

## Large File Playback Issues with Rapsody N35

Various people report about problems with playback of large video files when using the N35's PC network functions based on the smb/cifs protocol. As Rapsody/Dacos is not stating what cifs/smb client implementation they use, I did some brief analysis of the messages the N35 is sending to the smb/cifs server in order to get video data over the network.

### **The smb/cifs client implementation of the N35 (FW 1.26) has a hard limit at 4GB**

When you set your linux samba server to debug level 5, you can see that the N35 is requesting data from the server by SMBread messages. The documentation of Microsoft for the smb/cifs protocol (<http://www.jalix.org/ressources/reseaux/nfs-samba/samba/~smb/SMBPUB.DOC>) explains the parameters of the read message as follows:

#### **READ: Read File**

The read message is sent to read bytes of a resource indicated by *Fid* in the SMB header.

Client Request	Description
UCHAR WordCount;	Count of parameter words = 5
USHORT Fid;	File handle
USHORT Count;	Count of bytes being requested
ULONG Offset;	Offset in file of first byte to read
USHORT Remaining;	Estimate of bytes to read if nonzero
USHORT ByteCount;	Count of data bytes = 0

*Count* is used to specify the requested number of bytes.

*Offset* specifies the offset in the file of the first byte to be read. Note that this offset is limited to 32 bits, so this client request is inappropriate for files having 64 bit offsets.

In the next step I did an analysis what kind of messages my windows xp machine is sending to the samba server when starting playback with windows media player. This time the reads were done by means of SMBreadX messages. This type of messages is explained in the documentation as follows:

## READ\_ANDX: Read Data

Client Request	Description
UCHAR WordCount;	Count of parameter words = 10
UCHAR AndXCommand;	Secondary (X) command; 0xFF = none
UCHAR AndXReserved;	Reserved (must be 0)
USHORT AndXOffset;	Offset to next command WordCount
USHORT Fid;	File handle
ULONG Offset;	Offset in file to begin read
USHORT MaxCount;	Max number of bytes to return
USHORT MinCount;	Min number of bytes to return
ULONG Reserved;	Must be 0
USHORT Remaining;	Bytes remaining to satisfy request
USHORT ByteCount;	Count of data bytes = 0

Large File Client Request	Description
UCHAR WordCount;	Count of parameter words = 12
UCHAR AndXCommand;	Secondary (X) command; 0xFF = none
UCHAR AndXReserved;	Reserved (must be 0)
USHORT AndXOffset;	Offset to next command WordCount
USHORT Fid;	File handle
ULONG Offset;	Offset in file to begin read
USHORT MaxCount;	Max number of bytes to return
USHORT MinCount;	Min number of bytes to return
ULONG Reserved;	Must be 0
USHORT Remaining;	Bytes remaining to satisfy request
ULONG OffsetHigh;	Upper 32 bits of offset
USHORT ByteCount;	Count of data bytes = 0

Server Response	Description
UCHAR WordCount;	Count of parameter words = 12
UCHAR AndXCommand;	Secondary (X) command; 0xFF = none
UCHAR AndXReserved;	Reserved (must be 0)
USHORT AndXOffset;	Offset to next command WordCount
USHORT Remaining;	Bytes remaining to be read
USHORT DataCompactionMode;	
USHORT Reserved;	Reserved (must be 0)
USHORT DataLength;	Number of data bytes (min = 0)
USHORT DataOffset;	Offset (from header start) to data
USHORT Reserved[5];	Reserved (must be 0)
USHORT ByteCount;	Count of data bytes
UCHAR Pad[];	
UCHAR Data[ DataLength];	Data from resource

If the negotiated dialect is NT LM 0.12 or later, the client may use the Large File version of the request. This version allows specification of 64 bit file offsets.

*MinCount* in the request is valid only if *Fid* refers to a named pipe. *MinCount* informs the server that at least *MinCount* bytes should be returned, if possible.

Instead of the N35's read messages this does support large files with a length above  $2^{32}$  (4 294 967 296) bytes = 4GB. Now it is clear that the current smb/cifs implementation of the N35 cannot address data in files beyond the 4GB mark.

As a result, playback of .iso files (which in case of uncompressed dvd rips have a lot more than 4 GB) is not fully supported. During playback of such a file over smb/cifs by the N35 you will see that playback of chapters beyond the 4GB mark is aborted and the player returns to the main menu of the dvd.

### ***Avis beyond 2GB cannot be played over smb/cifs***

While .iso files only seem to be impacted by the hard 4GB limitation and do not show any problem with sizes between 2GB and 4GB, avis and mpeg do have additional problems.

Any avi file with more than  $2^{31}$  (2 147 483 648) Bytes = 2GB will not be played by the N35. The N35 throws an error message, that this file format is not supported. At first, this message puzzles me. This very much indicates that the N35 avi playback software might unfortunately make use of signed integers for file positioning instead of unsigned integers and therefore the smb/cifs client limit of 4GB comes down to the 2GB mark. Taking a wrap around at the 2GB mark into account it is quite obvious that the N35 doesn't get the expected data, but some data from the beginning of the file. This might explain why the error messages says "file format not supported". As the N35 seems always to take a look at the end of the file before starting playback, the N35 aborts when you try to playback avis greater 2 147 483 648 Bytes (2GB).

### ***Mpegs beyond 2GB partly work, Mpegs beyond 4GB don't work***

The issues with mpegs are a bit more complicated. First of all, it looks like the N35 reads mpegs just from the beginning and doesn't look at the file end. Therefore, mpeg playback always starts ok.

On any mpeg larger 2GB a press of the info button on the remote doesn't display the correct file size (it is always 1.0 KB) and it doesn't give any information about the duration of the video. What bothers me more than this hick-up on the info page is that the goto function on the remote is not working. You cannot enter a value there, the file end time stamp is displayed as --:--:--.

As a result, the navigation in the mpeg is limited to fast forward and rewind. No jumps at all. Like the avi bug in the previous chapter this looks very much like a bug in the software where signed integers are used for some kinds of file positioning. However, at least you can playback mpegs larger >2GB as long as you just let them run through.

I made a test with am mpeg larger 4GB. Here the info page shows a file size of just a few hundred MB and duration of just a few minutes. Looks like a wrap around at 4GB so that the N35 thinks the video is not that long. Playback starts ok, but aborts after having passed the wrong calculated end time of the video.

### ***Closing Thoughts***

I think I proved that the smb/cifs client implementation of the N35 does not handle large files >4GB. What adds to this limitation is that avis and mpegs seem to suffer from a signed/unsigned integer problem in the sw layers above the csmb/cifs client. I hope the development team is able to address both problems in the near future. As a first step, it would be nice if Rhapsody/Dacos would confirm these bugs.

If anyone has different findings, please let me know (n35 at pasternak.net). As far as we cannot look into the sources it is always partly guesswork what exactly goes wrong... but this is another story...