VGHD Data FOLDER - MODELS.LST
Another mystery solved by WyldAnimal

****
********** NEW VERSION 1.1.0.62 - New format Version 276 **********
****

models.lst structure New Version 00 00 01 14 = 276
start of file
4 bytes of code ? (possibly a version number ?) 00 00 01 14 = 276
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(Hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
4 bytes of code 00 00 00 – don't know what they are for
5 bytes of code ff ff ff ff ff -- don't know what they are for
4 bytes of Code 00 00 00 00 - not sure
4 bytes of Code 00 00 00 00 – not sure
New for Version 276 - VG GUI 1.1.0.62 and up
4 bytes of Code 00 00 00 00 - not sure
4 bytes of Code 00 00 00 00 – not sure
End of New for 276
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 bytes 00 00 00 01 - is clip active in clip list 00 00 00 01 Active - 00 00 00 00 Not Active
Repeat C**LIPLOOP**: for all clips in this card

Repeat **CARDLOOP**: for all cards in all collections, or until end of file is found.
How to read the Models.lst file for version 00 00 01 14 = 276
all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 14 = 276
read in 4 bytes of code = convert to number = Total Number of CARDS

*__Card loop__ - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = Number of Other cards model appears in.
Read in 4 Bytes of Code – not sure 00 00 00 00
Read in 5 Bytes of Code – not sure ff ff ff ff ff
Read in 4 Bytes of Code – not sure 00 00 00 00
Read in 4 Bytes of Code – not sure 00 00 00 00
Read in 4 Bytes of Code – not Sure 00 00 00 00 - new for version 276
Read in 4 Bytes of Code – not Sure 00 00 00 00
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**__Clip loop__ - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 Bytes = 00 00 00 01 - 1st byte = card Active or Not 01=active 00=not
Read 4 Bytes of Code - not yet sure what it is for? weight of clip? - how often it has been played?
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END NEW VERSION 1.1.0.62 Version 276 **********
****

****
********** NEW VERSION 1.1.0.54 - New format Version 273 **********
****

models.lst structure New Version 00 00 01 11 = 273
start of file
4 bytes of code ? (possibly a version number ?) 00 00 01 11 = 273
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
New for Version 272
4 bytes of code 00 00 00 – don't know what they are for
5 bytes of code ff ff ff ff ff -- don't know what they are for
End of new version changes 272
New for Version 273
4 bytes of Code 00 00 00 00 - not sure
4 bytes of Code 00 00 00 00 – not sure
End of additions to version 273
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 bytes 00 00 00 01 - is clip active in clip list 00 00 00 01 Active - 00 00 00 00 Not Active
4 bytes of code - 00 00 40 00 - not sure what this is for

Repeat C**LIPLOOP**: for all clips in this card
Repeat **CARDLOOP**: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 00 00 01 11 = 273
all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 11 = 273
read in 4 bytes of code = convert to number = Total Number of CARDS

*__Card loop__ - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = Number of Other cards model appears in.
Read in 4 Bytes of Code – not sure 00 00 00 00 - new for version 272
Read in 5 Bytes of Code – not sure ff ff ff ff ff
Read in 4 Bytes of Code – not Sure 00 00 00 00 - new for version 273
Read in 4 Bytes of Code – not Sure 00 00 00 00
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**__Clip loop__ - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 Bytes = 00 00 00 01 - 1st byte = card Active or Not 01=active 00=not
Read 4 Bytes of Code - not yet sure what it is for? weight of clip? - how often it has been played?
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END NEW VERSION 1.1.0.54 Version 273 **********
****

****
********** NEW VERSION 1.1.0.30 10-22-2012 - New format **********
****

models.lst structure New Version 00 00 01 10 = 272
start of file
4 bytes of code ? (possibly a version number ?) 00 00 01 10 = 272
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
New for this Version
4 bytes of code 00 00 00 00 – don't know what they are for
5 bytes of code ff ff ff ff ff -- don't know what they are for
End of new version changes
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 bytes 00 00 00 01 - is clip active in clip list 00 00 00 01 Active - 00 00 00 00 Not Active
4 bytes of code - 00 00 40 00 - not sure what this is for

Repeat C**LIPLOOP**: for all clips in this card
Repeat **CARDLOOP**: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 00 00 01 10 = 272

all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 10 = 272
read in 4 bytes of code = convert to number = Total Number of CARDS

*__Card loop__ - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = Number of Other cards model appears in.
Read in 4 Bytes of Code – not sure 00 00 00 00
Read in 5 Bytes of Code – not sure ff ff ff ff ff
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**__Clip loop__ - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 Bytes = 00 00 00 01 - 1st byte = card Active or Not 01=active 00=not
Read 4 Bytes of Code - not yet sure what it is for? weight of clip? - how often it has been played?
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END NEW VERSION 1.1.0.30 10-22-2012 **********
****

****
********** NEW VERSION 1.1.0.1 02-15-2012 - reverted back to 1.0.7.0 format **********
****

models.lst structure New Version 00 00 01 0f
start of file
4 bytes of code ? (possibly a version number ?) 00 00 01 0f
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 bytes 00 00 00 01 - is clip active in clip list 00 00 00 01 Active - 00 00 00 00 Not Active
4 bytes of code - 00 00 40 00 - not sure what this is for

Repeat C**LIPLOOP**: for all clips in this card
Repeat **CARDLOOP**: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 00 00 01 0f
all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 0f
read in 4 bytes of code = convert to number = Total Number of CARDS

*__Card loop__ - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = Number of Other cards model appears in.
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**__Clip loop__ - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 Bytes = 00 00 00 01 - 1st byte = card Active or Not 01=active 00=not
Read 4 Bytes of Code - not yet sure what it is for? weight of clip? - how often it has been played?
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END NEW VERSION 1.1.0.1 02-15-2012 **********
****

****
********** NEW VERSION 1.1.0.0 01-30-2012 **********
****

Models.lst structure New Version 00 00 01 0e
start of file
4 bytes of code ? (possibly a version number ?) 00 00 01 0e
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card

**** New Section for Cardloop:
4 bytes of code - hex - number of Other cards this model is in
loop:
for each card model is in from above
4 bytes of code - hex number - length of string of card name
next bytes of code - this is the ASCII card number as a two byte HL pair - read in number of bytes, then split into pairs.
end of loop:
**** end of New Section:
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
New - clips unchecked in the VIP clip view will be (a0001_64412.vghd.off) 4 extra bytes added to the end (.off)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 bytes 00 00 00 01 - is clip active in clip list 00 00 00 01 Active - 00 00 00 00 Not Active
4 bytes of code - 00 00 40 00 - not sure what this is for

Repeat CLIPLOOP: for all clips in this card
Repeat CARDLOOP: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 00 00 01 0e
all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 0e
read in 4 bytes of code = convert to number = Total Number of CARDS

*Card loop - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = Number of Other cards model appears in.
Loop for number of other cards
Read in 4 bytes and convert to a number = Length of Card Name (L)
Read in (L) Bytes = Unicode 2byte HL of Card name
End of Loop
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**Clip loop - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 Bytes = 00 00 00 01 - 1st byte = card Active or Not 01=active 00=not
Read 4 Bytes of Code - not yet sure what it is for? weight of clip? - how often it has been played?
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END NEW VERSION 1.1.0.0 01-30-2012 **********
****

****
********** NEW VERSION 1.0.7.0 **********
****

models.lst structure New Version 00 00 01 0a
start of file
4 bytes of code ? (possibly a version number ?)
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
Now using Big-Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
next 4 bytes of code - this is the date the card was added to your collection 00 mm yy dd - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011
4 bytes of code - card status - See *** Status codes only last two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
4 bytes of code - 00 00 10 00 - this is a new addition (4096 dec value) not sure what it is used for.
** the above 4 bytes have been removed from 1.1.0.0 and also 1.1.0.1
4 bytes of code - hex - number of clips you have rights for in this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
New - clips unchecked in the VIP clip view will be (a0001_64412.vghd.off) 4 extra bytes added to the end (.off)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

New - Addition of 4 extra bytes now.
4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4
4 new hex bytes 00 00 00 01 - not sure what this is for...

Repeat CLIPLOOP: for all clips in this card
Repeat CARDLOOP: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 00 00 01 0a
all hex numbers are in 4 byte order Big Endian
(hex numbers are High byte Low byte ordered, read H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 00 00 01 0a
read in 4 bytes of code = convert to number = Total Number of CARDS

*Card loop - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes = DATE in HEX - when card was added to the collection 00 mm yy dd
Base Year starts a 1890, add yy to that, for mm subtract 27, for dd subtract 26
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 bytes 00 00 10 00 = dec 4096
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**Clip loop - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
Read 4 bytes = 00 00 00 01
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.
****
********** END OF NEW VERSION 1.0.7.0 **********
****

****
********** NEW VERSION 1.0.5.x to 1.0.6.x **********
****

models.lst structure New Version 05 01 00 00
start of file
4 bytes of code ? (possibly a version number ?)
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
(hex numbers are low byte high byte ordered, read L2 L1 H2 H1 reorder to H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
4 bytes of code - hex - length of the date
next bytes of code - this is the date the card was added to your collection yyyy-mm-dd
4 bytes of code - card status - See *** Status codes only first two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
4 bytes of code - hex - number of clips for this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
New - clips unchecked in the VIP clip view will be (a0001_64412.vghd.off) 4 extra bytes added to the end (.off)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

New - Addition of 4 extra bytes now.
4 bytes of code - hex - 1st byte is clip series for the progression of erotic levels - New
(c0148_sc4_33503.vghd) the first byte would be 04 from the sc4

Repeat CLIPLOOP: for all clips in this card
Repeat CARDLOOP: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 05 01 00 00
all hex numbers are in 4 byte order
(hex numbers are low byte high byte ordered, read L2 L1 H2 H1 reorder to H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 05 01 00 00
read in 4 bytes of code = convert to number = Total Number of CARDS

*Card loop - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes and convert to a number = Length of Date (L)
Read in (L) Bytes = DATE in ASCII - when card was added to the collection yyyy-mm-dd
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**Clip loop - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes
Read 4 Bytes = 1st byte = Progression Series number
End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.
****
********** END OF NEW VERSION 1.0.5.x to 1.0.6.x **********
****

****
********** OLD VERSION **********
****

models.lst structure Old Version 04 01 00 00 and below..

start of file
4 bytes of code ? (possibly a version number ?)
4 bytes of code - hex - total number of cards in all collections, purchased or not, in the Models folder.
(hex numbers are low byte high byte ordered, read L2 L1 H2 H1 reorder to H1 H2 L1 L2)

CARDLOOP:
loop for number of cards
Start of cards
4 bytes of code - hex - length of card name a0001 = 5
next bytes - this will be the card name, use the hex above to determine how many bytes to read.
4 bytes of code - hex - length of Model name(s)
next bytes - this is the model(s) name - number of bytes from hex above
4 bytes of code - hex - length of Outfit name
next bytes - this is the outfit name
4 bytes of code - hex - length of the date
next bytes of code - this is the date the card was added to your collection yyyy-mm-dd
4 bytes of code - card status - See *** Status codes only first two bytes used
4 bytes of code - hex - Sum Total in bytes of total size of all clips for this card
4 bytes of code - hex - number of clips for this card set including demo clips

CLIPLOOP:
using clip (a0001_64412.vghd) for all the examples.
loop for number of clips in card
Start of Clips
4 bytes of code - hex - length of clip file name
next bytes - this is the clip filename - number of bytes from above (a0001_64412.vghd)
New - clips unchecked in the VIP clip view will be (a0001_64412.vghd.off) 4 extra bytes added to the end (.off)
4 bytes of code - hex - last two digits of the clip filename then CLIP NUMBER (0c hex or 12 dec)
4 bytes of code - hex - the clip level of explicitness from 00 to 05 (04 hex)
4 bytes of code - hex - bit sum for type of clip. add 80hex or 128dec for non-demo clip
(type 64 = c0 hex or 192 dec)
2nd byte is a 01 for new clip style (progressive hotness) - New
4 bytes of code - hex - Actual size of clip file in bytes

Repeat CLIPLOOP: for all clips in this card

Repeat CARDLOOP: for all cards in all collections, or until end of file is found.

How to read the Models.lst file for version 04 01 00 00
all hex numbers are in 4 byte order
(hex numbers are low byte high byte ordered, read L2 L1 H2 H1 reorder to H1 H2 L1 L2)
(H1 * 2^24) + (H2 * 2^16) + (L1 * 2^8) + (L2 * 2^1)
(H1 * 16,777,216) + (H2 * 65,536) + (L1 * 256) + L2
Largest number is - ff ff ff ff = 4,294,967,295

Open the models.lst file in binary read only

read in 4 bytes of code = version 04 01 00 00
read in 4 bytes of code = convert to number = Total Number of CARDS

*Card loop - Repeat this loop for total number of cards
Read 4 Bytes and convert to a number = Length of card name (L)
Read in (L) Bytes = Card name in ASCII
Read in 4 Bytes and convert to a number = Length of Model name (L)
Read in (L) Bytes = Model name in ASCII
Read in 4 Bytes and convert to a Number = Length of Outfit (L)
Read in (L) Bytes = Outfit in ASCII
Read in 4 Bytes and convert to a number = Length of Date (L)
Read in (L) Bytes = DATE in ASCII - when card was added to the collection yyyy-mm-dd
Read in 4 Bytes of Code = Card Status - See Status code description
Read in 4 Bytes and convert to a Number = Total Size of all clips for this card in Bytes.
Read in 4 Bytes and convert to a Number = How man Clips there are for this card
Including Demo clips.

**Clip loop - Repeat for total number of clips in this card
Read 4 Bytes and convert to a Number = Length of Clip Name (L)
Read (L) Bytes of Code = Clip Name
Read 4 Bytes = 1st byte = Clip Number (last 2 digits of clip name)
Read 4 Bytes = 1st byte = Erotic Level of Clip
Read 4 Bytes = 1st byte = Type of Clip (128 is added to Purchased clips)
2nd byte = 01 for new series with progression
Read 4 Bytes and convert to a number = Size of clip in Bytes

End of Clip Loop

End of Card Loop

End of Models.lst file
Close the File.

****
********** END OF OLD VERSION **********
****

****
********** CARD STATUS BITS Version 1.0.4.x to 1.0.6.x **********
****

version 1.1.0.0 added some new bits
I haven't taken the time to decipher their meaning yet.

*** Card Status codes described.
first two bytes of the 4 are used

a1a2 b1b2 c1c2 d1d2

	a1 a2 b1 b2
a1 = 0 not checked, full shows not downloaded	bits 0000 aaaa bbbb bbbb
a1 = 8 checked, full shows not downloaded	bits 1000 aaaa bbbb bbbb
a1 = 1 not checked, full shows downloaded	bits 0001 aaaa bbbb bbbb
a1 = 9 checked, full shows downloaded	bits 1001 aaaa bbbb bbbb
a1 = 2 being downloaded- blacked out	bits 0010 aaaa bbbb bbbb
a1 = 4 ?	bits 0100 aaaa bbbb bbbb
a2 = 0 demos not downloaded	bits aaaa 0000 bbbb bbbb
a2 = 8 demos downloaded	bits aaaa 1000 bbbb bbbb
a2 = 3 Free demos not downloaded and Delete?	bits aaaa 0011 bbbb bbbb
a2 = 1 free demos	bits aaaa 0001 bbbb bbbb
a2 = 2 Deleted	bits aaaa 0010 bbbb bbbb
a2 = 4 ?	bits aaaa 0100 bbbb bbbb
b1 = 0 not purchased	bits aaaa aaaa 0000 bbbb
b1 = 4 purchased	bits aaaa aaaa 0100 bbbb
b2 = 0 not favored	bits aaaa aaaa bbbb 0000
b2 = 4 selected for favorites	bits aaaa aaaa bbbb 0100

examples:

80 40 00 00 = need demos and full shows, checked in collection

88 40 00 00 = checked, demos downloaded, purchased, not favored

98 44 00 00 = checked, full shows downloaded, demos downloaded, purchased, on favorites list

18 40 00 00 = not checked, full shows downloaded, demos downloaded, purchased, not favored

Note - Possible bug in VGHD software
When Full Card is Downloaded before the Demos / trailers, a2 bit 3 is set, along with a1 bit 1
a2 bit 3 should not be set until the Demos / trailers are downloaded.

If a card has an error during download, reset the 4 byte pattern to 00 40 00 00
Shows not downloaded, demos not downloaded, owned, not checked, not favorite

If Shows and Demos ARE fully downloaded set the pattern to 18 40 00 00

If Demos are downloaded but full shows are needed 08 40 00 00

If Full shows are downloaded but demos needed 10 40 00 00

****
********** END CARD STATUS BITS **********
****

Known Clip types:

acccc_**0**Enn.vghd
acccc_**1**Enn.vghd
acccc_**2**Enn.vghd
acccc_**3**Enn.vghd --> duo One In front, one In back of task bar Only One clip.
acccc_**4**Enn.vghd
acccc_**16**Enn.vghd --> with Sign in Card a0001 only
acccc_**17**Enn.vghd --> with Sign not yet used
acccc_**18**Enn.vghd --> with Sign not yet used
acccc_**20**Enn.vghd --> with Sign not yet used
acccc_**32**Enn.vghd
acccc_**33**Enn.vghd
acccc_**34**Enn.vghd
acccc_**64**Enn.vghd
acccc_**66**Enn.vghd
acccc_**68**Enn.vghd
acccc_**80**Enn.vghd --> with Sign not yet used
acccc_**81**Enn.vghd --> with Sign not yet used
acccc_**82**Enn.vghd --> with Sign not yet used
acccc_**84**Enn.vghd --> with Sign not yet used
acccc_**96**Enn.vghd
acccc_**98**Enn.vghd
acccc_**100**Enn.vghd
acccc_**320**Enn.vghd --> by error

The type is the Sum of the following bits.

Bits:	value	IF SET IT MEANS
0	1	On the Taskbar – In front
1	2	Behind the Task bar
2	4	Pole Dance
3	8	reserved
4	16	Has a Sign
5	32	With an accessory
6	64	Enter / Exit
7	128	Reserved

A Value of 0 means Full View with No Enter / Exit

Enn?
E = Explicit level – Levels are from 0 to 5
0 = all audience – never used…
1 = Light Sexy
2 = Topless
3 = Full Nudity / Bottomless
4 = Explicit
5 = X rated

nn = clip number of that Explicit Level.

****
********** DATE CODES VERSION 1.0.8.x and up **********
****

How to figure out the models.lst date code:

For VG Models.lst version 1.0.6.4 and up use this formula:
The date the card was added to your collection 00 25 HH LL - 00 25 79 25 = 10 2011 11
Only the First two bytes are used. Total number of days past 11/13/1926
- birth of Max Vernon Mathews - Father of Computer / Digital Music
7925 hex = 31,013 days added to 11/13/1926 = 10/11/2011

****
********** END OF DATE CODES VERSION 1.0.8.x **********
****

Other interesting Date code formulas – NOT USED BY VG…

What Day of the week is it?

Uspensky and Heaslet gave the following formula, in Elementary Number Theory, 1939, for the Gregorian calendar:

W=D+floor(2.6m-0.2)+y+floor(y/4)+floor(c/4)-2c (mod 7)
Apparently, no tables are needed. Floor() means to drop the fraction; floor(6.8) is 6.
m is the month starting with March as 1 and ending with February as 12.
January and February of any year are considered to be in the previous year.

Other
W=C+Y+L+M+D (mod 7)

Where:

W is the day of the week (0=Sunday, through 6=Saturday)
C is a code for the century from this table (for the Gregorian calendar only):

1400s, 1800s, 2200s	2	1800 is not a leap year
1500s, 1900s, 2300s	0	1900 is not a leap year
1600s, 2000s, 2400s	5	2000 is a leap year
1700s, 2100s, 2500s	4	2100 is not a leap year

Y is the last two digits of the year.
L is the number of leap days since the beginning of the century. Step1: Divide the year (two digits) by 4 and throw away the fraction. Step 2: Notice that 1900 and 1800 were not leap years, and 2000 was. Only century years divisible by 400 are leap years. So, add 1 for those centuries divisible by 4 (as we haven't counted the leap day for year 00 yet). Step 3: Also, don't count a leap day if it happens after the date that you are calculating. In other words subtract one, if you are calculating a date of January or February of a leap year.
M is the code for the month, from this table:

1. Jan. 0	5. May 1	9. Sep. 5	
2. Feb. 3	6. June 4	10. Oct. 0	
3. Mar. 3	7. July 6	11. Nov.	3
4. Apr. 6	8. Aug. 2	12. Dec.	5

D is the date.

Here is a formula for finding the day of the week for ANY date.

$$N = d + 2m + [3(m+1)/5] + y + [y/4] - [y/100] + [y/400] + 2$$

where d is the number or the day of the month, m is the number of the month, and y is the year. The brackets around the divisions mean to drop the remainder and just use the integer part that you get.

Also, a VERY IMPORTANT RULE is the number to use for the months for January and February. The numbers of these months are 13 and 14 of the PREVIOUS YEAR. This means that to find the day of the week of New Year's Day this year, 1/1/98, you must use the date 13/1/97. (It sounds complicated, but I will do a couple of examples for you.)

After you find the number N, divide it by 7, and the REMAINDER of that division tells you the day of the week; 1 = Sunday, 2 = Monday, 3 = Tuesday, etc; BUT, if the remainder is 0, then the day is Saturday, that is: 0 = Saturday.

As an example, let's check it out on today's date, 3/18/98. Plugging the numbers into the formula, we get;

$$N = 18 + 2(3) + [3(3+1)/5] + 1998 + [1998/4] - [1998/100] + [1998/400] + 2$$

So doing the calculations, (remember to drop the remainder for the divisions that are in the brackets) we get;

$$N = 18 + 6 + 2 + 1998 + 499 - 19 + 4 + 2 = 2510$$

Now divide 1510 by 7 and you will get 358 with a remainder of 4. Since 4 corresponds to Wednesday, then today must be Wednesday.

Little Endian and Big Endian byte ordering:

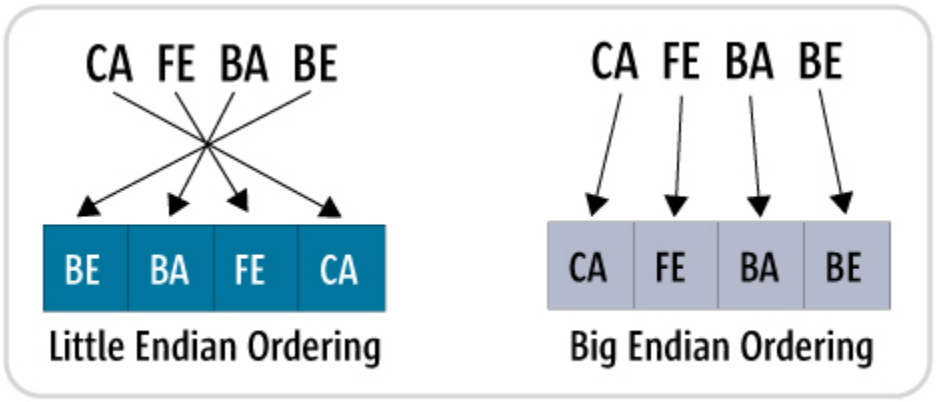

Figure 2 Byte ordering