone Mode	oJ-----
-----	296
oJ Set Program Dolby Headphone Mode	oJ-----
-----	295
oK Enquire Program A/D Converter Type	oK-----
-----	296
oK Set Program A/D Converter Type	oK-----
-----	296
oL Enquire Program High Pass Filter Enable	oL-----
-----	297
oL Set Program High Pass Filter Enable	oL-----
-----	296
oM Enquire Program Low Pass Filter Enable	oM-----
-----	298
oM Set Program Low Pass Filter Enable	oM-----
-----	297

oN Enquire Program LFE LPF Enable	oN	RB Get Audio File Duration	RB	-----264
-----	298	RC Enquire Number of Media Files	RC	-----66
oN Set Program LFE LPF Enable	oN	Rm Enquire Image Load Mode	Rm	--64
oO Enquire Program Surround 90	oO	Rm Image Load Mode	Rm	-----63
Degree Phase-Shift	oO	rm0 Enquire Salvo Count	rm0	-----356
-----	299	rm1 Enquire Salvo Information	rm1	1357
oO Set Program Surround 90 Degree	oO	rm1 Salvo Information Tally	rm1	--359
Phase-Shift	oO	rm2 Execute Salvo	rm2	-----357
-----	298	rm2 Salvo Tally	rm2	-----360
oP Enquire Program Surround 3dB	oP	rs0 Make Source Selection	rs0	-----354
Attenuation	oP	rs0 Source Selection Tally	rs0	-----358
-----	300	rs1 Enquire Source Selection	rs1	--354
oP Set Program Surround 3dB	oP	rs2 Enquire Channel Source Count	rs2	-----355
Attenuation	oP	-----		
-----	299	rs3 Enquire Source Information	rs3	355
oQ Enquire Program RF Pre-emphasis	oQ	rs3 Source Information Tally	rs3	----359
-----	300	S0 Start Animation	S0	-----79
oQ Set Program RF Pre-emphasis	oQ	S1 Stop Animation	S1	-----79
-----	300	S2 Select Animation Frame	S2	-----80
oR Enquire Program Compression	oR	S4 Restart Animation	S4	-----80
Profile/Word	oR	T0 Stop Timer	T0	-----82
-----	301	T1 Start Timer	T1	-----82
oR Set Program Compression	oR	T2 Reset Timer	T2	-----83
Profile/Word	oR	T3 Set Timer to Value	T3	-----83
-----	301	U0 Cut to A	U0	-----125
oS Set Program Dynamic Range Value	oS	U1 Cut to B	U1	-----125
-----	302	U2 Fade to A	U2	-----126
oS Enquire Program Dynamic Range	oS	U3 Fade to B	U3	-----126
Value	oS	U4 Cut AB	U4	-----126
-----	302	U5 Fade AB	U5	-----127
oT Enquire Metadata Preset	oT	U6 Enquire Transition Type	U6	-----128
-----	281	U6 Set Transition Type	U6	-----127
oT Load Metadata Preset	oT	U6 Transition Type Tally	U6	-----144
-----	281	U8 Asymmetric V-Fade AB	U8	-----129
oU Enquire Program Dynamic Range	oU	UA Asymmetric Transition	UA	-----129
Type	oU	Ua Enquire Mix Mode	Ua	-----138
-----	303	Ub Enquire A/B Cut Mode	Ub	-----140
oU Set Program Dynamic Range Type	oU	Ub Set A/B Cut Mode	Ub	-----139
-----	302	UB Transition Complete	UB	-----129
p0 Disarm	p0	UC Fade to Specified Position	UC	--130
-----	346	UD Enquire V-Fade Colour	UD	-----131
p1 Arm	p1	UD Set V-Fade Colour	UD	-----130
-----	346	UE Enquire Mixer Input	UE	-----132
p4 Take	p4	UE Mixer Input Tally	UE	-----145
-----	346	UE Select Mixer Input	UE	-----132
q00 Set Up-Mix Mode	q00	UF Channel Input Tally	UF	-----146
-----	279	UF Enquire Channel Input	UF	-----133
q10 Set Dolby E Encode Mode	q10	UF Select Channel Input	UF	-----133
-----	279	UH Enquire A/B Mix Profile	UH	-----135
q20 Set Dolby Digital (AC3) Encode	q20			
Mode	q20			
-----	280			
q30 Set Dolby Decode Mode	q30			
-----	280			
R0 Enquire Load Image	R0			
-----	57			
R0 Load Image	R0			
-----	56			
R3 Enquire File Info	R3			
-----	58			
R4 Query First File	R4			
-----	59			
R5 Query Subsequent File	R5			
-----	59			
R6 Enquire Extended File Information	R6			
-----	60			
R7 Preload Image	R7			
-----	62			
R8 Enquire Audio File Information	R8			
-----	263			
R9 Emergency to Air	R9			
-----	62			
Ra Enquire Image Count	Ra			
-----	63			
RA Validate Template	RA			
-----	64			

UH Set A/B Mix Profile UH-----	134	X9 GPI Special X9-----	339
UI Enquire Mixer Transition Duration UI-----	137	XA Enquire Slave Layer Status XA --	314
UI Mixer Transition Duration Tally UI -----	146	Xb Enquire Full Version Number Xb320	
UI Set Mixer Transition Duration UI	136	XB GPI Output Status XB -----	340
UJ Channel Transition Mode Tally UJ -----	147	Xc Enquire Close Caption Xc-----	322
UJ Enquire Channel Transition Mode UJ -----	138	XC Enquire Serial Baud Rate XC-----	315
UJ Set Channel Transition Mode UJ	137	Xc Pass Close Caption Xc-----	321
W1 PGM DVE Mode Tallies W1-----	162	Xe Set Date and Time Xe-----	322
W1 Set PGM DVE Mode W1-----	149	Xh Enquire Extended Health Commands Xh -----	331
W2 Load and Run DVE Sequence W2 -----	151	XH Enquire Preview Output XH ----	166
W7 Run DVE Sequence W7-----	152	XH Set Preview Output XH-----	165
W8 Run DVE Sequence in Reverse W8 -----	152	Xhf Enquire Extended Fan Health Xhf -----	331
Wb DVE Sequence Tallies Wb-----	161	Xhp Enquire Extended PSU Health Xhp -----	332
Wb Play DVE Sequence Wb-----	158	Xht Enquire Extended Temperature Health Xht-----	333
Wd Enquire DVE ID Wd-----	159	Xhv Enquire Extended Voltage Health Xhv-----	334
Wd Enquire DVE ID Count Wd-----	159	XI Enquire Reference XI -----	316
Wd Set DVE ID Wd-----	159	Xi Enquire Store-Keyer Configuration Xi-----	71
WP PVW DVE Mode Tallies WP-----	162	Xi Set Store-Keyer Configuration Xi -	70
WP Set PVW DVE Mode WP-----	153	Xi Store-Keyer Configuration Tally Xi	77
WR Enquire DVE Routing Input WR	155	XJ Enquire Clock Offset XJ-----	316
WR Set DVE Routing Input WR-----	154	XK Enquire Serial Number XK -----	317
WV Set DVE Load Action WV-----	155	XM Enquire Input Mode XM-----	121
WW Enquire DVE Load Action WW	156	Xn Enquire Product Name Xn -----	323
WX Enquire DVE Configuration WX	157	XN Enquire Serial Protocol XN -----	318
WX Set DVE Configuration WX-----	156	XO Enquire Serial Type XO -----	318
WY Load DVE Sequence with Action WY-----	157	XP Enquire Clean Feed Output XP--	168
WZ Enquire DVE Sequence Name WZ -----	158	XP Set Clean Feed Output XP-----	166
X0 Enquire Temperature X0-----	328	Xq Enquire Post-A/B Mix Video Delay Xq-----	122
x0 Trigger Macro x0-----	342	Xq Set Post-A/B Mix Video Delay Xq -----	122
X1 Enquire Licence X1 -----	309	XR Enquire Number of Fill/Key XR--	319
x1 Enquire Macro Count x1-----	342	XS Enquire Preview Count XS -----	319
x2 Enquire Macro By ID x2-----	343	Xs Enquire Video Standard Xs-----	123
X2 Insert Log Message X2-----	310	XT Enquire Monitor Feed Output XT -----	170
X3 Enquire Command Availability X3 -----	310	Xt Enquire Output Frame Delay Xt -	124
x3 Enquire Macro By Index x3-----	343	XT Set Monitor Feed Output XT ----	169
X4 Enquire Device Bypass X4-----	312	Xt Set Output Frame Delay Xt-----	123
X4 Set Device Bypass X4-----	311	Xu Enquire AES Audio Signal Type Xu -----	323
X5 Enquire Input Colour Field X5--	312	XU Enquire Extended Serial Number XU-----	319
X6 Enquire Voltages X6-----	328		
X7 Virtual GPI X7-----	338		
X8 Enquire GPI Status X8-----	338		

Xv Enquire Embedded Audio Signal Type	YE Enquire Easyplay Tallies YE -----	277
Xv -----	Ye Enquire Passive Mode Ye -----	305
XV Enquire System Health XV -----	Yf Enable Router Tallies Yf -----	358
XX Enquire Input Status XX -----	YF Enquire Router Tallies YF -----	358
Xz Enquire URS Reference Xz -----	Yf Fade-To-Black Position Tally Yf ---	52
Y0 Audio Metering Tally Y0 -----	YG Load Status Tally YG -----	75
Y0 Enable Audio Metering Tallies Y0	Yg Text Crawl Count Tally Yg -----	104
-----	Yi Enable Arm/Take Tallies Yi -----	350
Y1 Set Preview Keyer Y1 -----	Yi Enquire Arm/Take Tallies Yi -----	350
Y3 Set Preview A/B Mix Y3 -----	Yip0 Disarm Tally Yip0 -----	348
Y6 Enable Video Tallies Y6 -----	Yip1 Arm Tally Yip1 -----	348
Y6 Enquire Video Tallies Y6 -----	Yip4 Take-Start Tally Yip4 -----	349
Y6 Video Tally Y6 -----	Yip5 Take-End Tally Yip5 -----	349
Y7 Request Wake-up Packet Y7 -----	Yj Animation/Clip Cycle Tally Yj -----	105
Y7 Wake-up Tally Y7 -----	YO Enable Image Status Tallies YO --	76
Y8 Audio Tally Y8 -----	YO Enquire Image Status Tallies YO -	77
Y8 Enable Audio Tallies Y8 -----	Yp Enquire Active Preview Keyers Yp	
Y8 Enquire Audio Tallies Y8 -----	-----	164
Y9 Image Load Tally Y9 -----	Yp Set Active Preview Keyers Yp ---	164
Ya Enable Passive Mode Tallies Ya -	YS Enable Store-Keyer Configuration	
304	Tallies -----	78
YA Image Preload Tally YA -----	YS Enquire Store-Keyer Configuration	
72	Tallies YS -----	78
YB Enable Media Tallies YB -----	YU6 Enable Transition Type Tallies YU6	
YB Enquire Media Tallies YB -----	-----	145
YB Media Tally YB -----	YUI Enable Mixer Transition Duration	
72	Tallies YUI -----	147
Yb Set Passive Mode Transmission	YUI Enquire Mixer Transition Duration	
Interval Yb -----	Tallies YUI -----	147
305	YWb Enable DVE Tallies YWb -----	161
YC Enable Advanced Audio Tallies YC	YWb Enquire DVE Tallies YWb -----	161
-----	Z0 Update Text Field Z0 -----	87
250	Z1 Change Box Size and Position Z1	87
YC Enquire Advanced Audio Tallies YC	Z2 Set Text Font and Colour Z2 -----	88
-----	Z3 Render Box Z3 -----	89
251	Z4 Change Image Z4 -----	89
Yc Layer Rate Tally Yc -----	Z5 Run Strap Z5 -----	100
52	Z6 Set Text Background to Matte Z6	96
YC_jp0 Enable Audio Mute Tallies	Z7 Set Text Background to Gradient Z7	
YC_jp0 -----	-----	96
253	Z8 Set Text Background to Clear Z8 -	98
YC_jp1 Enable Audio Trim Tallies	Z9 Set Text Drop Shadow Z9 -----	90
YC_jp1 -----	Za Enquire Textbox Za -----	93
254	ZA Set Transparency ZA -----	90
YC_jp2 Enable Audio Invert Tallies	Zb Enquire Template Zb -----	94
YC_jp2 -----	ZB Set Text Tracking ZB -----	91
255	Zc Enquire Crawl Count Loop Zc ---	102
YC_jp3 Enable Audio Level Tallies	Zc Enquire Crawl Loop Count Zc ---	101
YC_jp3 -----	ZC Set Strap Speed ZC -----	100
256	ZD Set Template Background ZD ----	98
YC_jq6 Enable Voice-over Fade Rate		
Tallies YC_jq6 -----		
257		
YC_js Enable Audio Meter Tallies YC_js		

258		
YC2js Enquire Audio Meter Tallies		
YC2js -----		
259		
YD Enquire Heartbeat Tallies YD ---		
327		
YD Heartbeat Tally YD -----		
326		
YD Set Heartbeat Tallies YD -----		
327		
Yd Strap Loop Count Yd -----		
104		
YE Enable Easyplay Tallies YE -----		
277		

Zd	Text Box Update	Zd-----	94	Zf	Stop Animation	Zf-----	102
Ze	Image Update	Ze-----	95	Zg	Pause/Restart Strap	Zg-----	100, 103
ZE	Set Text Alignment	ZE-----	92	Zh	Set Crawl Loop Count	Zh-----	103
ZF	Set Text Wrapping	ZF-----	92	ZH	Set Datasource Timer	ZH-----	101

Oxtel v16-001 PROVISIONAL