

ReplayTV Network Service Specifications

Version 0.14

Ian Prest

`rns@prest.ca`

<http://rns.prest.ca>

January 9, 2002

Contents

1	Introduction	1
2	RNS Server Specifications	1
2.1	NTP Requests	2
2.2	HTTP Requests	2
2.2.1	Common HTTP Headers	2
2.2.2	Login: <code>login.pl</code>	4
2.2.3	Authentication: <code>getauth2.pl</code>	5
2.2.4	Get Television Providers: <code>getzipcode2.pl</code>	7
2.2.5	Get Channel Descriptions: <code>getheadend.pl</code>	9
2.2.6	Get Channel Guide: <code>getcg2.pl</code>	12
2.2.7	Time Zone Information: <code>gettimezone.pl</code>	16
2.2.8	Update Data Files: <code>getmirror2.pl</code>	17
2.2.9	Get Time of Next Call: <code>getnextcall.pl</code>	19
2.2.10	Get RNS Server Names: <code>getservers.pl</code>	20
2.2.11	Get Global Dialup Phone Numbers: <code>getglobaldial3.pl</code>	21
2.2.12	Get Local Dialup Phone Numbers: <code>getlocaldial2.pl</code>	23
2.2.13	Transfer System Log: <code>putsyslog.pl</code>	24
2.2.14	Logout: <code>logout.pl</code>	25

1 Introduction

This document details all of the traffic that is exchanged between a ReplayTVunit and the ReplayTV Network Service (RNS) servers. It is intended that this document be used to create “simulated” RNS servers for people who are, for whatever reason, unable to connect to the “real” RNS server provided by ReplayTV itself, or who are otherwise unsupported by ReplayTV.

This document covers only the 3.0 software release, and the ReplayTV 3000-series hardware. Different software releases or hardware series may act differently.

The author of this document is in no way affiliated with ReplayTV or SonicBlue. This document is the result of countless hours spent logging and reverse-engineering the data sent between ReplayTV units and the RNS server.

The information in this document is provided in the hope that it will somehow be useful, but no guarantee is made as to the correctness or usefulness of the information presented in this document. Specifically, the author will not be held liable for damages to any ReplayTV units caused by the use of information presented in this document.

Furthermore, the author accepts no responsibility for what is done with this document; the user of this information is fully responsible for his or her own actions.

2 RNS Server Specifications

A typical RNS session will resemble the following:

1. The ReplayTV will connect to the RNS server via NTP to set the clock.
2. The ReplayTV will make several connections to the RNS server via HTTP to get channel listings, guide information, etc. This involves the following steps:
 - (a) login and get authentication;
 - (b) get local television providers and the channel names/numbers they provide;
 - (c) get channel-guide information for all channels;
 - (d) get ReplayZone updates, system commands, messages, etc.;
 - (e) get/put MyReplayTV information;
 - (f) get time of next call;
 - (g) get new server names, phone numbers, shell commands to perform;
 - (h) get any software updates;
 - (i) logout

An “initial setup” connection session will involve only a subset of the above steps, and the steps may occur out of order.

The following sections explain the steps in detail.

2.1 NTP Requests

A ReplayTV box, immediately upon connection to a dial-up server, will attempt to contact the RNS server to set the clock, via the NTP protocol.

2.2 HTTP Requests

After the clock has been set via NTP, all remaining traffic uses the HTTP protocol. The following sections detail each request, in the order they occur during a session.

2.2.1 Common HTTP Headers

Each outgoing HTTP request shares a common set of HTTP headers:

X-Replay-Serial-Number: XXXXXXXXXXXXXXXXXXXX

Specifies the eighteen-character serial number for the ReplayTV unit.

Authorization: Basic Uk5TQmFzaWM6QTd4KjgtUXQ=

Unknown. A minimal RNS server can simply ignore it. It is interesting to note that this authorization data seems to be common to *all* ReplayTV units.

User-Agent: Replay/111301030

Specifies that it is a ReplayTV unit trying to connect to the server, along with the software version number.

Host: rns.replaytv.net

Specifies the name of the RNS server.

Accept-Encoding: gzip

Names the various encodings that the ReplayTV is willing to accept responses in. ReplayTV units currently support `text/plain` and `gzip` encodings. The server should only respond with gzipped data if the ReplayTV unit says it will accept gzipped data.

Each incoming response to an HTTP request will use some or all of the following header items:

Date: Sat, 03 Feb 2001 03:15:35 GMT

Specifies the date and time of the response.

Server: Apache/1.3.12 (Unix) mod_perl/1.23

Specifies the type of HTTP server in use.

Content-Length: 0

Specifies the length of any additional content after the header.

Content-Type: text/plain

Specifies the type of any additional content after the header.

Connection: close
Always close.

In addition, the first line of an incoming HTTP response will indicate the HTTP success or error code. See RFC 2068 for more details.

2.2.2 Login: login.pl

Example:

```
GET /cgi-bin/2.0/login.pl?  
    action=nightly&  
    attempts=1&  
    total=157 HTTP/1.1
```

Parameters:

action: indicates what dial-up action is being performed. This parameter has three possible values:

nightly: indicates that this is a normal nightly connection, used to get new channel-guide information;

SetupGlobal: indicates that this is a “global” setup connection to the toll-free number, used to find a set of phone numbers local to the user; and

SetupLocal: indicates that this is a “local” setup connection, used to get the initial channel listings after a provider has been selected.

attempts: an integer that indicates how many connection attempts have been made—including the current attempt—since the last successful connection; reset after every successful connection, and incremented after every failed connection attempt.

total: an integer that indicates how many successful connections have been made by this unit; this number is maintained separately for each “action” type.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

The incoming response will contain only header items described in Section 2.2.1, and contains no additional content. If the login was successful, a “200 OK” response will be generated.

2.2.3 Authentication: getauth2.pl

Example:

```
GET /cgi-bin/2.0/getauth2.pl?  
    serial=XXXXXXXXXXXXXXXXXXXX&  
    oem=0&  
    udate=973934292 HTTP/1.1
```

Parameters:

serial: the eighteen-character serial number of the ReplayTV unit.

oem: *Unknown.* It is hypothesized that this parameter will have a different value for each manufacturer—presumably ‘0’ for ReplayTV-branded units, and ‘1’ for Panasonic-branded units.

udate: *Unknown.* The value of this parameter seems to match part of the **If-None-Match** header line—see below. It doesn’t appear to vary between separate **getauth2** requests.

Request:

In addition to the header items described in Section 2.2.1, this request contains the following header items:

If-None-Match: "allok-2-973729063-973934292-1-1-1"

Unknown. The two large integers in the middle are obviously dates in *seconds-since-1970* format, and the second date seems to match the **udate** parameter of the request. The first date probably acts as an **If-Modified-Since** date.

The request contains no additional content.

Response:

If the authorization information for this unit has not changed since the **If-None-Match** date, a “304 Not Modified” response is generated, and the response will contain no additional data. Otherwise, a “200 OK” response is generated, and it will contain the following header items, in addition to the header items described in Section 2.2.1:

ETag: "allok-2-1005067000-973934292-1-1-1"

Unknown. This is obviously connected to the **If-None-Match** header line in the request, and as before, the two large integers in the middle are obviously dates. The second date seems to match the **udate** parameter. The first number is probably the date the authorization information was last modified.

Content-Type: text/plain

Specifies the type of any additional content after the header. For **getauth2** requests this is always **text/plain**.

The content that follows the HTTP headers appears to be a text file specifying shell commands to execute, typically to edit the ‘SysConfig/Auth’ registry key, and subkeys.

Remarks:

This request appears to be completely unrelated to the **Authorization** header lines that appear in all requests. Instead, it is likely related to the ‘registration’ of individual units with ReplayTV.

A minimal RNS server can simply reply with “304 Not Modified”.

2.2.4 Get Television Providers: getzipcode2.pl

Example:

```
GET /cgi-bin/2.0/getzipcode2.pl?
    zipcode=98052&
    country=US HTTP/1.1
```

Parameters:

zipcode: the zipcode of the end-user.

country: the country of the end-user. This parameter will only ever have the value “US”, though it may be possible to change it by changing the country in a hidden menu on the ReplayTV.

Request:

In addition to the header items described in Section 2.2.1, this request contains the following header items:

If-Modified-Since: Sat, 06 Jan 2001 18:06:06 GMT

Optional. The server will only send provider information if it has changed since the specified date. Otherwise, it responds with “304 Not Modified”

Response:

If the list of television providers has not changed since the **If-Modified-Since** date, a “304 Not Modified” response is generated, and the response will contain no additional data. Otherwise, a “200 OK” response is generated, and it will contain the following header items, in addition to the header items described in Section 2.2.1:

Last-Modified: Thu, 25 Jan 2001 17:59:17 GMT

Specifies the date the television provider information was last changed for this zipcode.

Content-Type: application/vnd.replay.zipcode

Specifies the type of any additional content after the header. For **getzipcode2** requests this is always **application/vnd.replay.zipcode**.

The content that follows the HTTP headers is a sequence of 124-byte records. Each record represents a single television provider, and has the following format:

Location	Name	Description
0 → 7	<i>UID</i>	Null-terminated ASCII string representing the unique identifier (UID) for the television provider. Pad with null characters.
8 → 63	<i>Name</i>	Null-terminated ASCII string that gives the textual description for the television provider. Pad with nulls.
<i>continued...</i>		

Location	Name	Description
64	<i>Type</i>	ASCII character that represents the type of television provider. Known types include ‘A’ for antenna-based providers, ‘C’ for cable systems, and ‘S’ for satellite-based providers.
65 → 67	Unknown	Fill with nulls.
68 → 123	<i>Params</i>	Null-terminated ASCII string specifying various parameters for the television provider. Pad with nulls. Known parameters are: <code>irb=:776:</code> Specifies the code set to be used by the IR Blaster. Replace ‘776’ with the code set appropriate for the provider.

Remarks:

The *Name* and *Params* items may not be as long as indicated; other, as yet unknown items may exist between *Name* and *Type*, and after *Params*. For this reason, make sure that all fields are padded with nulls.

Values of *Name* of up to 37 characters in length have been tested.

Even though the ReplayTV unit says it will accept gzipped data, the “real” RNS server does not compress provider information.

2.2.5 Get Channel Descriptions: getheadend.pl

Example:

```
GET /cgi-bin/2.0/getheadend.pl?  
    headend=NY60586&  
    country=US HTTP/1.1
```

Parameters:

headend: the *UID* of the television provider to get channel descriptions for. See Section [2.2.4](#).

country: the country of the end-user. This parameter will only ever have the value “US”, though it may be possible to change it by changing the country in a hidden menu on the ReplayTV.

Request:

In addition to the header items described in Section [2.2.1](#), this request contains the following header items:

If-Modified-Since: Sat, 06 Jan 2001 18:06:06 GMT

Optional. The server will only send channel descriptions information if it has changed since the specified date. Otherwise, it responds with “304 Not Modified”

Response:

If the list of channel descriptions has not changed since the **If-Modified-Since** date, a “304 Not Modified” response is generated, and the response will contain no additional data. Otherwise, a “200 OK” response is generated, and it will contain the following header items, in addition to the header items described in Section [2.2.1](#):

Last-Modified: Thu, 25 Jan 2001 17:59:17 GMT

Specifies the date the channel descriptions were last changed for this television provider.

Content-Type: application/vnd.replay.headend

Specifies the type of any additional content after the header. For **getheadend** requests this is always **application/vnd.replay.headend**.

The content that follows the HTTP headers contains a 36-byte header, followed by a sequence of 56-byte records.

The contents of the header are unknown. See below.

Each record represents a single television channel, and has the following format:

Location	Name	Description
0 → 3	<i>UID</i>	DWORD representing the unique identifier (UID) for the channel. On the “real” RNS server, these are the UIDs that Tribune Media Services (TMS) gives to channels; some TV-listings web sites use the same UIDs. However, any values can be used in a custom RNS server.
4 → 5	<i>Number</i>	WORD representing the channel number for the channel <i>for this provider</i> .
6	Unknown	Known values include 0x00, 0x44, 0x4c, and 0x58. Channels in the list are often duplicated, differing only by this value. Some or all of the known values may appear for the same channel. It is hypothesized that this value corresponds to the <i>Lineup</i> for a television provider. For example, 0x00 would represent the “normal” lineup, 0x44 would represent the “re-build” lineup, etc. More work needs to be done in this area. A minimal RNS server can get by using only 0x00.
7	<i>Type</i>	The type of channel. Known values include: 0x00 for antenna-based channels (e.g., networks) 0x01 for basic-cable channels (e.g., networks, CSPAN) 0x02 for extended-cable channels (e.g., ESPN, TSN) 0x03 for pay-TV channels (e.g., HBO, TMN) 0x04 for pay-per-view channels
8 → 23	<i>Name</i>	Null-terminated ASCII string representing the short name of this channel. Generally, this value is set to the call-letters for the station. Pad with nulls.
24 → 55	<i>Desc</i>	Null-terminated ASCII string representing the name of this channel. Can be more descriptive than with <i>Name</i> . Pad with nulls.

Remarks:

Even though the ReplayTV unit says it will accept gzipped data, the “real” RNS server does not compress channel descriptions.

While little is known about the 36-byte header, the following header will work for minimal RNS servers:

```
const char header[] = {
    0x00,0x00,0x00,0x01,  0x01,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,
```

```
    0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,  
};
```

Evidence suggests that some of the header information refers to the number of “lineups” (see above), or the channel types used. Until more is known about the header, it is recommended that only “basic cable” channel types be used, and that the unknown/lineup value is set to 0x00.

2.2.6 Get Channel Guide: `getc2.pl`

Example:

```
GET /cgi-bin/2.0/getc2.pl?  
    @20010204:  
    23679,980193731;  
    11326+947909;  
    10161-1092;  
    ...  
    11911-87230; HTTP/1.1
```

Parameters:

This request doesn't use "normal" CGI parameters in `name=value` format. Instead, the only parameter is one long string that uses the following syntax:

1. An '@' sign.
2. The date to get listings for, in `YYYYMMDD` format.
3. A colon (':').
4. One or more channel UID/date pairs, in the following format:
 - (a) The channel UID (see Section 2.2.5).
 - (b) A date/time, representing the "If-Modified-Since" time for the channel, in one of the following formats:
 - A comma (','), followed by an absolute date/time, in *seconds-since-1970* format.
 - A plus sign ('+') followed by an integer. The integer is the number of seconds to *add* to the date/time from the previous channel.
 - A minus sign ('-') followed by an integer. The integer is the number of seconds to *subtract* from the date/time from the previous channel.
 - (c) A semicolon (;).

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A "200 OK" response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

Content-Encoding: `gzip`

Specifies the content encoding. To allow for faster downloads, a RNS server should gzip the channel guide.

Content-Type: `application/vnd.replay.cg`

Specifies the type of any additional content after the header. For `getc2` requests this is always `application/vnd.replay.cg`.

The content that follows the HTTP header contains is a sequence of variable-length “channel” records. Each channel record has the following format:

Location	Name	Description
0 → 3	<i>Modified</i>	DWORD representing the modified date for the channel information to follow, in <i>seconds-since-1970</i> format.
4 → 7	<i>UID</i>	DWORD representing the unique identifier of the channel whose information is to follow.
8 → 11	<i>Len</i>	DWORD representing the length of all the show records to follow (<i>not</i> including the header).
12 → 12+ <i>Len</i> −1	<i>Shows</i>	A sequence of variable-length “show” records. See below.

Each variable-length show record represents a single show on the channel that contains it. The record has the following format:

Location	Name	Description
0 → 3	<i>Flags</i>	DWORD specifying flags for the show. Known flag values are: 0x00000020 indicates a movie. <i>EX</i> = 8. 0x00000040 <i>Unknown</i> . <i>EX</i> = 4. 0x00000800 indicates a TV-14 rating. 0x00001000 indicates a TV-G rating. 0x00002000 indicates a TV-MA rating. 0x00004000 indicates a TV-PG rating. 0x00008000 indicates a TV-Y rating. 0x00010000 indicates a TV-Y7 rating.
4 → 7	<i>Time</i>	DWORD specifying the date/time of the show, in <i>seconds-since-1970</i> format.
8 → 11	<i>UID</i>	DWORD specifying the unique identifier of the <i>channel</i> that the show plays on. Should match the UID of the containing channel record.
12 → 13	<i>Duration</i>	WORD specifying the duration of the show, in minutes
<i>continued...</i>		

Location	Name	Description
14 → 17	Unknown	A minimal RNS implementation can fill with nulls.
18 → 19	<i>Len</i>	Length of the entire show record, including this header. Each show record is padded to end on a four-byte boundary.
20 → 27	<i>StrLen</i> [8]	Each CHAR represents the length (including the null terminator) of the corresponding entry to follow.
28 → 28 + <i>EX</i> - 1	<i>ExtraData</i>	<i>Unknown</i> . The length of this field depends on the value of the <i>Flags</i> field, and may not exist at all. If the show is a movie (<i>Flags</i> &0x00000020), part of <i>ExtraData</i> contains the year of release, as well as the MPAA rating.
28 + <i>EX</i> → <i>Len</i> - 1	<i>StringData</i>	<p>Eight Null-terminated ASCII strings. The strings represent, in the order they occur:</p> <ol style="list-style-type: none"> 1. show title 2. episode title 3. show description 4. actors 5. guest actors 6. suzuki identifiers. This is a comma-separated list of integers, which are used by ReplayZones. For unknown reasons, each integer is known as a “suzuki-id.” This field can be left blank, but at the expense of ReplayZones functionality. 7. producer? Evidence suggests that one of the strings is a ‘producer’ field, and another is a ‘director’ field, but relative position hasn’t yet been determined. 8. director? See above. <p>Each string occurs immediately after the null terminator for the previous string. If a string is empty, its null terminator must still be present.</p> <p>Any extra space (padding to ensure the record ends on a four-byte boundary) should be filled with nulls.</p>

Remarks:

The RNS response will include shows for the 24-hour period beginning at 12:00 AM of the requested date. Since all times are in GMT, this means that—in the Eastern Time Zone (GMT-5)—the response will include shows beginning at 7:00 PM of the night *prior to* the requested date, or 8:00 PM during daylight savings.

More specifically, shows that *start* during this time period should be included. This may

mean that the entire 24-hours is not covered; a show that starts prior to 12:00 AM but end after 12:00 AM is not included. Likewise, some included shows might extend past 12:00 AM the next day.

The maximum length of the *StringData* strings is unknown. Strings up to 127 characters in length work well. Longer strings have been known to cause problems.

There's lots that remains unknown about the channel guide format, but a minimal RNS server can ignore a lot of it. All of the unknown flags and fields can be set to null.

2.2.7 Time Zone Information: `gettimezone.pl`

Example:

```
GET /cgi-bin/2.0/gettimezone.pl?  
    zipcode=14201&  
    country=US HTTP/1.1
```

Parameters:

zipcode: the zipcode of the end-user.

country: the country of the end-user. This parameter will only ever have the value “US”, though it may be possible to change it by changing the country in a hidden menu on the ReplayTV.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A “200 OK” response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

Content-Type: `text/plain`

Specifies the type of any additional content after the header. For `getzipcode` requests this is always `text/plain`.

The content that follows the HTTP headers is a simple timezone string, followed by a linefeed character (0x0A). The timezone string follows the format of the TZ environment variable on Unix systems. Some examples are:

EST5EDT Eastern Time

PST8PDT Pacific Time

Remarks:

It is clear that the ReplayTV stores time internally as GMT. This request is sent only during the initial setup to determine which timezone the ReplayTV is operating in.

2.2.8 Update Data Files: getmirror2.pl

Example:

```
GET /cgi-bin/2.0/getmirror2.pl?  
    name=rz-home HTTP/1.1
```

Parameters:

name: the name of the mirror directory to update (i.e., a directory under the /usr/mirror directory in the ReplayTV's file system)

Request:

In addition to the header items described in Section 2.2.1, this request contains the following header items:

If-None-Match: "974939366"

Unknown. This appears that this line works as an **If-Modified-Since** header line.

The request contains no additional content.

Response:

If file manifest for the specified mirror directory has not changed since the **If-None-Match** date, a “304 Not Modified” response is generated, and the response will contain no additional data. Otherwise, a “200 OK” response is generated, and it will contain the following header items, in addition to the header items described in Section 2.2.1:

ETag: "979678836"

Unknown. Presumably the date of the newest file in the mirror directory. This is actually a standard header tag—IIS 5.0 generates them—but the contents might be important to the Replay.

Content-Type: text/plain

Specifies the type of any additional content after the header. For **getmirror2** requests this is always **text/plain**.

The content that follows the HTTP headers is a text file called a *manifest*. The manifest contains a list of all the files and directories under the mirror directory, along with their modified date. Lines that start with a hash character (`#`) are comments, but the “real” RNS server always generates files that start with the following five lines (with the “files” and “bytes” values set appropriately):

```
#server rns.replaytv.net  
#server rns-1.replaytv.net  
#server rns-2.replaytv.net  
#files 23  
#bytes 129033
```

Each additional line in the manifest specifies one file or directory. Each line has the following format, with each field separated by a space character:

filetype: the type of the file; either an ‘f’ for a file, or a ‘d’ for a directory;

date: the modified date for the file, in *seconds-since-1970* format;

localname: the quoted name of the file in the ReplayTV’s file system (a local path, not an absolute path);

remotename: the quoted name of the file on the RNS server (an absolute path, but omitting the server name).

As an example, the following lines come from actual manifests:

```
d 968903945 "ads" "/mirror/amgr-grp0/ads"
d 979678836 "ads/pause" "/mirror/amgr-grp0/ads/pause"
...
f 977185311 "1008" "/mirror/messages/1008.gz"
f 975715158 "1007" "/mirror/messages/1007.gz"
```

Remarks:

This is a kind of catch-all request. The mirror directories on the ReplayTV store a lot of different information, including *ReplayZones* (each has its own directory), system messages, pause ads, and other system files that are updated frequently.

The ReplayTV will follow up on this request by requesting any new or modified files in the list. It compares the file time for its version of the file against the file time specified in the manifest to determine which files it needs to download.

If you examine the example above, you will notice that the remote files are gzipped, but the local names are not. The files on the server are compressed to save transfer time. After the ReplayTV has downloaded the file, it will uncompress it. In fact, it appears as if *every* mirrored file is compressed but, rather oddly, the *Content-Encoding* for the downloaded files is never specified.

In addition to the header items described in Section 2.2.1, the response for each additional file will contain the following header items:

ETag: "9b8334-68-39d3fb32"

Unknown. The final number in the **ETag** is a *seconds-since-1970* date, in hex. This number will match the modified date in the manifest. It is unclear what the other numbers represent.

For a minimal RNS server, the easiest way to handle `getmirror2.pl` requests would be to always respond with “304 Not Modified.” Assuming the ReplayTV has connected (at least once) to the real RNS server, it already has the required files for complete operation. Handling this request is complicated, and requires knowledge of various other ReplayTV file formats and the ReplayTV directory structure.

2.2.9 Get Time of Next Call: `getnextcall.pl`

Example:

```
GET /cgi-bin/2.0/getnextcall.pl?  
    time=7200,14400&  
    tz=EST5EDT HTTP/1.1
```

Parameters:

time: *Unknown*. This line seems to match the `time` line in the response.

tz: The timezone of the caller. See Section 2.2.7.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A “200 OK” response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

Content-Type: `text/plain`

Specifies the type of any additional content after the header. For `getnextcall` requests this is always `text/plain`.

The content that follows the HTTP header contains is text file that consists of exactly two lines, with each line terminated by a linefeed (0x0A) character. Here is an example from an actual response:

```
time=7200,14400  
wtime=3600,10;3600,25;3600,45;3600,20
```

The first line appears to match the `time` parameter that was passed as an argument to the `getnextcall` script. While it is not clear what the numbers represent, it is assumed that they are specified in seconds. 7,200 seconds equals two hours, and 14,400 seconds equals four hours. It is hypothesized that this means the call window begins at 2:00 AM, and extends for a total of four hours.

The meaning of the second line is also unknown. It is assumed that first number in each pair is, again, in seconds (3,600 seconds equals 1 hour). The second number from each pair always sum to 100, so it is hypothesized that they are *probabilities*, and that `wtime` stands for *weighted time*.

Remarks:

ReplayTV undoubtedly uses this request to randomize the call times for each unit, to help minimize server load. This could also, perhaps, be used to get each ReplayTV unit to dial the server more frequently.

A minimal RNS server can simply respond with the sample text, above.

2.2.10 Get RNS Server Names: `getservers.pl`

Example:

```
GET /cgi-bin/2.0/getservers.pl?  
    version=111301030&  
    serial=XXXXXXXXXXXXXXXXXXXX HTTP/1.1
```

Parameters:

version: The software version number. Note that this number matches the version number reported in the `User-Agent` string (see Section 2.2.1).

serial: The eighteen-character serial number of the ReplayTV unit.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A “200 OK” response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

`Last-Modified: Thu, 26 Oct 2000 14:58:40 GMT`
The date the list of servers was last updated.

`Content-Type: text/plain`

Specifies the type of any additional content after the header. For `getservers` requests this is always `text/plain`.

The content that follows the HTTP headers is a text file where each line specifies the name of one RNS server, in order of priority. Each line is separated by a linefeed character (0x0A).

Remarks:

In a “local” RNS server, where all HTTP traffic from the modem is intercepted and handled, the server names specified are unimportant. This request might prove useful if the goal is to act as a replacement RNS server for ReplayTV units which are not directly connected to the server itself.

2.2.11 Get Global Dialup Phone Numbers: getglobaldial3.pl

Example:

```
GET /cgi-bin/2.0/getglobaldial3.pl?  
    phone=716888&  
    zipcode=14201&  
    country=US&  
    all=1 HTTP/1.1
```

Parameters:

phone: the first six digits of the end-user's telephone number.

zipcode: the zipcode of the end-user.

country: the country of the end-user. This parameter will only ever have the value "US", though it may be possible to change it by changing the country in a hidden menu on the ReplayTV.

all: *Unknown*. Always '1'.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A "200 OK" response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

Content-Type: `text/plain`

Specifies the type of any additional content after the header. For `getglobaldial3` requests this is always `text/plain`.

The content that follows the HTTP headers is a text file where each line specifies one global (toll-free) dial-up phone number. Each line, delimited by linefeed characters (0x0A), has the following format:

number: The phone number string, which can include other characters besides digits (including '(', ')', '-', and spaces).

format: *Unknown*. A string indicating the display or dial format for the phone number. More than one format can be present, separated from each other (and from the **number** field) by a pipe character (0x7C).

name: The phone number's name, separated from the **format** field by a tab character (0x09). For global numbers, this is the two-character country code for which the phone number is valid.

auth: *Unknown.* An optional authentication string that tells the ReplayTV unit what password to use, and specifies additional authentication parameters, separated from the **country** field by a 0x1E character.

Here is an example of an actual RNS response, with the phone numbers removed for security reasons, and unprintable characters represented inside braces:

```
(800) xxx-xxxx|1-[1,3]-[6,8]| [1,3]-[6,8]{0x09}US{0x1E}login=%s@replaytv.net:pap=:1:
(877) xxx-xxxx|1-[1,3]-[6,8]| [1,3]-[6,8]{0x09}CA{0x1E}login=%s@replaytv.net:pap=:1:
xxxxxxxxxx| [0,10]{0x09}JP{0x1E}login=%s@replaytv.net:pap=:1:
xxxxxxxxxx| [0,10]{0x09}GB
01800xxxxxxxx| [0,12]{0x09}MX
```

Remarks:

It is interesting to note that while ReplayTV only supports the United States, many of the server requests have a **country** parameter. The actual response from the server for this request indicates that support for Canada, Mexico, Japan, and Great Britain (?) was at least taken into consideration when the software was written, and ReplayTV went so far as to acquire toll-free access numbers for these countries.

2.2.12 Get Local Dialup Phone Numbers: getlocaldial2.pl

Example:

```
GET /cgi-bin/2.0/getlocaldial2.pl?  
    phone=716888&  
    zipcode=14201&  
    country=US HTTP/1.1
```

Parameters:

phone: the first six digits of the end-user's telephone number.

zipcode: the zipcode of the end-user.

country: the country of the end-user. This parameter will only ever have the value "US", though it may be possible to change it by changing the country in a hidden menu on the ReplayTV.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A "200 OK" response is generated containing the following header items, in addition to the header items described in Section 2.2.1:

Content-Type: `text/plain`

Specifies the type of any additional content after the header. For `getlocaldial2` requests this is always `text/plain`.

The content that follows the HTTP headers is a text file where each line specifies one local dial-up phone number. The format is exactly the same as that for the global dial-up numbers (see Section 2.2.11), except that the `name` field specifies a city and state, rather than a two-character country code.

2.2.13 Transfer System Log: putsyslog.pl**Example:**

```
POST /cgi-bin/2.0/putsyslog.pl HTTP/1.1
```

Request:

Unknown. The contents of this request have not been analyzed, but it seems to contain information on time zone, cable providers, and modem status, among other things.

Response:

A “200 OK” response is generated containing only header items described in Section [2.2.1](#).

Remarks:

A minimal RNS server can simply ignore the content from the ReplayTV unit and respond with “200 OK.”

2.2.14 Logout: logout.pl

Example:

```
GET /cgi-bin/2.0/logout.pl?  
    action=SetupGlobal&  
    result=0 HTTP/1.1
```

Parameters:

action: indicates what dial-up action is being performed, as described in Section 2.2.2.

result: *Unknown*. A value of '0' seems to indicate success.

Request:

The outgoing request will contain only header items described in Section 2.2.1, and contains no additional content.

Response:

A "200 OK" response is generated containing only header items described in Section 2.2.1.