

G2_Command

I. Purpose of this document:

To provide an authorized standard, quality document detailing remote use of the G2 command. This document details what our remote software products **in the field** expect (what the developers think the software expects) and how it behaves in response to an incoming data stream with G2 commands.

These documents will be used to design tests of the remote software. Product deviation from these documents (verified by the developers either by test or by analysis of code) will initiate an ECR to update this document to reflect the deviation of the product.

Prevue BackOffice transmission Specifications are beyond the scope of this document.

II. Current code bases (remote Products) that use this command:

1	Amiga Prevue	ESQ (includes prevue laser, laserguide and international versions)
2	Amiga Sneak Prevue	VD (includes international versions)
3	PC Prevue (PC Prevue Laser and PC Prevue Junior)	Pcepg (includes international versions)

III. G2 Command overview:

CommandType = legacy EPG (55aa / body / xor checksum)

Function: The G2 command is used to instruct the c.g. at what point in time it locally enters daylight savings time (DST) and at what point it locally exits DST. The times specified in the command itself will always be based on a standard time (CST). For example, for all domestic units, the local DST entrance times will be identical (2:00 a.m.) but because the g2 command is specified in CST, Eastern time 2:00 becomes 1:00 CST, mountain time becomes 3:00 CST etc. so that there is a unique g2 for each time zone.

Addressing: This command is normally sent in all local transmissions. The state of the remote c.g. addressing must be 'BOXON' (i.e. must have received a valid address command) to process this command. The c.g. should process the information immediately and update all pertinent displays upon reception of a single valid G2 command.

Parsing: Although this command is a fixed length, it uses an optional embedded command length to help determine the end of the command.

Command Template:

```
<Command_Header><Command_ID><Length_plus1>
<G2.Range_String>
<term> <XOR_CheckSum>
```

g2 command template (Binary|Fixed)

Seq	Tag	Field Type	Description	Validation range	Min bytes	Max bytes	Offset
1	Command_Hdr	Binary Fixed	Standard Header	55h AAh	2	2	1-2
2	Command_ID	Ascii Fixed		'g2'	2	2	3-4
3	Length_plus1	Ascii Fixed	The # of bytes in the G2.Range_String +1	'27'	2	2	5-6
4	G2.Range_String	Ascii Fixed Multi	Subfield	See below	26	26	7-32
5	Term	Binary Fixed	Terminator	00h	1	1	33
6	XOR_CheckSum	Binary Fixed		00-FFh	1	1	34

```
< DST_InMarker ><JDay_In ><Time_In_ >
<DST_OutMarker><JDay_Out><Time_Out>
```

G2.Range_String template (Ascii |Fixed)

Seq	Tag	Field Type	Description	Validation range	Min bytes	Max bytes	Offset
1	G2.InMarker	Ascii Fixed	Delimiter	04h (^D)	1	1	1
2	G2.JDay_In.Year	Ascii Fixed	Yyyy	1970 – 2030	4	4	2-5
3	G2.JDay_In.Day	Ascii Fixed	Julian Day of change (day	001 – 366	3	3	6-8

G2_Command

			begins at Midnight CST)				
4	G2.Time_In	AsciiFixed	military time of change CST	00:00 – 23:59	5	5	9-13
5	G2.InMarker	Ascii lFixed	Delimiter	04h (^D)	1	1	1
6	G2.JDay_In.Year	Ascii lFixed	Yyyy	1970 – 2030	4	4	2-5
7	G2.JDay_In.Day	Ascii lFixed	Julian Day of change (day begins at Midnight CST)	001 – 366	3	3	6-8
8	G2.Time_In	AsciiFixed	military time of change CST	00:00 – 23:59	5	5	9-13
9	Time_Out	AsciiFixed		00:00 - 23:59	5	5	29-33
10							
11							

G2.JDay_In (BinarylFixed)

Seq	Tag	Field Type	Description	Validation range	Min bytes	Max bytes	Offset
1	G2.Jday_In.Year	Ascii lFixed	Yyyy	1970 – 2030	4	4	1-4
2	G2.Jday_In.Day	Ascii lFixed	Ddd	001 – 365	3	3	5-7

G2.JDay_Out (BinarylFixed)

Seq	Tag	Field Type	Description	Validation range	Min bytes	Max bytes	Offset
1	G2.Jday_Out.Year	Ascii lFixed	Yyyy	1970 – 2030	4	4	1-4
2	G2.Jday_Out.Day	Ascii lFixed	Ddd	001 – 366	3	3	5-7

Example G2 command (in hex): 55 AA 67 32 32 37 04 31 39 39 37 30 39 36 30 32 3A 30 30 13 31 39 39 37 32 39 39 30 31 3A 30 30 00 B9 0D 0A

This command is the same for all 3 members of the code bases (listed in section II above).

G2 command

This command will tell the box what time(s) it will go into and out of Daylight Savings Time (DST). The times specified in the command itself will always be based on a standard time (CST), to ensure ease of use as well as a global format.

# bytes	Description	Comment
1	0x55	Header pt. 1
1	0xAA	Header pt. 2
1	G	g family command
1	2	type '2' or local settings
2	length of command in bytes	## ((in ASCII) The number of bytes in the command, up to and including the null, but NOT the checksum!
1	^D	time (in std) to enter DST.
7	Julian Day of change	yyyddd - year/ Julian Day (in ASCII) of DST change.
5	military time of change	##:## - military time (also in ASCII) of DST change.
1	^S	time (in std) to re-enter non-DST.
7	Julian Day of change	yyyddd - year/ Julian Day (in ASCII) of DST change.
5	military time of change	##:## - military time (also in ASCII) of DST change.
1	0x00	NULL terminator
1	CS	checksum

G2_Command

G2_Command

G3 command

This command will tell the box what time(s) the data will go into and out of Daylight Savings Time (DST). The times specified in the command itself will always be based on a standard time (either DST or GST), to ensure ease of use as well as a global format.

<i># bytes</i>	<i>Description</i>	<i>Comment</i>
1	0x55	Header pt. 1
1	0xAA	Header pt. 2
1	g	g family command
1	3	type '3' or data settings
2	length of command in bytes	## ((in ASCII) The number of bytes in the command, up to and including the null, but NOT the checksum!
1	^D	time (in std) to enter DST.
7	Julian Day of change	yyyddd - year/ Julian Day (in ASCII) of DST change.
5	military time of change	##.## - military time (also in ASCII) of DST change.
1	^S	time (in std) to re-enter non-DST.
7	Julian Day of change	yyyddd - year/ Julian Day (in ASCII) of DST change.
5	military time of change	##.## - military time (also in ASCII) of DST change.
1	0x00	NULL terminator
1	CS	checksum