		DOCUMENTATION	
Auteurs : Marc Beckers		EVS Video Disk Odetics Communication Protocol For XT Server	
Date 24/03/2006	Version : 1.00.02		

Historical modifications

Date	Auteurs	Version	Objet
26/09/2003	Marc Beckers	01.00.00	Created
17/06/2004	Marc Beckers	01.00.01 For multicam 06.01.25	A8.18 =>Returns Xfile flags (toStore and isStored) B8.04 MakeClip BA.05 Set ID EVS Status =>To set/Reset the Xfiles flags. 28.31 Modified on a train=> do not go to live any more but just load at Out – 2.
01/10/2004	Marc Beckers	01.00.01	- Add Keywords functions
02/02/2005	Marc Beckers	Since multicam 06.02.22	New ID's conversion functions.
12/09/2005	Pierre Théate		New NetMove and NetCopy functions
13/10/2005	Pierre Théate	Since multicam 07.00.11	New GetMachine functions New Get/Set options functions New SetInOut function
23/03/2006	Marc Beckers	01.00.02	Function B1.09.01 Function B.8.13 Live

Contents

Odetics Commands For EVS Server

Commands that are shared gray are required for Odetics VDR protocol.

1 Odetics commands summary

Command	Return	Comment
00.0C Local Disable	10.01:Ack	Not Used
00.11 Device Type Request	12.11.D9.10 for PAL 12.11.D8.10 for NTSC	
00.0D Local Enable	10.01:Ack	Not Used

Command	Return	Comment
10.01 Ack		
11.12 Nack		

Command	Return	Comment
20.00 Stop	10.01:Ack	
20.01 Play	10.01:Ack	
20.02 Record	10.01:Ack	
20.04 StandByOff	10.01:Ack	Not Used
20.05 StandByOn	10.01:Ack	Not Used
20.0F Eject	10.01:Ack	Used to scan Lives on the player => EVS specific.
20.10 Forward	10.01:Ack	
2X.11 Jog Fwd	10.01:Ack	
2X.12 Var Fwd	10.01:Ack	
2X.13 Shuttle Fwd	10.01:Ack	
20.20 Rewind	10.01:Ack	
2X.21 Jog Rew	10.01:Ack	
2X.22 Var Rew	10.01:Ack	
2X.23 Shuttle Rew	10.01:Ack	
20.30 Preroll	10.01:Ack	Not Used
2X.31 CueUp With Data	10.01:Ack	

Command	Return	Comment
44.05 User Bits Preset	10.01:Ack	
40.10 Set In	10.01:Ack	Used with “MakeClip” & “SetInOut”
40.11 Set Out	10.01:Ack	Used with “MakeClip” & “SetInOut”
44.14 In Preset	10.01:Ack	If TC is 00:00:00:00 or is not specified then load and cue up on the first clip frame.
4X.15 Out Preset	10.01:Ack	If 40.15 then set Out point with the clip out point(.i.e. the last clip frame)

40.20 In Reset	10.01:Ack	
40.21 Out Reset	10.01:Ack	
40.30 Edit Preset	10.01:Ack	Not Used
44.31 Preroll Preset	10.01:Ack	Not Used
41.36 Set Timer Mode	10.01:Ack	Not Used
40.40 Set Auto Mode OFF	10.01:Ack	
40.41 Set Auto Mode ON	10.01:Ack	

Command	Return	Comment
61.0A Request TCGen	74.04	Not Used.
61.0C Request TimeCode	74.04 LTC Time Data (-current Time Code) 74.05 LTC User Bits Data 74.06 VITC Time Data 74.07 VITC User Bits Data 78.04 LTC Time and User Bits Data 78.06 VITC Time and User Bits Data	
60.10 Request IN	74.10	Not Used
60.11 Request OUT	74.11	Not Used
61.20 Status Sens	7X.20	
60.2E Request Speed	7X.2E	Not Used
60.31 Request Preroll	74.31	Not Used
60.36 Request Timer Mode	71.36	Not Used
A0.01 Auto Skip	10.01:Ack	
AX.02 Record Cue Up With Data.	10.01:Ack	
AX.04 Preview In Preset	10.01:Ack	
AX.05 Preview Out Preset	10.01:Ack	
A0.06 Preview In Reset	10.01:Ack	
A0.07 Preview Out Reset	10.01:Ack	
AX.10 Erase ID	10.01:Ack	If the clip ID is loaded in a channel, the clip deletion will failed.
A8.11 Erase Segment	Nack	Not implemented.
A0.14 List First ID	80.14 => No clip in the server 88.14 => first ID	
A0.15 List Next ID	Same as A0.14	
A8.16 List Clip Tc	80.16 or 88.16	Return the clip first frame time code and the clip duration. The Tc returned include the Drop/NoDrop bit on NTSC configuration (so does the function A8.17 and B8.06)
A8.17 List Clip Tc (EVS specific)	80.17 or 89.17	Same as A8.16 but after the clip duration, it also returns the clip machine number (used mainly with

		“ID Louth” configuration) on one byte.
A8.18 ID Status Request	81.18	Return the clip status bit. Bit 0 : 1 if the clip exist. Bit 1 : 1 if the clip is preloaded. Bit 4: ToStore (EVS) if the clip has to be saved on the Xfile. Bit 5: IsStored, if the clip is saved on the XFile.
A0.1C Longuest Contiguous Available Storage.	84.1C	This is the storage of the recorder or the recorder associated with the player.
A0.21 Device ID Request	88.21	
A8.20 Set Device ID	10.01:Ack	

B0.00 GetEvent	90.00 if no event 9X.00 if there is an event.	Asks for an EVS event from the queue.	
B1.01 SetTargetMachine	Ack	Sets the target machine used with the RecordCueUpWhithData (only whith IDLouth)	NOT YET IMPLEMENTED
B8.02 SetIdForData	Ack	Store a clip ID for future used by BC.02	
BC.02 SetData	Ack	Associates the data with the ID previously store by the command B8.02	
B8.03 GetData	9C.03	Gets the data associated with the ID supplied in the command.	
BX.04 MakeClip	Ack	Create a new clip with the given ID and with the values defined into IN and OUT. A guardband can also be defined.	
BA.05 Set ID EVS Status	Ack	Set some status on a given ID. Presently the Xfiles flags toStore and isStored.	
B8.06 List Clip Protect TC	98.06	This function is the same as the A8.17 (List clip TC) but it returns the Protects IN/OUT	
B9.07 GetKeyword	9D.07	Get a Keyword	
B8.08 SetKeyword 1	Ack	SetKeyword: Set the clip ID used with the function BD.08	
BD.08 SetKeyword 2	Ack	SetKeyword: set a keyword on the clip ID defined by the function B8.08.	
B9.09.04 ID LSM=>IDLouth	99.09.04 Or 91.09.04 if ID not found	Given an ID Lsm, the function returns the corresponding ID Louth	

B9.09.05 ID Louth=>ID LSM	99.09.05 Or 91.09.05 if ID not found	Given an ID Louth, the function returns the corresponding ID Lsm	
B9.0A NetMoveClipIdVDCP	Ack + Events	Move a clip to target machine. The source clip is identified by a VDCP id.	
B9.0B.53 NetMoveClipIdLsm 1	Ack + Events	Move a clip to target machine. Set the source clip ID (ID Lsm)	
B9.0B.54 NetMoveClipIdLsm 2	Ack + Events	Move a clip to target machine. Set the target clip ID (ID LSM)	
B8.0C NetCopyClipIdVDCP 1	Ack + Events	Copy a clip to target machine. Set the source clip ID (VDCP Id)	
B9.0C NetCopyClipIdVDCP 2	Ack + Events	Copy a clip to target machine. Set the target clip ID (VDCP Id)	
B9.0D.53 NetCopyClipIdLsm 1	Ack + Events	Copy a clip to target machine. Set the source clip ID (ID Lsm)	
B9.0D.54 NetCopyClipIdLsm 2	Ack + Events	Copy a clip to target machine. Set the target clip ID (ID LSM)	
B0.0E GetFirstMachine	90.0E if no machine. 9A.0E	Get the list of machines in the SDTI network. Use this function to get the first machine in the SDTI network.	
B0.0F GetNextMachine	90.0F if no machine. 9A.0F	Get the list of machines in the SDTI network. Use this function to get the next machine in the SDTI network.	
B4.10 SetOptions	Ack	SetOptions	
B0.11 GetOptions	94.11	GetOptions	
BX.12 SetInOut	Ack + Event	Update clip short in & short out	
B8.13 Live	Ack	Goes to Live on the given camera.	

Clip ID

The EVS server is able work in two different modes with the clip ID's.

The first mode is related with a "server" using ID of 8 bytes characters with no restrictions. That is what we called 'ID Louth' type of ID.

The second mode is closely related with the LSM. The clip ID for the LSM mode, we call 'IDLSM', has 8 bytes with the following conventions:

- Page (1 byte): [0..9]
- Bank(1 byte): [1..9]
- Clip number in the bank (1 byte): [0..9]
- Camera ID (1 byte): A, B, C, D, E or F (upper case !)
- The machine number (2 bytes): [1..29]
  defined in the EVS software under "Network" and in the field "Net Number".
- Last byte is a blank.

For instance 114A/00 is a valid ID. It is located on the local machine. The clip 120C/12 is also valid and is located on the remote machine numbered 12.

In the IDLSM mode, the functions to get the database also upload to the controller all the cameras (that we also call 'train'). These 'special' clips can not be used in playlist ('auto mode') and some functions have a special behaviour when used with cameras (i.e. mainly the CueUpWithData function).
Example: the LSMID for the CamA on machine number 17 is: 0 0 0 "A" 17 "blank"

In the IDLouth mode, the cameras can not be not uploaded. Only clips can be viewed from a controller, although a live can be view on a channel by the use of the Record or Eject command.

IDLouth or IDLSM mode has to be selected into the XT's VGA menu (shift-F2) in the "RS422 Protocols / ID Type" field.

Commands that have been extended for EVS server

Some commands are not implemented like it is specified in the Odetics protocol. This chapter explained how these commands are implemented and extended for the EVS server.

As the EVS server can not manage editing on an existing clip ID, the command RecordCueUpWithData is a subset of the corresponding command defined in the Odetics protocol documentation: *"Video Disk Recorder: Command and Control Specification" rev.F (6/16/98).*

2 CueUpWithData

This command is working differently if used with IDLSM or IDLouth.
The main is difference is when the function is used with a train.

20.31
If the current clip is a train then goto Live on this train.
If the current clip is not a train then cueup on the clip IN point.

24.31
This is the CueUp as implemented with the Sony protocol.

28.31
If the given clip is a train, then load at out – 2 (also works on a remote train).
If the given clip is not a train then load and cueup on the clip point IN.
If ID is a train that has never been started (IN point equal to OUT point) than does nothing.

2C.31
Load the given clip ID and cueUp on the given time code. If the given time code doesn't exist and ID is a train then cueup on the train out point(it is useful for controller using only TC=00:00:00:00 like the TDC100).
If ID is a clip and TC=00:00:00:00 then cueUp on the clip IN point.
If ID is a clip and TC doesn't exist in the clip then does nothing.

3 RecordCueUpWithData

A0.02
RecordCueUpWithData with no ID and no time code. Since the EVS server doesn't support clip editing, a new clip ID is created with an automatic ID. For IDLouth, the new clip ID will have the format CXXXXXXX where XXXXXXX is a number generated inside the server. The time code corresponding to the IN point of the newly created clip will be 00:00:00:00.

A4.02
This function is identical to the function A0.02 excepted for the time code corresponding to the IN point of the newly created clip will be the one given in the command.

A8.02 and **AC.02**
They have the same behaviour as explained in the Odetics protocol documentation excepted if the given ID already exists. In this case, the command is simply refused.

This function reset the out preset value. The function uses the couple Preroll/CueUp (see CueUpWithData) to indicate the command status. If, after the command, these two bits are not set then an error occurred.

4 Record

This function is working differently if a RecordCueUpWithData has been sent or not.

- if RecordCueUpWithData has been sent, then it is the record command as defined for the Odetics protocol.
- If RecordCueUpWithData has not been sent before then there is two cases:
    - If no clip has been previously preloaded or the current clip is not a train then the default train is loaded on live.
    - If the current clip is already a train then it goes on live.

It is possible to play a clip while it is recording on another player channel.

5 EVS Additional commands

5.1 GetEvent (B0.00)

Each Odetics communication port maintains a queue of events.
The return of this function depends of the event that has occurred into the system.

Format : B0.00
Return : 9X.00.XX.
XX is the event type (not present when there is no event in the queue, i.e. when 90.00)
Following bytes are the data (optional).

	Event byte value	Data bytes	Comment
90.00	None	0	No event into the queue.
99.00	0x01	8 bytes: an ID	The clip ID has been created on the server (or on another server in the network)
99.00	0x02	8 bytes: an ID	The clip ID has been deleted on the server (or on another server in the network)
99.00	0x03	8 bytes: an ID	The clip ID creation has failed.
99.00	0x04	8 bytes: an ID	The clip ID deletion has failed
99.00	0x05	8 bytes: an ID	A clip ID has been created by the RecordCueUpWithData. It can not be played yet but it is already present in the database.
92.00	0x06	1 byte: machine number (in binary)	The given machine (not the local) has been disconnected from the network. Those clips are no longer available.
91.00	0x07	No data	The local machine has been disconnected from the network. Only local clips are now available.
92.00	0x08	1 byte: machine number (in binary)	The given machine has been connected to the network. Local clips are now available.
99.00	0x09	8 bytes: clip ID	This clip id is already used.
92.00	0x0A	1 byte: machine number (in binary)	The machine specified is a XFile that does not support

			the requested function.
92.00	0x0B	1 byte: machine number (in binary)	The target machine is not found.
99.00	0x0C	8 bytes: clip ID	This clip id is not found.
99.00	0x0D	8 bytes: clip ID	This clip id is protected.
92.00	0x0E	1 byte: machine number (in binary)	No more free space on the target machine.
93.00	0x0F	2 bytes: error code	SDTI network error.

This function is not fixed and new events can be added in the future.

5.2 SetIdForData/SetData (BX.02)

A 12 characters data can be associated with a clip ID. This function selects the target ID that will be used with the command SetData. SetData contains the data that will be store with the previously defined clip ID. These functions have been split to keep the Sony protocol format for the commands.

Ex:
First, we send the target clip ID that will receive a new 12 characters data:
B8 02 43 6C 69 70 30 30 31 37 cs = 0a (The ID is Clip0017)
Then we sent the data (the data is ABCDEFGHIJKL):
BC 02 41 42 43 44 45 46 47 48 49 4A 4B 4C cs = 0c

5.3 List Clip Protect TC (B8.06)

This function is the same as the A8.17 (List clip TC) but it returns the Protects IN/OUT instead of the short In/ short Out points. The difference is the “GuardBands” included with the Protects.

5.4 GetData (B8.03)

Return the data (.i.e the name as called in the XT Server) associated with the clip ID specified in the command.
Ex: To get the data associated with the clip Clip0017, the command will be:
B8 03 43 6C 69 70 30 30 31 37 cs = 0b
The return will be:
9C.03 XX XX XX XX XX XX XX XX XX XX XX XX

5.5 MakeClip (BX.04)

There is two groups of functions to create a clip.
First group is related to ID LSM:

- B4.04: only use a format like 111A
- B6.04: like the previous function but with a guardband.
- B7.04: use a full LSM ID like 113B/00 for the local machine or 114C/23 for a networked machine.
- B9.04: use a full LSM ID and a guardband.

Second group is related to ID Louth:

- B8.04 only use a clip ID Louth.
- BA.04 use an ID Louth and a guardband.

The guardband is always transmitted on two bytes.

Ex:
First use the functions 40.10 and 40.11 to define the time code IN and time code OUT of the new clip. Then use the MakeClip to create the clip 111A/00 with a guardband of 0:

```
B9 04 31 31 31 41 2F 30 30 00 00   cs = 20
```

Here is an example with an ID Louth=Clip0101 and a guardband of 5 frames:

```
BA 04 43 6C 69 70 30 31 30 31 05 00 cs = 0d
```

5.6 SetIDEVSStatus (BA.05)

This function set EVS specifics status on a given clip. The format is:

ID: 8 bytes. The target clip ID.

Bitmap: 1 byte. Defines the content of the command. Presently only bit 0x01 is used to specify Xfile status.

Status: 1 byte. The new status:

- Bit 0x01:”toStore” => the clip will be stored in the Xfile.
- Bit 0x02:”stored” => the clip is stored in the Xfile.

To get the Xfile status, the function A8.18 has been extended. The bit 0x10 is the “toStore” status and the bit 0x20 is the “stored” status.

Example: To Set the Xfile flag “ToStore” on the clip “Clip0017”:

```
BA 05  43 6C 69 70 30 30 31 37 01 01 cs = 11
```

The last byte is 01 to set the "ToStore" and reset the "Stored".
To Set the "Stored" flag and reset the ToStore", use 02
To reset both flags, use 00.

To get these flags on a given clip, use the standard A8.18 ID Status Request Odetics command.
The returns is 81.18. The returned byte (let say "rc") contains:
- rc & 0x10 is the "ToStore" XFile flag.
- rc & 0x20 is the "Stored" XFile flag
- rc & 0x01means "Does this clip exist on the system"
- rc & 0x02 means "Does this clip is loaded on the channel"

5.7 GetKeyword (B9.07)

Gets a keyword associated with a clip. A clip can have three keywords. A keyword is a 12 characters long data. The format is:

Format : B9.07 + Clip ID (8 bytes) + keyword selector (1 byte)
The keyword selector is in the range [1..3] to select the target keyword.
Return : 9D.07 + keyword (12 bytes) + keyword selector (1 byte).

If the clip ID doesn't exist, the return is 90.07.

5.8 SetKeyword (BX.08)

This function is split into two parts:

Format :
B8.08 + clip ID
This function is used to store a clip ID for a future used for the keyword setting.
BD.08 + keyword (12 bytes) + keyword selector (1 byte)
This function is used to store keyword with the clip ID previously defined with the B8.08 function. The keyword selector defines the target selector ([range is [1..3]).

Events (see GetEvent here above) :
0x0C : clip id not found

The target clip ID is not reset after this function. It is then possible to use the BD.08 function three times with the appropriate keyword selector to define the three keywords associated with a clip.

5.9 General information function (BX.09)

This new function is used to get information specifics to the XT server. The third byte is used as a selector for the information requested.

Format :
BX 09 YY Data, where
BX 09 is the function, X depending on the selector.
YY is the function selector.
Data is the data depending on the function selector.

5.9.1 Current ID Loaded on the channel (ActiveIDRequest).

Returns the clip ID currently loaded on the channel. This function can be useful if the clip has been loaded by an external source (another protocol that also has the control on this channel for instance).

Format:
B1.09.01

Return:

0x99.09.01 + ID (Louth or LSM) of the clip loaded.
0x91.09.01 if no clip loaded on the channel.

5.9.2 Conversion IDLSM⇔ID Louth.

Format:

B9.09.04 + ID LSM(8 bytes) Convert the given ID LSM to its corresponding ID Louth.
B9.09.05 + ID Louth(8 bytes) Convert the given ID Louth to its corresponding ID LSM.

Return :

The return is 99.09.04 + ID Louth(8 bytes) or 91.09.04 if the given ID LSM has not been found.
The return is 99.09.05 + ID LSM(8 bytes) or 91.09.05 if the given ID Louth has not been found.

5.10 NetMoveClipIdVDCP (B9.0A)

Move a clip to target machine.
The source clip is identified by a VDCP id.

Format: B9.OA + VDCP clip id (8 bytes) + machine target id (1 byte)

Events (see GetEvent here above) :

0x0A : XFile does not support this move clip
0x0B : target machine not found
0x0C : clip id not found
0x0D : clip id is protected
0x0E : No more free space on the target machine
0x0F : SDTI network error

5.11 NetMoveClipIdLsm (B9.0B)

Move a clip to target machine.
The source clip is identified by a Lsm id.
The target machine is identified by the target Lsm id

Format : B9 0B ‘X’ + LSM id (8 bytes), where ‘X’ is the function selector :

X = ‘S’ (53) => set the source clip id
X = ‘T’ (54) => set the target clip id and do the move

Events (see GetEvent here above) :

0x09 : clip id is already used
0x0A : XFile that does not support this move clip
0x0B : target machine not found
0x0C : clip id not found

0x0D : clip id is protected
0x0F : SDTI network error

This function is split into two parts :

First, set the source clip ID :
B9 0B 53 + Lsm source id (8 bytes)
Then do the move
B9 0B 54 + Lsm target id (8 bytes)

5.12 NetCopyClipIdVDCP (BX.0C)

Copy a clip to target machine.
The source and target clips are identified by a VDCP id.

This function is split into two parts :

Format :
B8 0C + source VDCP ID (8 bytes)
First, set the source clip ID :
B9 0C + ID target machine (1 byte) + target VDCP ID (8 bytes)
Then do the copy

Events (see GetEvent here above) :
0x09 : clip id is already used
0x0A : XFile that does not support this copy clip
0x0B : target machine not found
0x0C : clip id not found
0x0F : SDTI network error

5.13 NetCopyClipIdLsm (B9.0D)

Copy a clip to target machine.
The source and target clips are identified by a Lsm id.
The target machine is identified by the target Lsm id

Format : B9 0D ‘X’ + ID LSM (8 bytes), where ‘X’ is the function selector :
X = ‘S’ (53) => set the source clip ID
X = ‘T’ (54) => set the target clip ID and do the copy

Events (see GetEvent here above) :
0x09 : clip id is already used
0x0A : XFile that does not support this copy clip
0x0B : target machine not found
0x0C : clip id not found
0x0F : SDTI network error

This function is split into two parts :

First, set the source clip ID :
B9 0D 53 + Lsm source id (8 bytes)
Then do the move
B9 0D 54 + Lsm target id (8 bytes)

5.14 GetFirstMachine (B0.0E)

Get the list of machines in the SDTI network.
Use this function to get the first machine in the SDTI network.

Format : B0.0E
To get the first machine.

Return :
If the return is 90.0E => there is no machine running in the SDTI network.
Else, the return is as following :
9A.0E + machine name (8 first bytes) + machine id (1 byte) + bits field (1 byte)
The bits field is as following :

7	6	5	4	3	2	1	0
Reserved				machine type			clip edit by network

Bits 1-3 = machine type => 8 types possible
For now, only 4 types are used :
0 = Unknown
1 = LSM
2 = XFile
3 = Max
4-7 = Reserved

Example for this bits field :
Bits field = 0x03 means the machine is a LSM and the Clip edit by network is set to “Yes”
Bits field = 0x06 means the machine is a Max and the Clip edit by network is set to “No”

5.15 GetNextMachine (B0.0F)

Get the list of machines in the SDTI network.
Use this function after a first call to previous function GetFirstMachine (B0.0E).

Format : B0.0F
To get the next machine.

Return :

As long as the return is not 90.0E, call this function to get all the machines running in the SDTI network.
The return is as following :

9A.OF + machine name (8 first bytes) + machine id (1 byte) + bits field (1 byte)
The bits field is described here above in the previous function GetFirstMachine (B0.0E).

5.16 SetOptions (B4.10)

Format : B4.10 + options (4 bytes)

options is a bits field (32 bits), used as following :

31 … 1	0
Reserved	Use guard bands

For now, only bit 0 is used :
If it is set, it is allowed to use guard bands, i.e. outside the range short in ← → short out

5.17 GetOptions (B0.11)

Format : B0.11

Return : 94.11 + options (4 bytes)

See options described here above in the previous function SetOptions (B4.10).

5.18 SetInOut (BX.12)

Update clip short in & short out
First use the functions 40.10 and 40.11 to define the time code Short IN and time code Short OUT of the clip to be updated, identified by the given clip id.

Format :

B4.12 + ID LSM format like 111A for the local machine (4 bytes)
B7.12 + ID LSM format like 113B/00 for the local machine or 114C/23 for a networked machine (7 bytes)
B8.12 + ID Louth (8 bytes)

Events (see GetEvent here above) :

0x0C : clip id not found

5.19 Live (B8.13)

This command goes to live on the current channel with the camera specified as argument (8 bytes for the camera id). The id can be specified as an ID LSM or an ID Louth. It can also be a remote camera (i.e. not local).