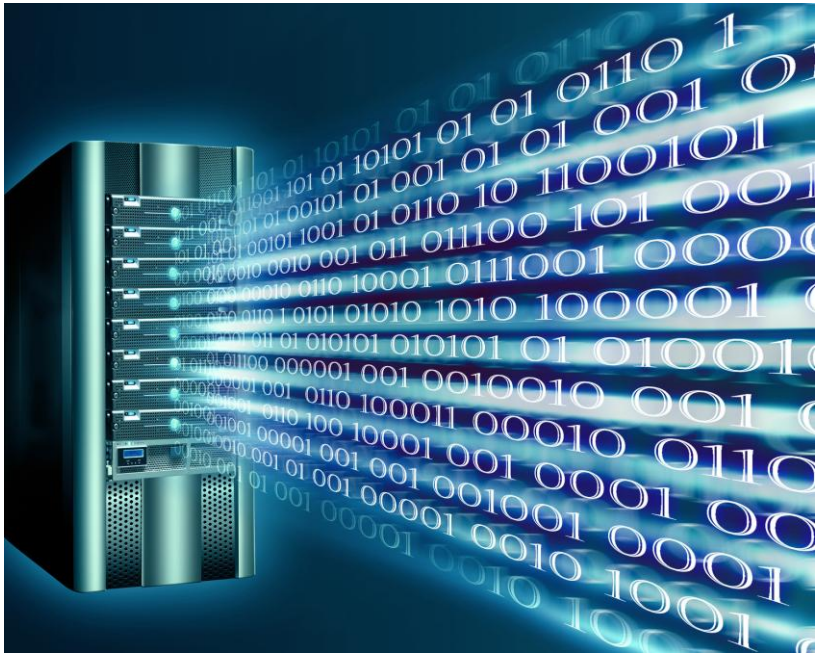
Ingenieurbüro David Fischer AG | A Company of the Apica Group
http://www.proxy-sniffer.com

Apica
ProxySniffer

Executing Load Tests with Google Protocol Buffer (protobuf) Messages

English Edition



© 2013 All Rights Reserved September 26, 2013

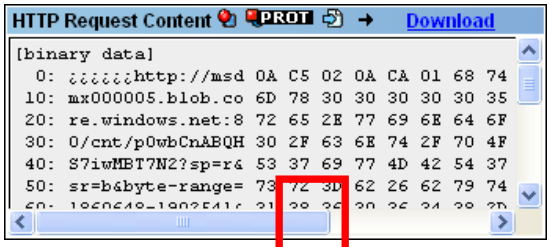
Table of Contents

1 Introduction 3
1.1 Full Data Integration 3
2 Postprocessing the Recorded Session 4
2.1 Automatic Detection of Protobuf Messages 4
2.2 Loading the Protobuf FileDescriptorSet (*.desc file) into ProxySniffer 5
2.3 Configuring the Message Type for all Protobuf Requests and Responses 7
2.4 Extracting and Assigning Variables from/to Protobuf Message Fields 12
3 Generating the Load Test Program and Executing the Load Test 16
4 Manufacturer, Sales and Support 17

1 Introduction

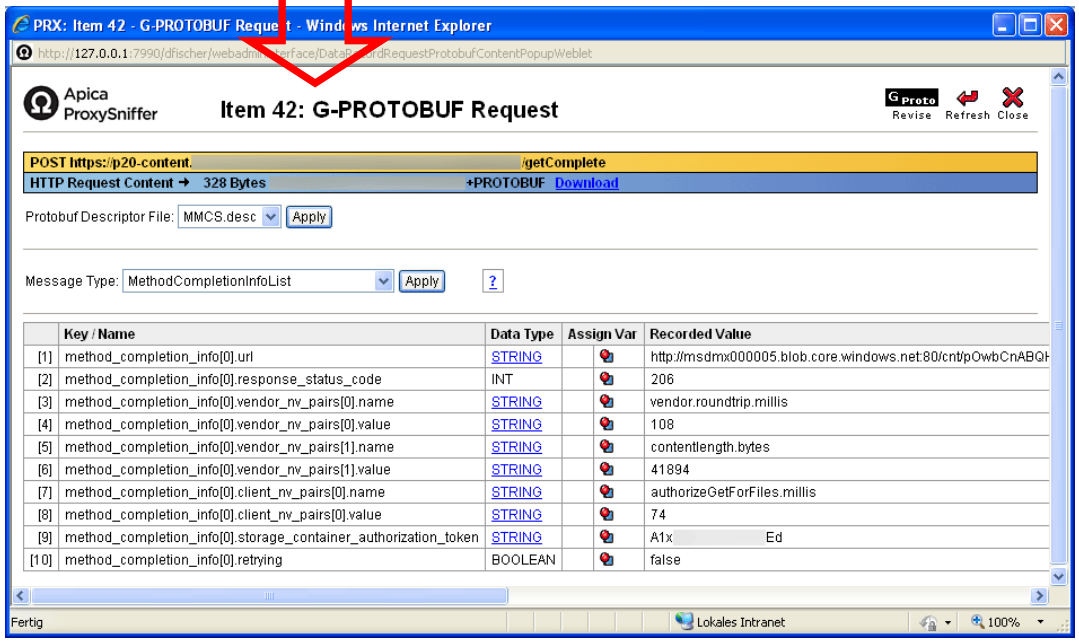
Wikipedia: *"**Protocol Buffers** are a method of serializing structured data, developed by Google Inc.*
The design goals for Protocol Buffers emphasized simplicity and performance. In particular, it was designed to be smaller and faster than XML. Protocol Buffers are widely used at Google for storing and interchanging all kinds of structured information. Protocol Buffers serve as a basis for a custom remote procedure call (RPC) system that is used for nearly all inter-machine communication at Google."

From a tester's view, Protocol Buffers (**protobuf**) is just another way of serializing data, like XML and JSON are. However, in opposite to XML, the "names" of the data fields and the name of the data record type (message type) are <u>not included</u> into the transmitted message, meaning that the (binary) bytes of the data field values are transmitted only – without any metadata information.



Therefore, when recording protobuf messages with ProxySniffer you will see in the GUI at a first step unstructured binary data only.

Starting from ProxySniffer version 5.1, you can load the metadata information for protobuf messages into ProxySniffer, by loading one or more Google Protobuf FileDescriptorSets (*.desc files). After that you have to select the corresponding "protobuf message type" and, starting from this point, you can handle and post-process protobuf messages in an easy and convenient way in ProxySniffer.



1.1 Full Data Integration

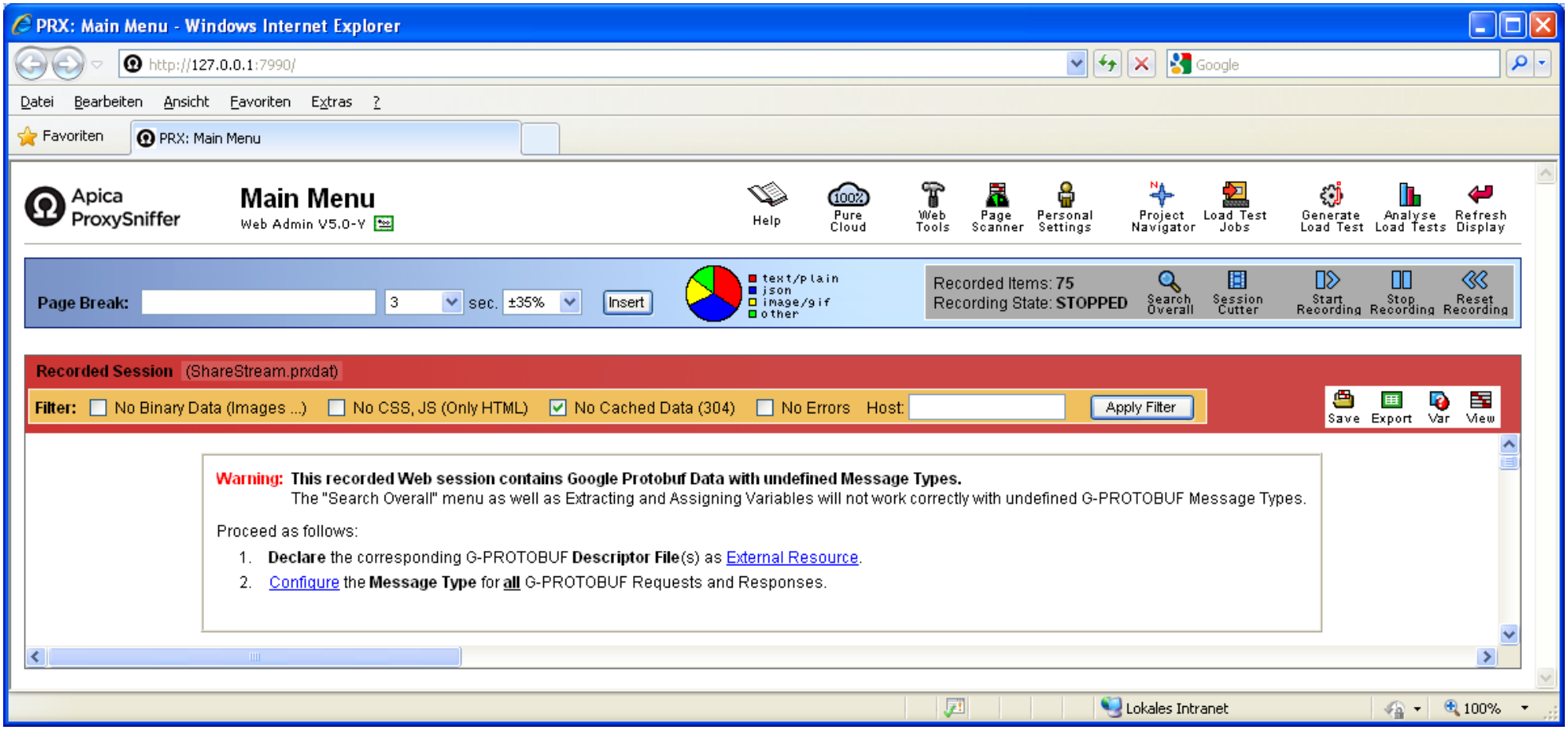
ProxySniffer provides full data integration for Protocol Buffers, including:

- Parsing the binary protobuf data and showing them in a human-readable view.
- Extracting and Assigning session variables from/to protobuf data fields, by interpreting and with protecting the binary protbuf message format.
- Searching any ASCII text-fragments over the whole recorded session – and overall error snapshots in the load test result – by interpreting and (temporary) converting the binary protobuf data fields to their corresponding ASCII text presentation.
- Automated distribution of the Protobuf FileDescriptorSets (*.desc files) to the load generators.

2 Postprocessing the Recorded Session

2.1 Automatic Detection of Protobuf Messages

After recording a session with ProxySniffer all recorded data are automatically checked if they contain protobuf data. This mechanism checks all recorded HTTP request and response and searches for a matching text fragment containing the string "protobuf" in all HTTP header fields named "Content-Type". If any match is found the following warning text is shown in the ProxySniffer GUI:



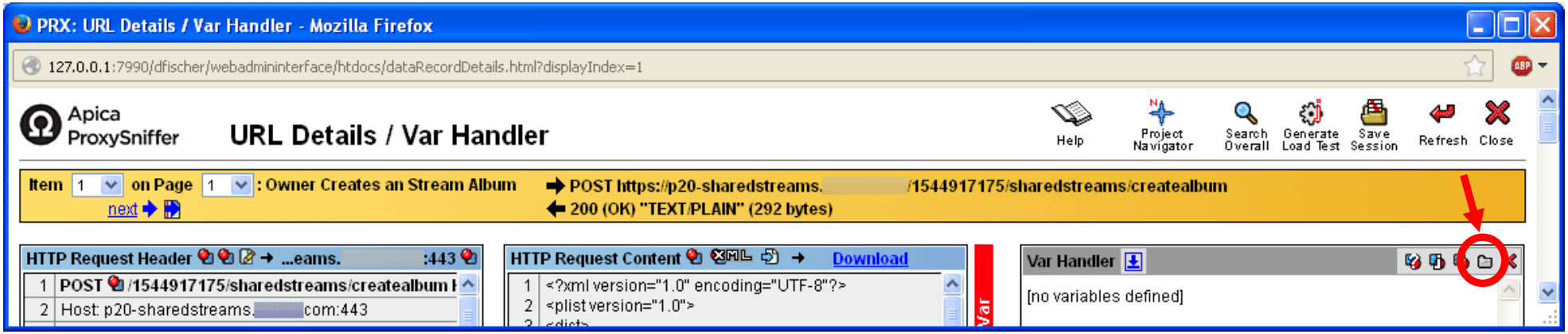
2.2 Loading the Protobuf FileDescriptorSet (*.desc file) into ProxySniffer

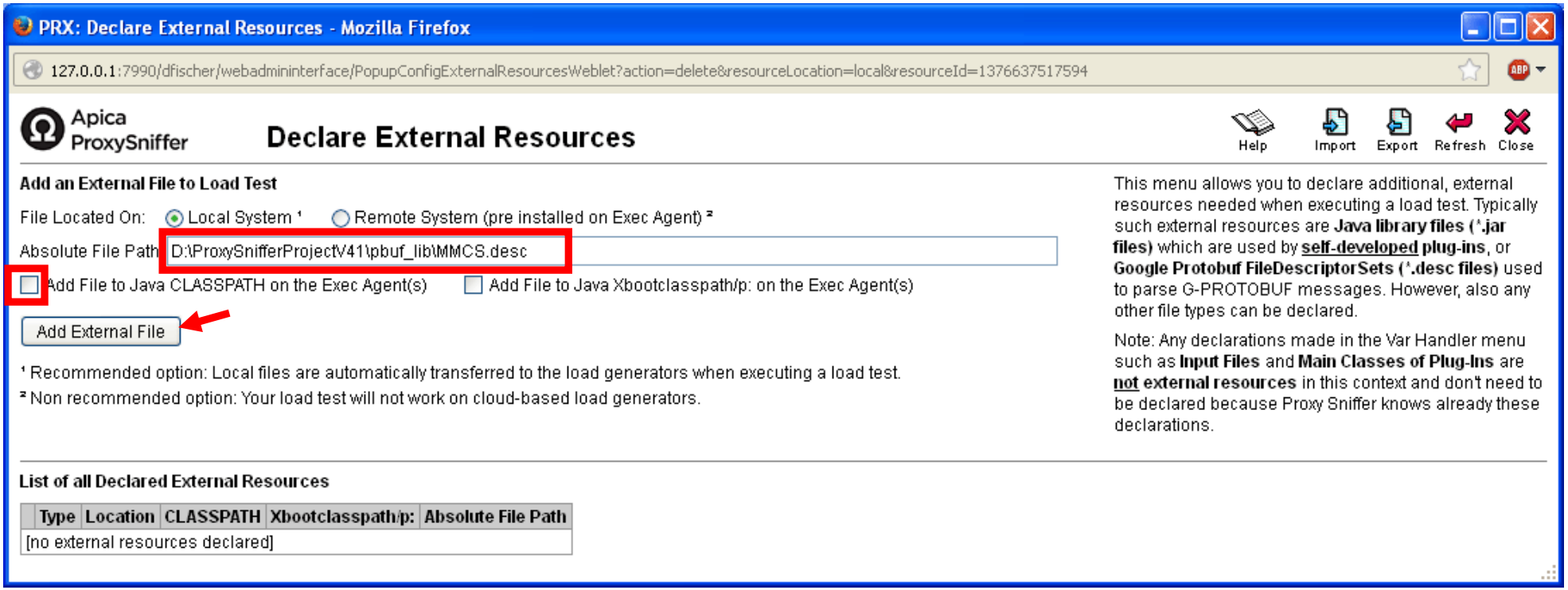
In order that ProxySniffer can interpret the binary protobuf data you have to load the corresponding Protobuf FileDescriptorSet (*.desc file) into ProxySniffer, which contains metadata information about the recorded protobuf messages. Such a protobuf *.desc file can be gained from a protobuf source file (*.proto) by using the **protoc** compiler. Example:

```
protoc --descriptor_set_out=addressbook.desc addressbook.proto
```

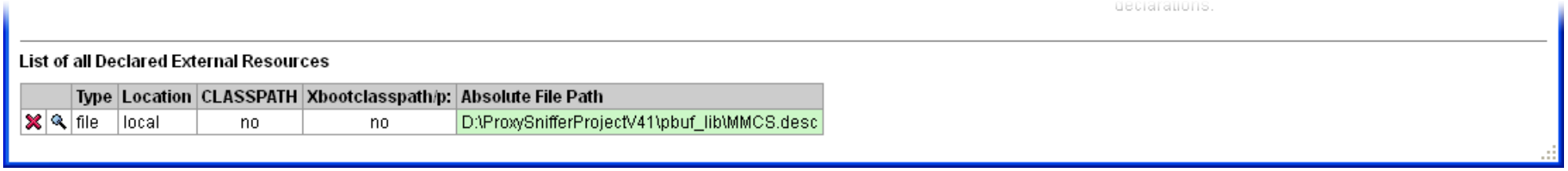
You have to use the proper *.proto file that matches to the corresponding recorded data. You may ask the development team of the Web application to get the proper *.proto file.

Once you have the protobuf *.desc file you can place it to any directory on your local machine and then register it as an "External Resource" of your recorded session by calling the "Declare External Resources" menu:



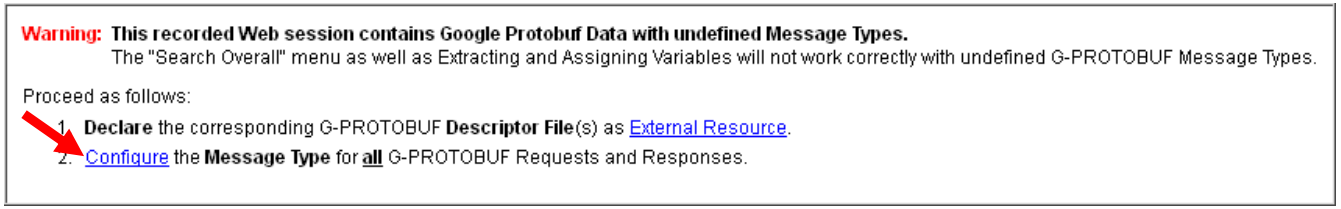


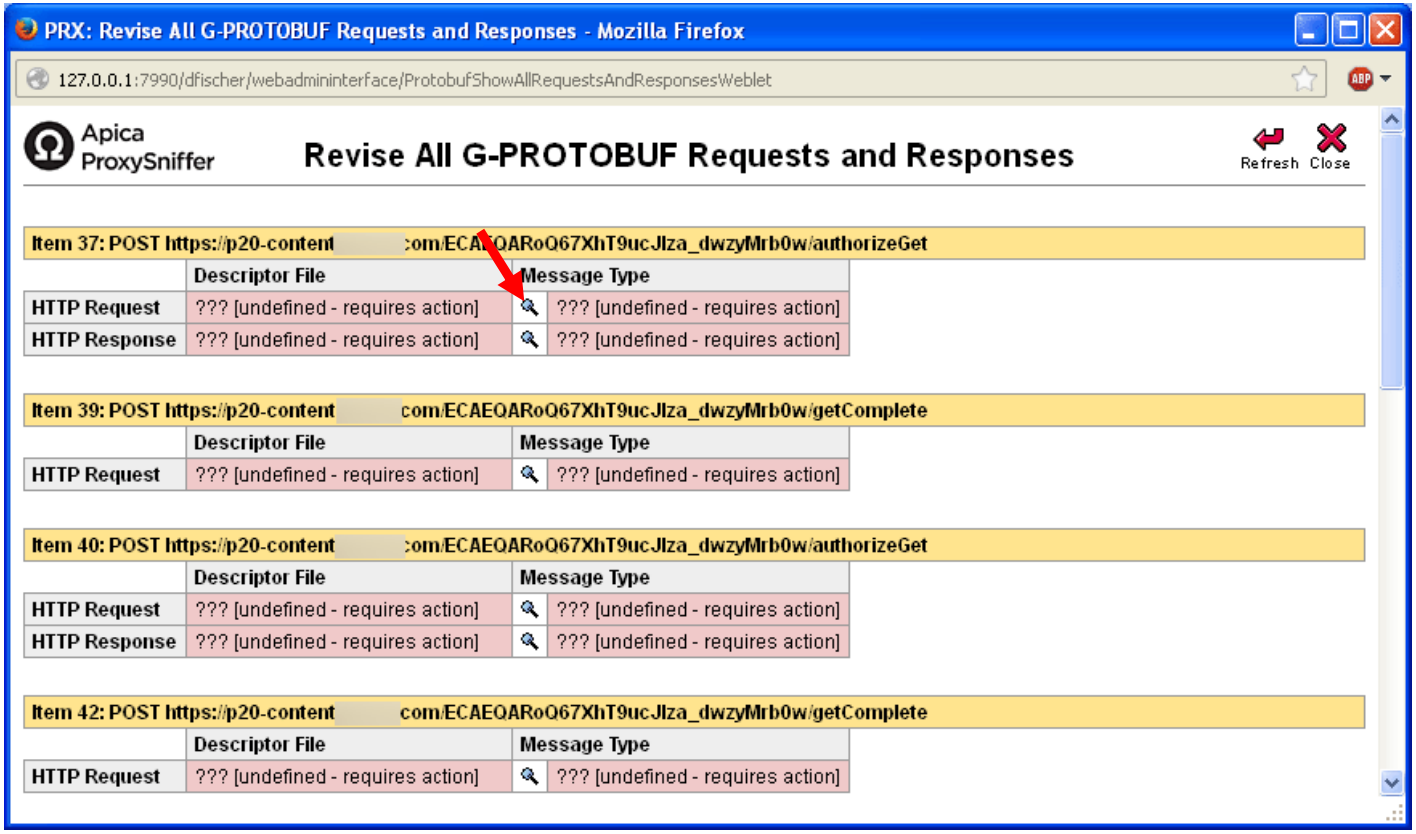
Note: you should also <u>uncheck</u> the checkbox "Add File to Java CLASSPATH on the Exec Agent(s)"



2.3 Configuring the Message Type for all Protobuf Requests and Responses

In order that the protobuf data can be shown in a human-readable view, and that you can extract and assign variables from/to protobuf messages you have to configure the proper "Message Type" for **all** Protobuf requests and responses. Return to the ProxySniffer main menu and click on the second link in the warning text:

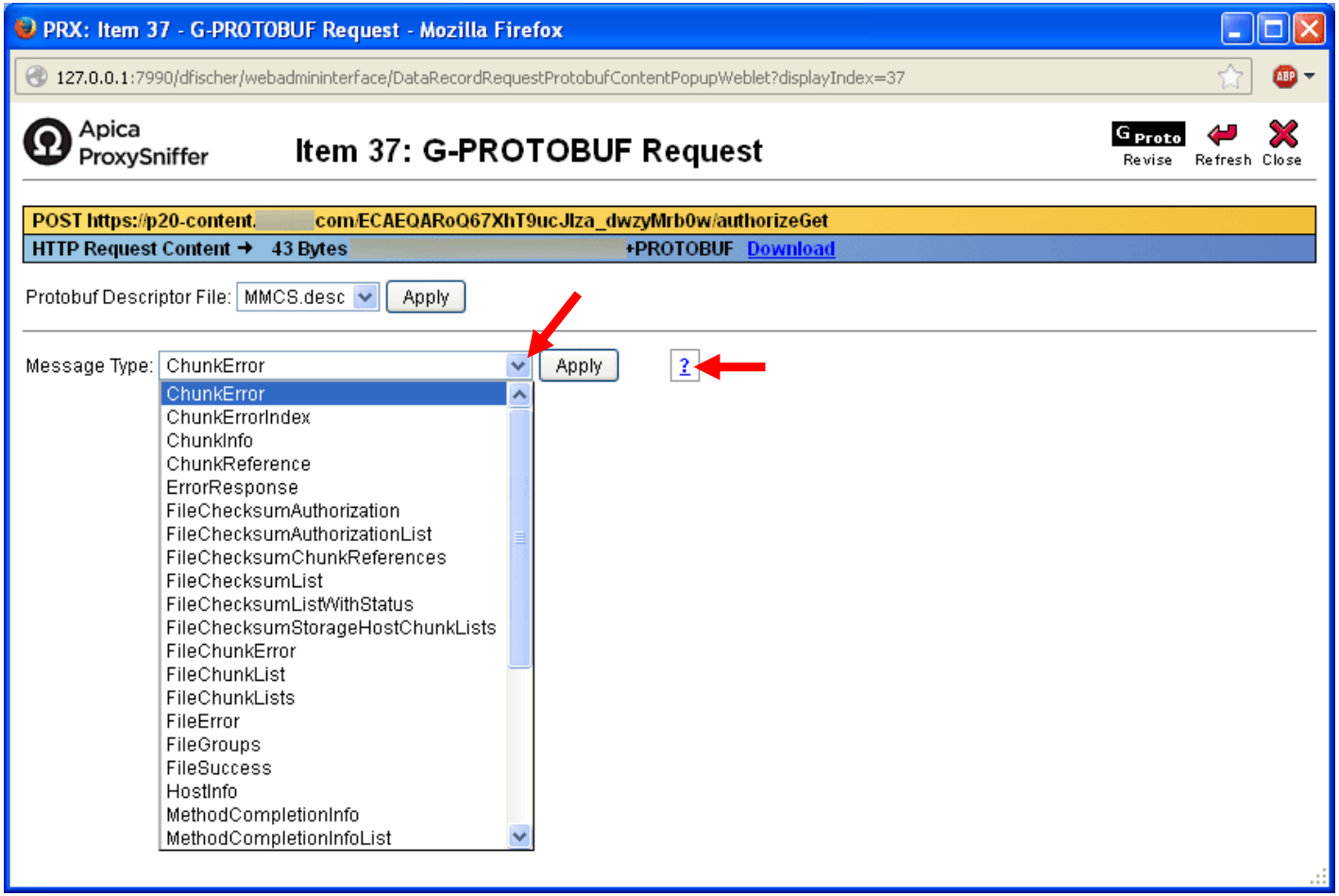




You will see a compilation of all protobuf messages over the whole recoded session.

Click on the magnifier icon of each message to define the "message type".

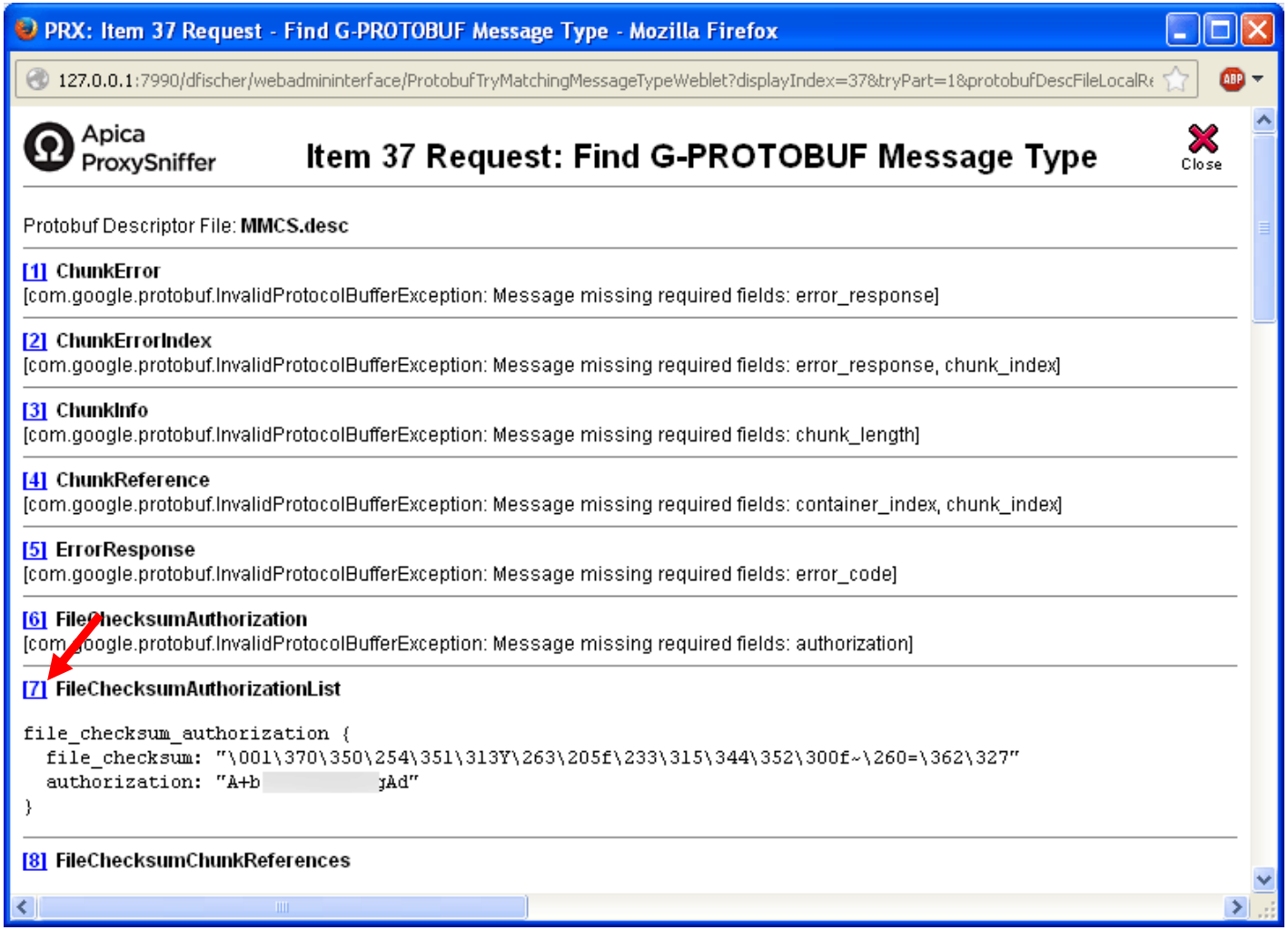
Then you can select the corresponding protobuf "message type" for this HTTP request or response:



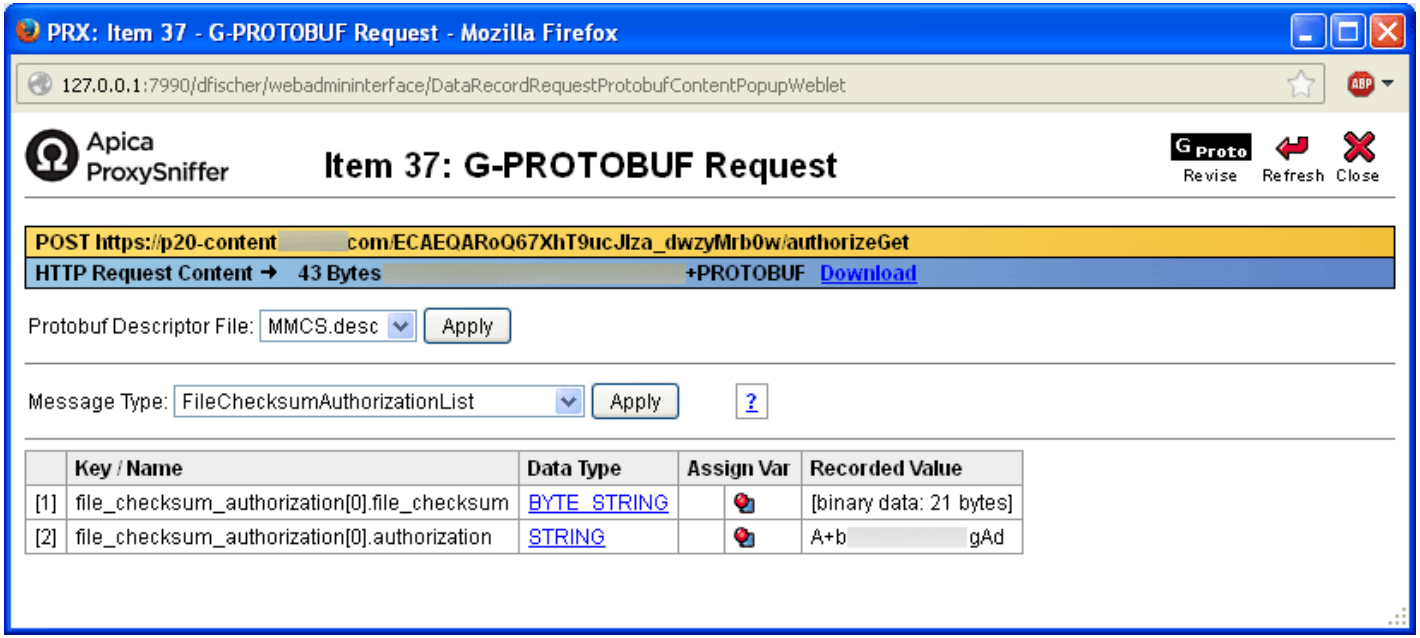
If you don't know which "message type" matches to the binary data you can try to guess it by clicking on the **?** icon.

However, instead of guessing, best practice is to ask the development team of the Web application for that information – for each HTTP request and response that contains protobuf data.

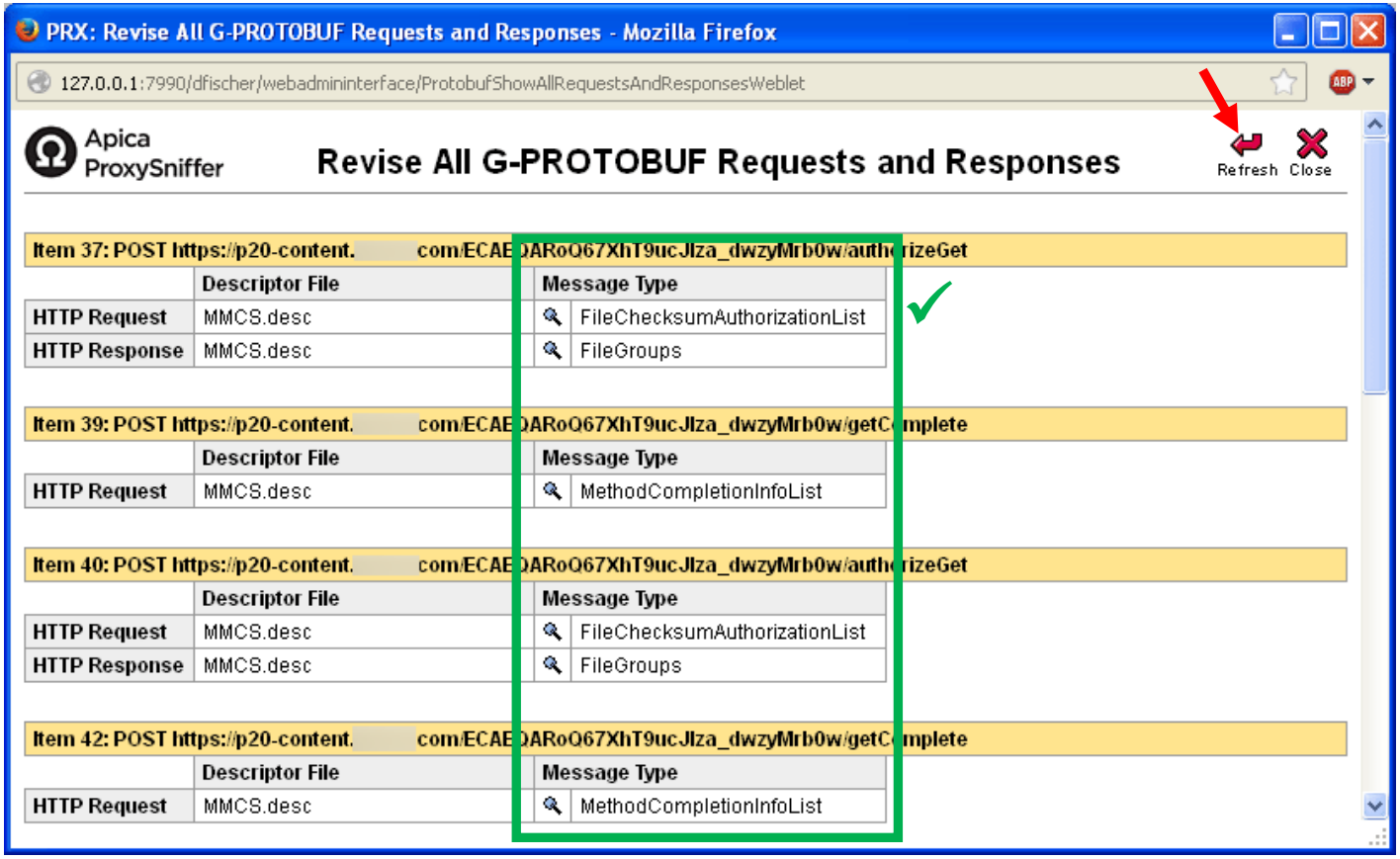
If you have to guess the "message type" then select the "parsed message-result" that makes the most sense:



After you have selected the "message type" you will see the parsed protobuf data:



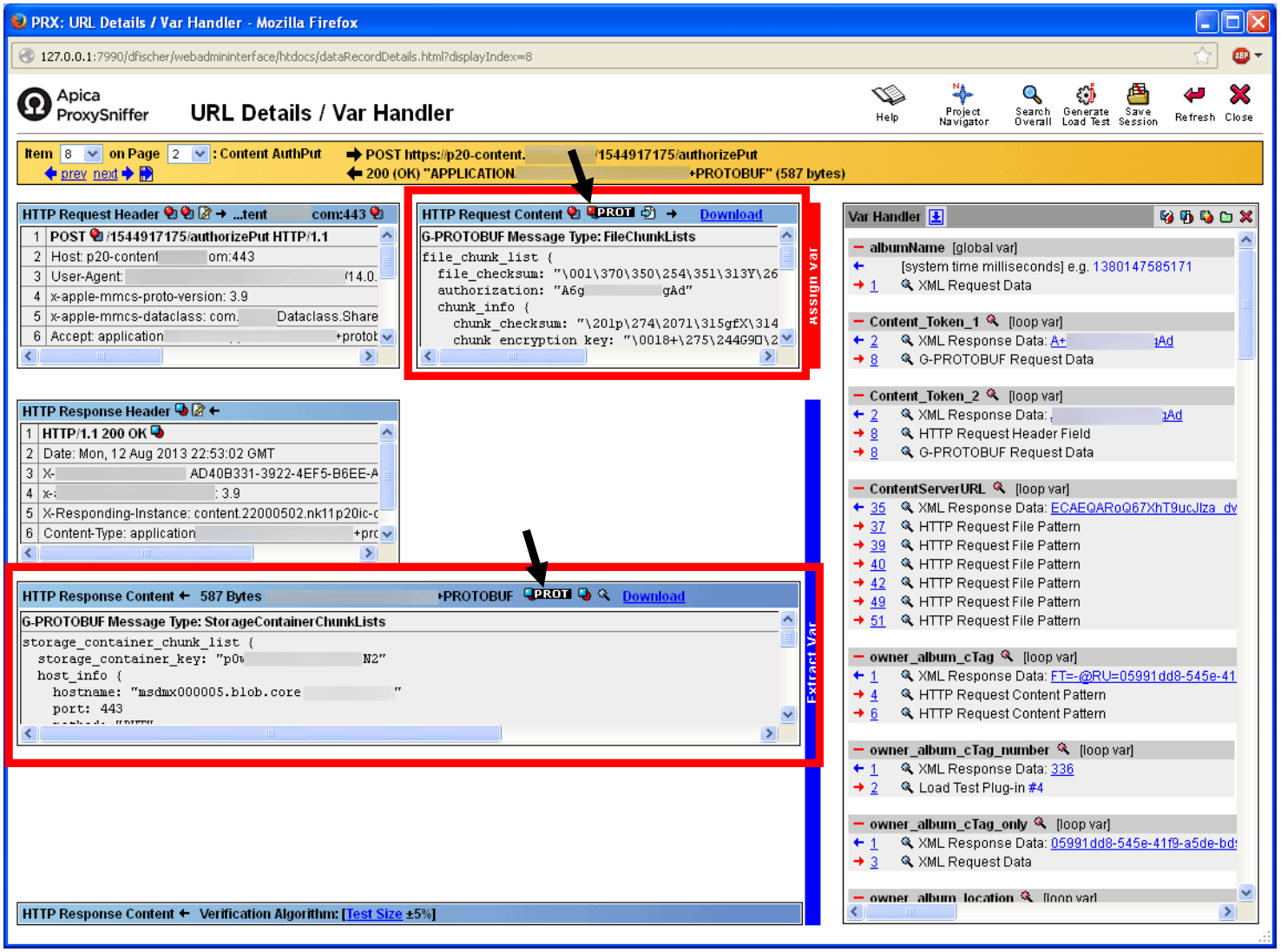
Return to the "Revise All G-PROTOBUF Requests and Responses" menu and then click on the "Refresh " icon. After that continue to complete the "message types" **for all other** HTTP requests and responses. Finally all "message types" should be defined:



Now you are ready to search any ASCII text fragments overall recorded data and you are ready to extract and assign variables from/to protobuf data fields. At this step it's recommended that you save your recorded session once again.

2.4 Extracting and Assigning Variables from/to Protobuf Message Fields

After you have defined the "message type" for all protobuf messages you can extract and assign variables from/to protobuf message fields – in a standard and convenient way as provided for all other supported data formats:



PRX: Item 8 - G-PROTOBUF Response - Mozilla Firefox

127.0.0.1:7990/dfischer/webadmininterface/DataRecordResponseProtobufContentPopupWeblet?displayIndex=8

Apica ProxySniffer

Item 8: G-PROTOBUF Response

G Proto Revise Refresh Close

POST https://p20-content[redacted]m/1544917175/authorizePut

HTTP Response Content ← 587 Bytes [redacted] PROTOBUF Download

Protobuf Descriptor File: MMCS.desc Apply

Message Type: StorageContainerChunkLists Apply ?

	Key / Name	Data Type	Extract Var		Recorded Value
[1]	storage_container_chunk_list[0].storage_container_key	STRING			pO[redacted]N2
[2]	storage_container_chunk_list[0].host_info.hostname	STRING	vendor_hostName		m[redacted]
[3]	storage_container_chunk_list[0].host_info.port	INT			443
[4]	storage_container_chunk_list[0].host_info.method	STRING			PUT
[5]	storage_container_chunk_list[0].host_info.uri	STRING	vendor_host_uri		/cnt/pOwl[redacted]IT7N2?sp=w&sr=b&se=2013-08-1
[6]	storage_container_chunk_list[0].host_info.transport_protocol	STRING			HTTP
[7]	storage_container_chunk_list[0].host_info.transport_protocol_version	STRING			1.1
[8]	storage_container_chunk_list[0].host_info.scheme	STRING			https
[9]	storage_container_chunk_list[0].host_info.headers[0].name	STRING			x-ms-client-request-id
[10]	storage_container_chunk_list[0].host_info.headers[0].value	STRING			AD4[redacted]\0F
[11]	storage_container_chunk_list[0].host_info.headers[1].name	STRING			x-ms-blob-type
[12]	storage_container_chunk_list[0].host_info.headers[1].value	STRING			BlockBlob
[13]	storage_container_chunk_list[0].host_info.headers[2].name	STRING			x-ms-version
[14]	storage_container_chunk_list[0].host_info.headers[2].value	STRING			2012-02-12
[15]	storage_container_chunk_list[0].host_info.headers[3].name	STRING			Content-Length
[16]	storage_container_chunk_list[0].host_info.headers[3].value	STRING			1902542
[17]	storage_container_chunk_list[0].host_info.headers[4].name	STRING			x-ms-date
[18]	storage_container_chunk_list[0].host_info.headers[4].value	STRING			Thu, 08 Aug 2013 22:27:30 GMT
[19]	storage_container_chunk_list[0].host_info.headers[5].name	STRING			Content-Type
[20]	storage_container_chunk_list[0].host_info.headers[5].value	STRING			application/octet-stream
[21]	storage_container_chunk_list[0].host_info.datacenter_name	STRING			ms_us_dmx
[22]	storage_container_chunk_list[0].chunk_checksum[0].	BYTE_STRING			[binary data: 21 bytes]
[23]	storage_container_chunk_list[0].chunk_checksum[1].	BYTE_STRING			[binary data: 21 bytes]
[24]	storage_container_chunk_list[0].storage_container_authorization_token	STRING	vendor_authToken		Az[redacted]Ed
[25]	verbosity_level	INT			2

PRX: URL Details / Var Handler - Mozilla Firefox

Apica ProxySniffer URL Details / Var Handler

Help | Project Navigator | Search Overall | Generate Load Test | Save Session | Refresh | Close

Item 8 on Page 2 : Content AuthPut
prev next
POST https://p20-content /1544917175/authorizePut
200 (OK) " +PROTOBUF" (587 bytes)

HTTP Request Header ...tent.icloud.com:443

1	POST /1544917175/authorizePut HTTP/1.1
2	Host: p20-conten 443
3	User-Agent:]
4	: 3.9
5	Dataclass.Share
6	Accept: -protob

HTTP Request Content PROT Download

G-PROTOBUF Message Type: FileChunkLists

```
file_chunk_list {
  file_checksum: "\001\370\350\254\351\313Y\26
  authorization: "A6 Ad"
  chunk_info {
    chunk_checksum: "\201p\274\207l\315gfX\314
    chunk encryption key: "\0018+\275\244G9□\2
```

Assign Var

HTTP Response Header

1	HTTP/1.1 200 OK
2	Date: Mon, 12 Aug 2013 22:53:02 GMT
3	X- 1-3922-4EF5-B6EE-A
4	x- 3.9
5	X-Responding-Instance: content.22000502.nk11p20ic-c
6	Content-Type: application +pro

HTTP Response Content 587 Bytes +PROTOBUF PROT Download

G-PROTOBUF Message Type: StorageContainerChunkLists

```
storage_container_chunk_list {
  storage_container_key: "p0w 7N2"
  host_info {
    hostname: "m :"
    port: 443
```

Extract Var

Var Handler

Extract Var from G-PROTOBUF Data

Protobuf File	MMCS.desc
Message Type	StorageContainerChunkLists
Key / Name	storage_container_chunk_list[0].storage_conta
Data Type	STRING

Recorded Value: 'Az0 Ed'

Extract whole Value
Extract Token from Value:
Token Delimiters <>"='&?
Extract Token No. 1 ?
Trim Whitespaces yes

Test Extract

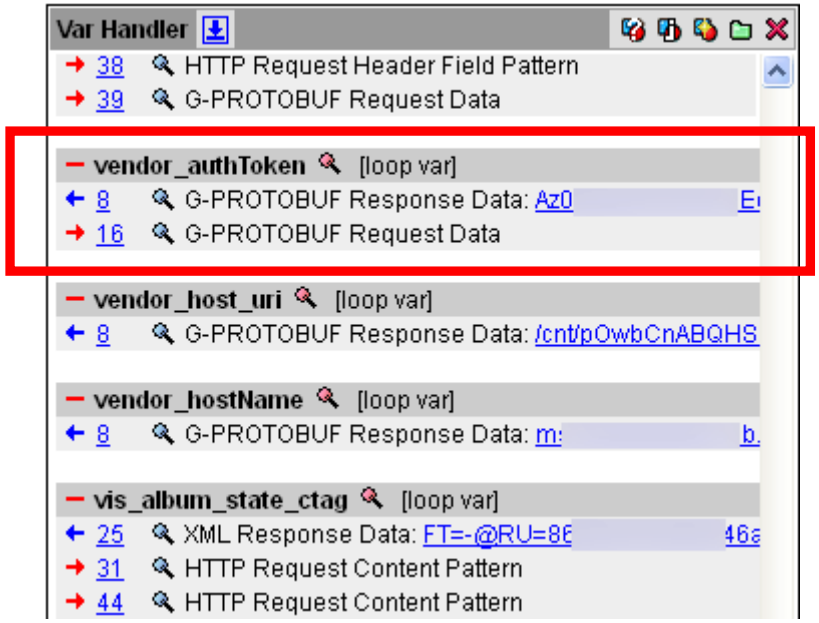
Extracted Value: 'Az0 Ed'

Map to Var Name:
vendor_authToken

Assign Var automatically to all HTTP requests which contain the same text pattern (full replacement of Extracted Value overall requests)

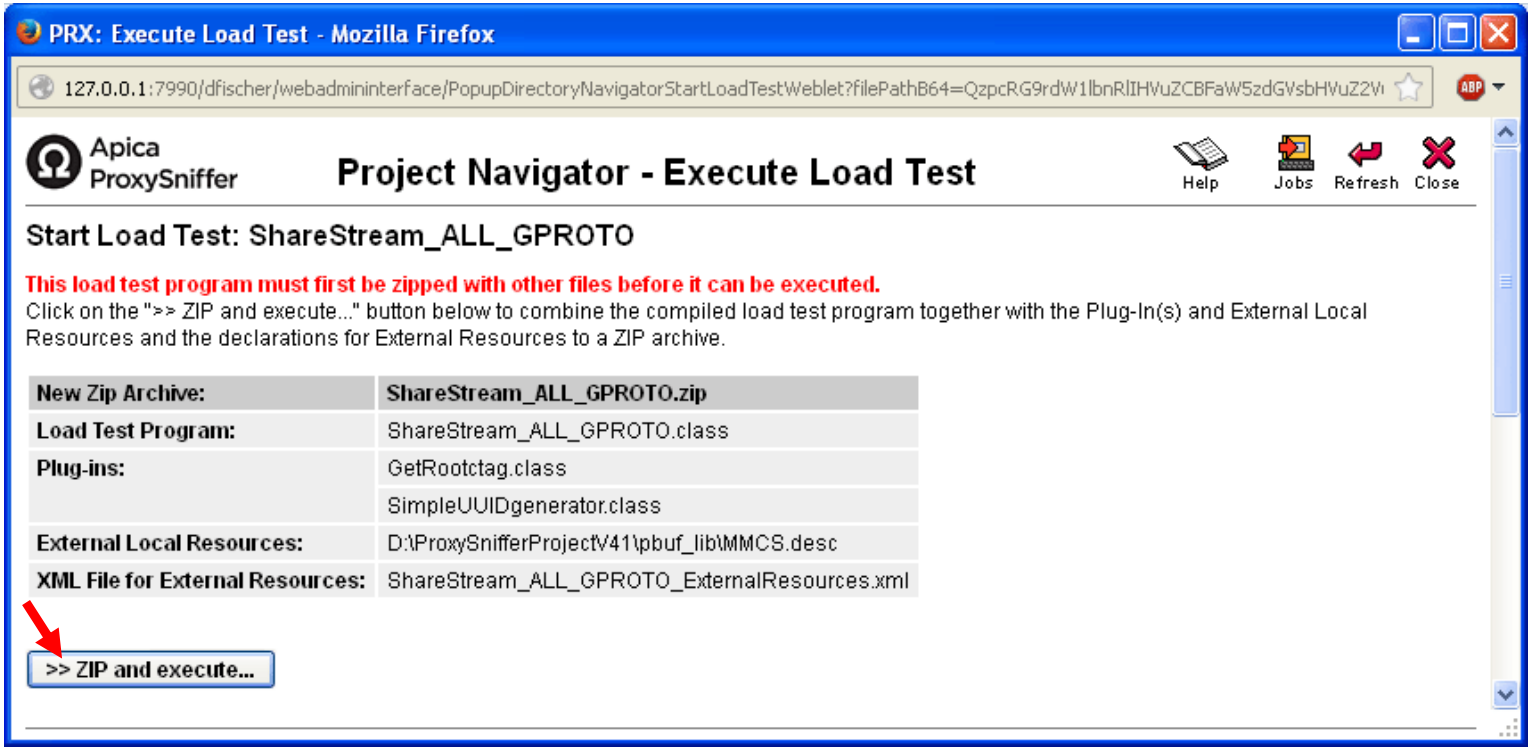
Extract

albumName [global var]
[system time milliseconds] e.g. 1380149513328
1 XML Request Data


Var Handler
38 HTTP Request Header Field Pattern
39 G-PROTOBUF Request Data
vendor_authToken [loop var]
8 G-PROTOBUF Response Data: Az0
16 G-PROTOBUF Request Data
vendor_host_uri [loop var]
8 G-PROTOBUF Response Data: /cnt/pOwbCnABQHS
vendor_hostName [loop var]
8 G-PROTOBUF Response Data:
vis_album_state_ctag [loop var]
25 XML Response Data: FT=-@RU=86
31 HTTP Request Content Pattern
44 HTTP Request Content Pattern

3 Generating the Load Test Program and Executing the Load Test

Generating the load test program and executing the load test can be done as usual – without any special additional configuration. All required resources are automatically combined to a ZIP archive and are automatically distributed to the load generators.

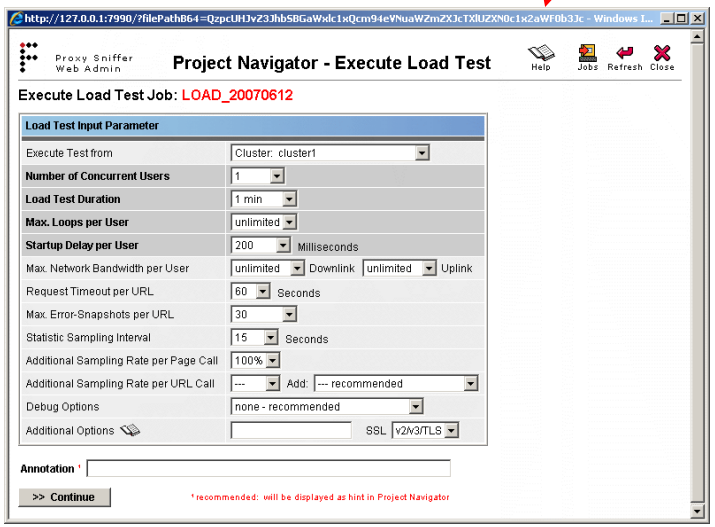


4 Manufacturer, Sales and Support

Ingenieurbüro David Fischer AG, Switzerland | A company of the Apica Group

Manufacturer's Web Site: www.proxy-sniffer.com

Apica Support: support@apicasystem.com

Apica Sales: sales@apicasystem.com

Note: All menus provide *context specific help text*, available using the Help Icon: