

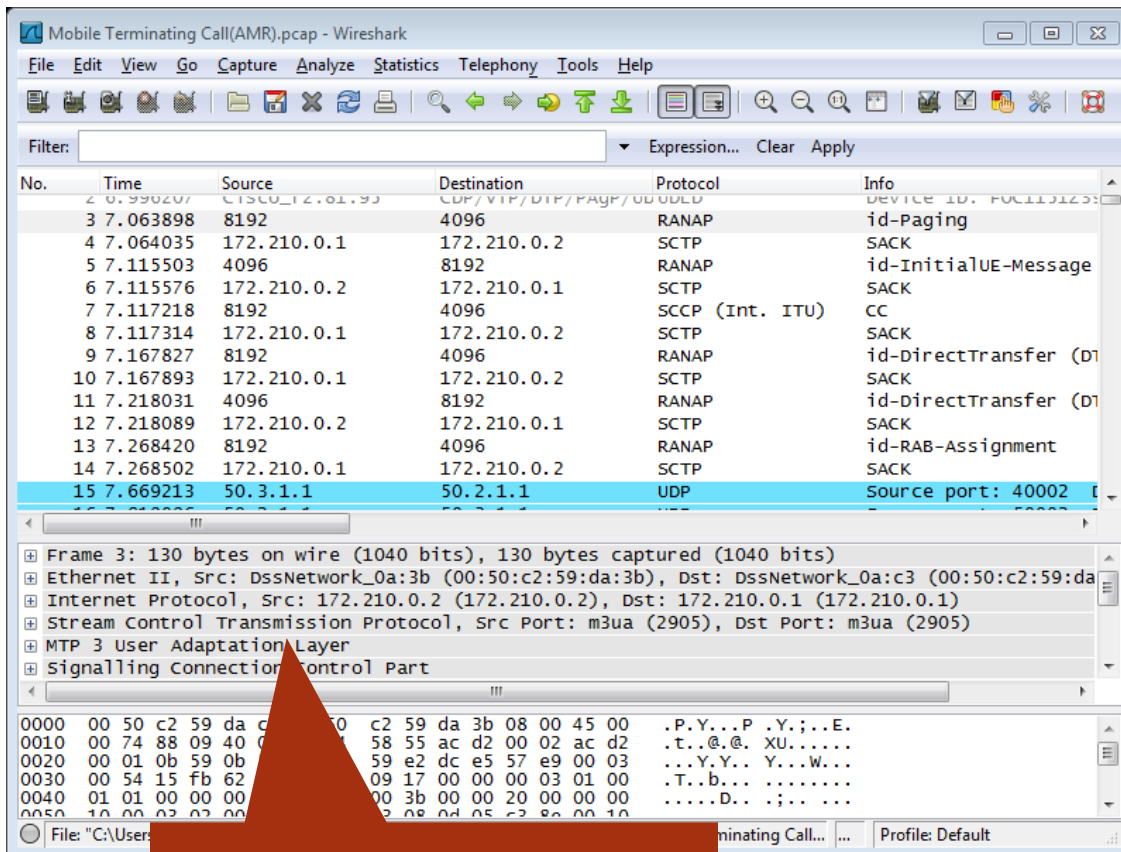
# VisualEther Protocol Analyzer 7.2

<b>WIRESHARK TO SEQUENCE DIAGRAMS.....</b>	<b>1</b>
Convert Wireshark pcap to sequence diagrams .....	1
Select messages and parameters to include in sequence diagrams .....	4
Bookmark messages for quick access .....	7
Use regular expressions for content based filter selection and styling .....	8
Use regular expression substitution to customize the displayed text .....	9
Specify a host file to map IP addresses to meaningful names .....	10
Choose between port level and IP address level sequence diagrams .....	11
Filter out periodic and traffic messages .....	14
Extract tunneled messages.....	16
Specify the color and style for messages.....	18
<b>EXPLORE THE EXAMPLES .....</b>	<b>21</b>
run-all.bat – Script diagram generation .....	23
<b>FXT REFERENCE .....</b>	<b>24</b>
Supported protocols.....	24
Define your own protocols.....	26
Specify the message type and parameters .....	27
Include remarks .....	28
Attributes .....	29
Regular expressions .....	31
Working around incomplete Wireshark field definitions.....	32
<b>COLORS .....</b>	<b>33</b>

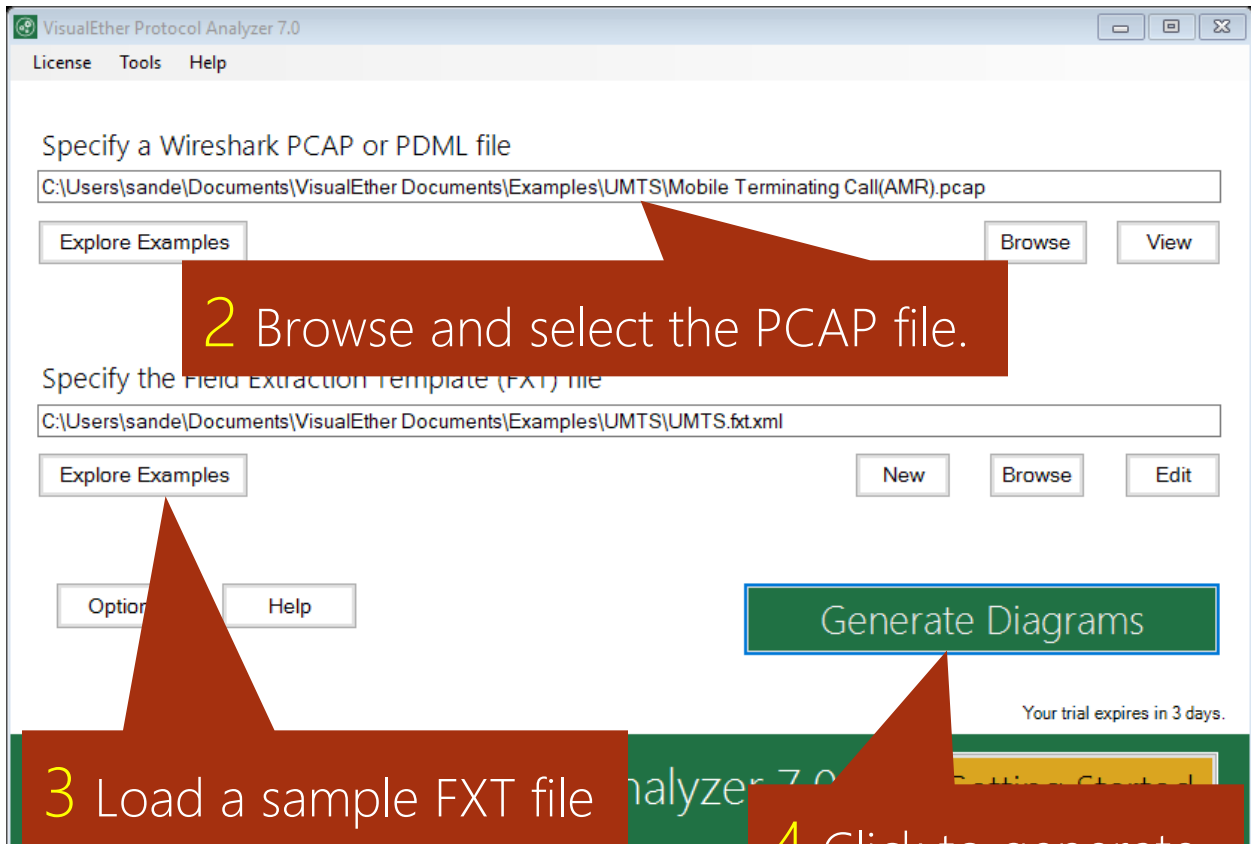
# Wireshark to sequence diagrams

Convert Wireshark pcap to sequence diagrams

 [Wireshark PCAP to sequence diagrams](#)



1 Save Wireshark capture in a PCAP file.



3 Load a sample FXT file that defines the templates for the messages to be included in the sequence diagram.

4 Click to generate sequence diagrams.

5 VisualEther generates a sequence diagram.

6 Click on any message in the PDF file to see field level details.

The screenshot shows a PDF viewer displaying a sequence diagram. The diagram includes messages such as 'RANAP Paging (14)' and 'RANAP InitialUE-Message (19)'. A callout box points to a specific message in the diagram, and another callout box points to the corresponding message in the PDF's message list. The message details for 'RANAP InitialUE-Message (19)' are visible, showing fields like 'id-CN-DomainIndicator (3)', 'id-LAI (15)', and 'pLMNIdentity: 62f110'.

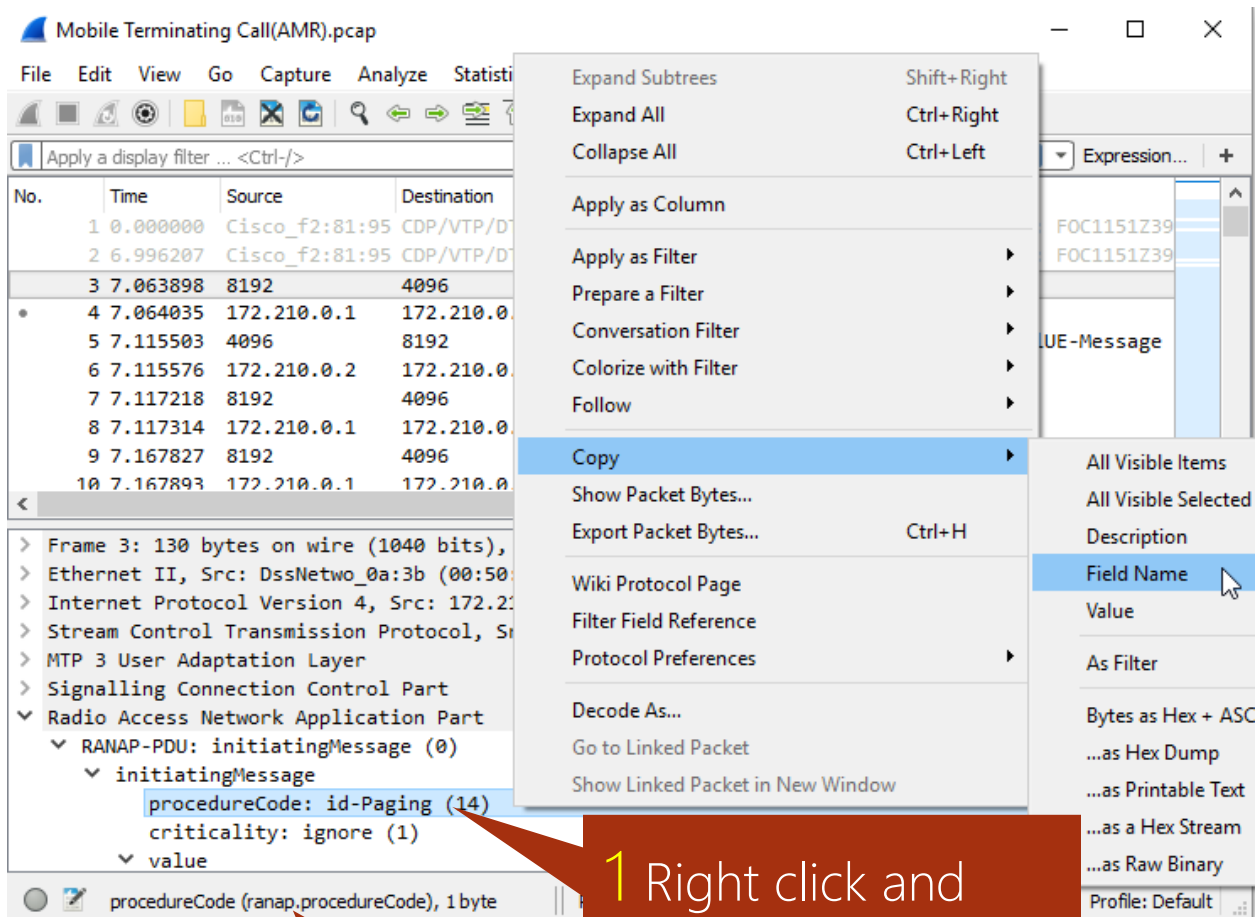
**Professional Edition feature:** Clicking on messages is not supported in the Community Edition.

7 See a detailed dump of the message.

The screenshot shows a detailed protocol dump for 'Frame Number: 5'. The dump is organized into a tree structure with expandable sections. The 'gsm\_a.dtap' section is expanded, showing fields like 'gsm\_a.dtap.protocol\_discriminator', 'gsm\_a.dtap.msg\_rr\_type', and 'gsm\_a.rr.ciphering\_key\_seq\_num'. A callout box points to the 'gsm\_a.rr.ciphering\_key\_seq\_num' field.

Select messages and parameters to include in sequence diagrams

**YouTube** [Add parameters to messages](#)

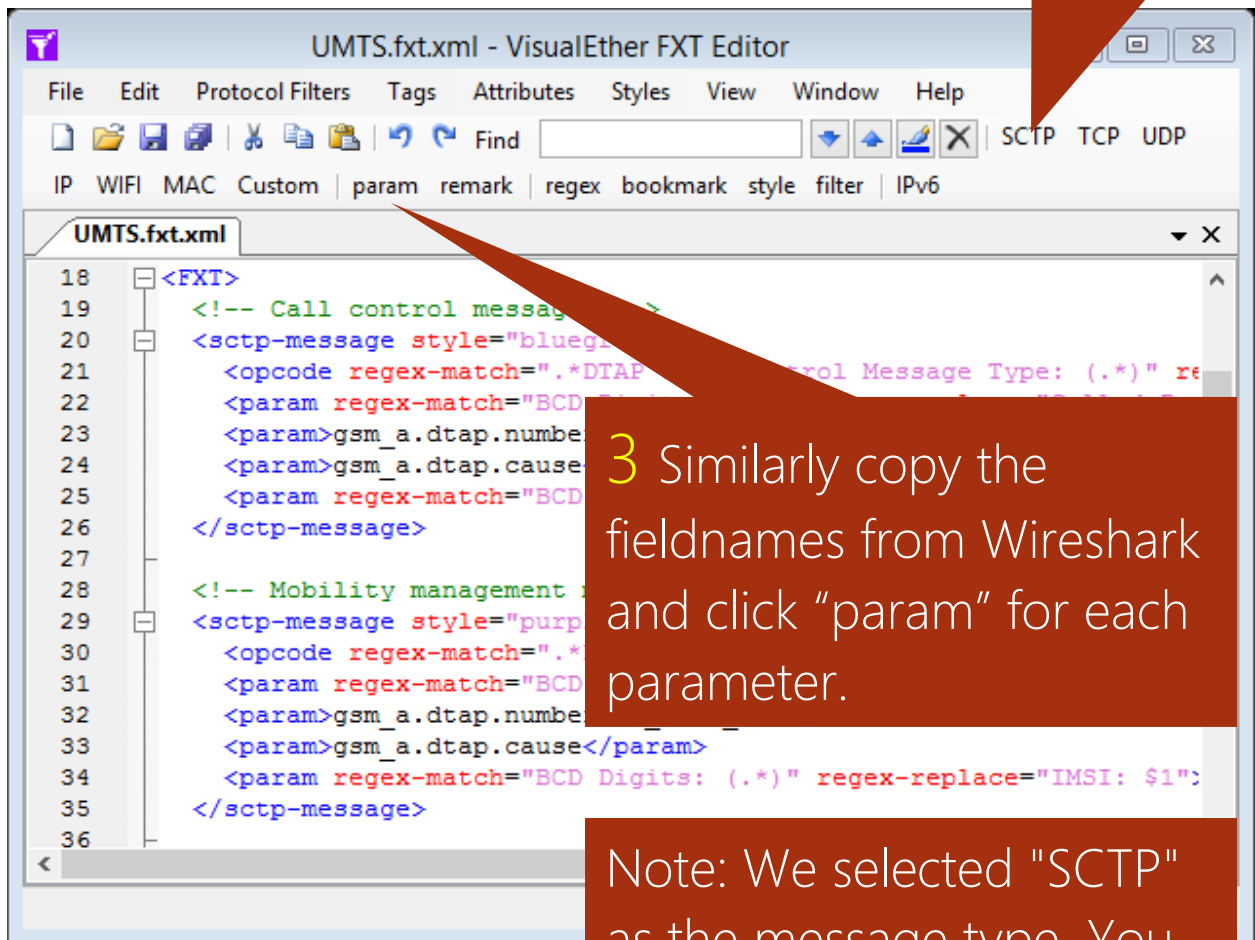


1 Right click and copy the field name.

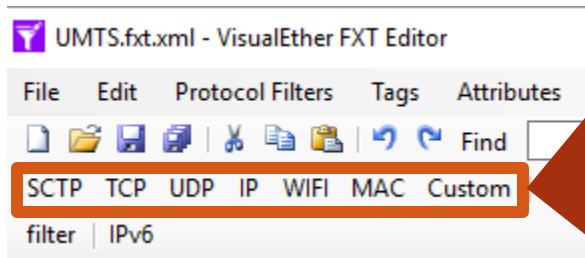
Note that Wireshark also displays the fieldname in the status

Define a FXT file with templates for messages you wish to include in the message.

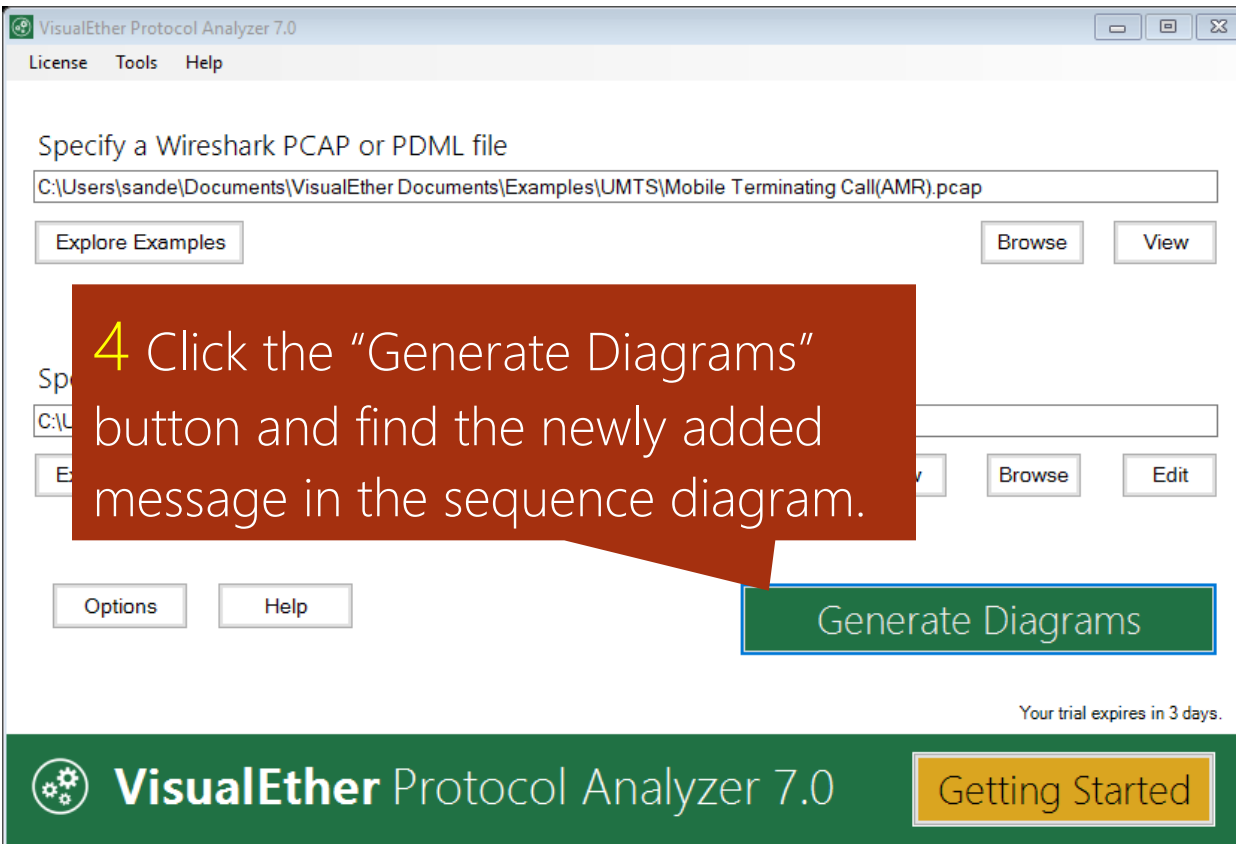
2 Click "SCTP". The selected field code is used as the opcode.



3 Similarly copy the fieldnames from Wireshark and click "param" for each parameter.



Note: We selected "SCTP" as the message type. You can choose between Sctp, TCP, UDP, IP, WIFI, MAC, or roll out your custom message type.



# Bookmark messages for quick access

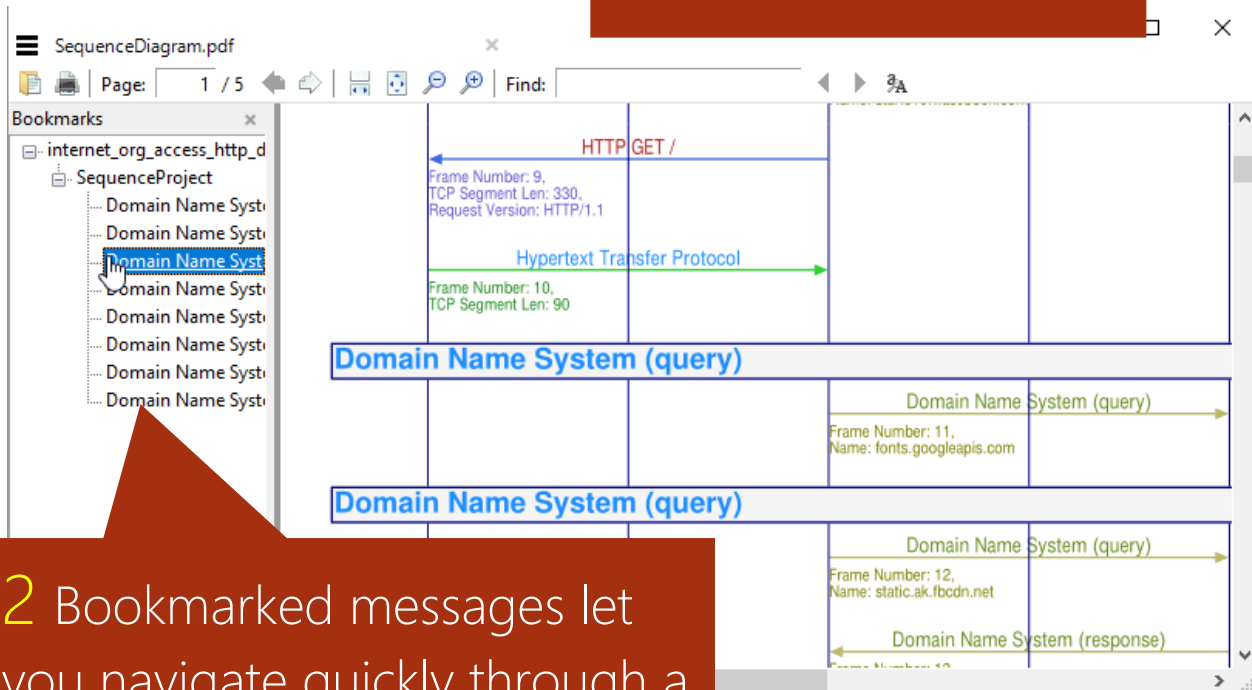
## Bookmark error conditions in PDF

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>

  <!-- Message Template for Domain Name System (DNS) Extraction -->
  <!-- Capture DNS messages apply the greenkhaki style. Use the
  "bookmark" attribute for quick navigation in the PDF file. -->
  <udp-message style="greenkhaki" bookmark="true" >
    <opcode regex-match="^Domain">dns</opcode>
    <param>dns.qry.name</param>
    <param>dns.resp.name</param>
    <param>dns.Addr</param>
  </udp-message>

  <!--More templates ... -->
</FXT>
```

1 Select the messages to be bookmarked.



2 Bookmarked messages let you navigate quickly through a large sequence diagram.



# Use regular expressions for content based filter selection and styling

**YouTube** Add style and color

**YouTube** Flag error messages with regular expressions

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>
  <!-- Template for Domain Name System (DNS) Extraction -->
  <!-- Capture DNS messages that end with "(query)".
  Apply the greenkhaki style.
  Also bookmark in PDF.
  -->
  <udp-message style="greenkhaki" bookmark="true" >
    <opcode regex-match="\((query)\)$">dns</opcode>
    <param>dns.qry.name</param>
    <param>dns.resp.name</param>
    <param>dns.Addr</param>
  </udp-message>

  <!-- Other DNS messages are included but they are not bookmarked -->
  <udp-message style="greenkhaki">
    <opcode regex-match="^Domain">dns</opcode>
    <param>dns.qry.name</param>
    <param>dns.resp.name</param>
    <param>dns.Addr</param>
  </udp-message>

  <!-- Template for Hypertext Transfer Protocol (HTTP) request Extraction -->
  <tcp-message style="redblue">
    <opcode>http.request.method</opcode>
    <param>http.request.uri</param>
    <param>http.request.version</param>
    <param>http.response.code</param>
    <param>http.If-Modified-Since</param>
    <param>tcp.len</param>
  </tcp-message>

  <!--more... -->
</FXT>
```

Filters can be applied on the content of the captured field.

# Use regular expression substitution to customize the displayed text

## Customize message titles with regex

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>

  <!-- Call control messages -->
  <sctp-message style="bluegreen">
    <opcode regex-match=".*DTAP Call Control Message Type: (.*)"
           regex-replace="DTAP CC $1">gsm_a.dtap_msg_cc_type</opcode>
    <param>gsm_a.cld_party_bcd_num</param>
    <param>gsm_a.numbering_plan_id</param>
    <param>gsm_a_dtap.cause</param>
    <param>gsm_a.imsi</param>
  </sctp-message>

  <!-- Mobility management messages -->
  <sctp-message style="purpleblue" bookmark="true">
    <opcode regex-match=".*DTAP Mobility Management Message Type: (.*)"
           regex-replace="DTAP MM $1">gsm_a.dtap_msg_mm_type</opcode>
    <param>gsm_a.cld_party_bcd_num</param>
    <param>gsm_a.numbering_plan_id</param>
    <param>gsm_a_dtap.cause</param>
    <param>gsm_a.imsi</param>
  </sctp-message>

  <!-- RANAP signaling -->
  <sctp-message style="redblue">
    <opcode regex-match="procedureCode: id-(.*)"
           regex-replace="RANAP $1">ranap.procedureCode</opcode>
    <param>ranap.pLMNidentity</param>
    <param>ranap.id</param>
    <param>gsm_a.imsi</param>
  </sctp-message>

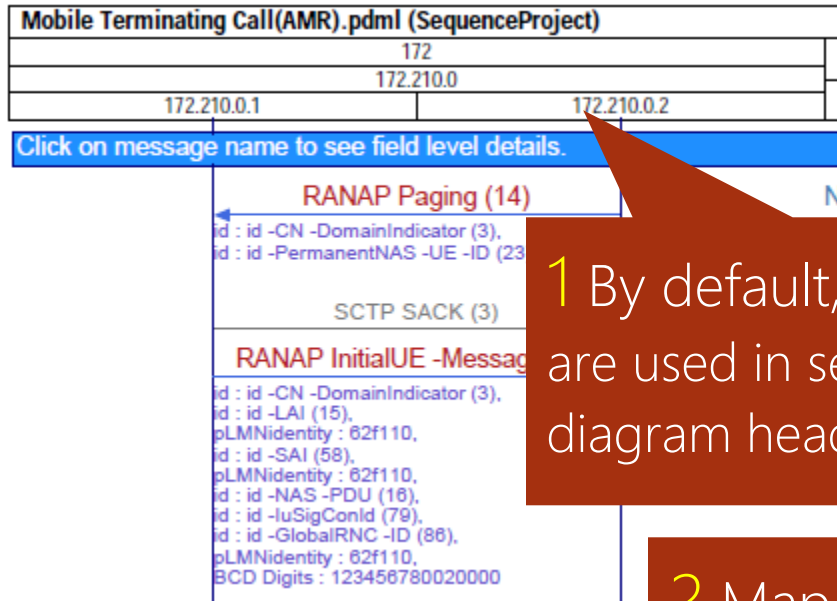
  <!-- ... -->
</FXT>
```

You can further customize the content of opcodes and parameters with regular expression substitution.

Group the patterns you are interested in and reference them with \$1, \$2 ...

# Specify a host file to map IP addresses to meaningful names

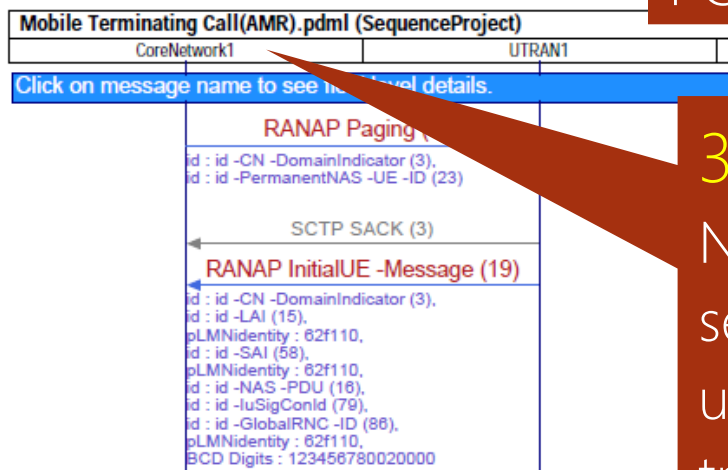
**Professional Edition feature:** HOST files are not supported in the Community Edition.



1 By default, IP addresses are used in sequence diagram headings.

2 Map the addresses to names in HOSTS.txt file and place it in the same directory as the PCAP file.

```
# Hosts file for the UMTS example
172.210.0.1 UTRAN1
172.210.0.2 CoreNetwork1
```



3 Generate diagrams. Notice that the sequence diagrams now use the hosts file translation.

# Choose between port level and IP address level sequence diagrams

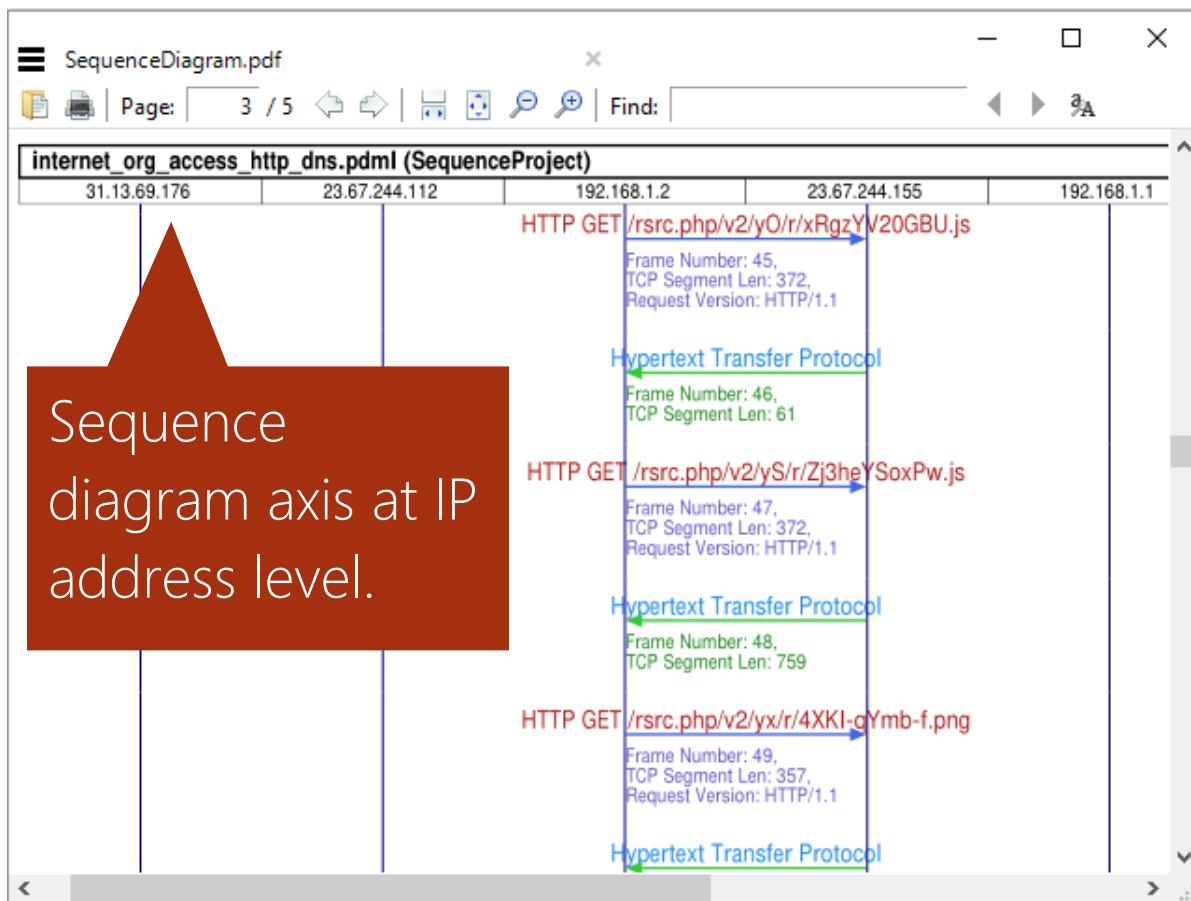
**Professional Edition feature:** Community Edition is limited to IP axis based diagrams.

VisualEther lets to draw sequence diagrams at IP address level or port level. The difference between the three options is best explained with the Options dialog selection and the generated sequence diagram.

## Draw instance axis at IP address level

Draw sequence diagram instance axis at:

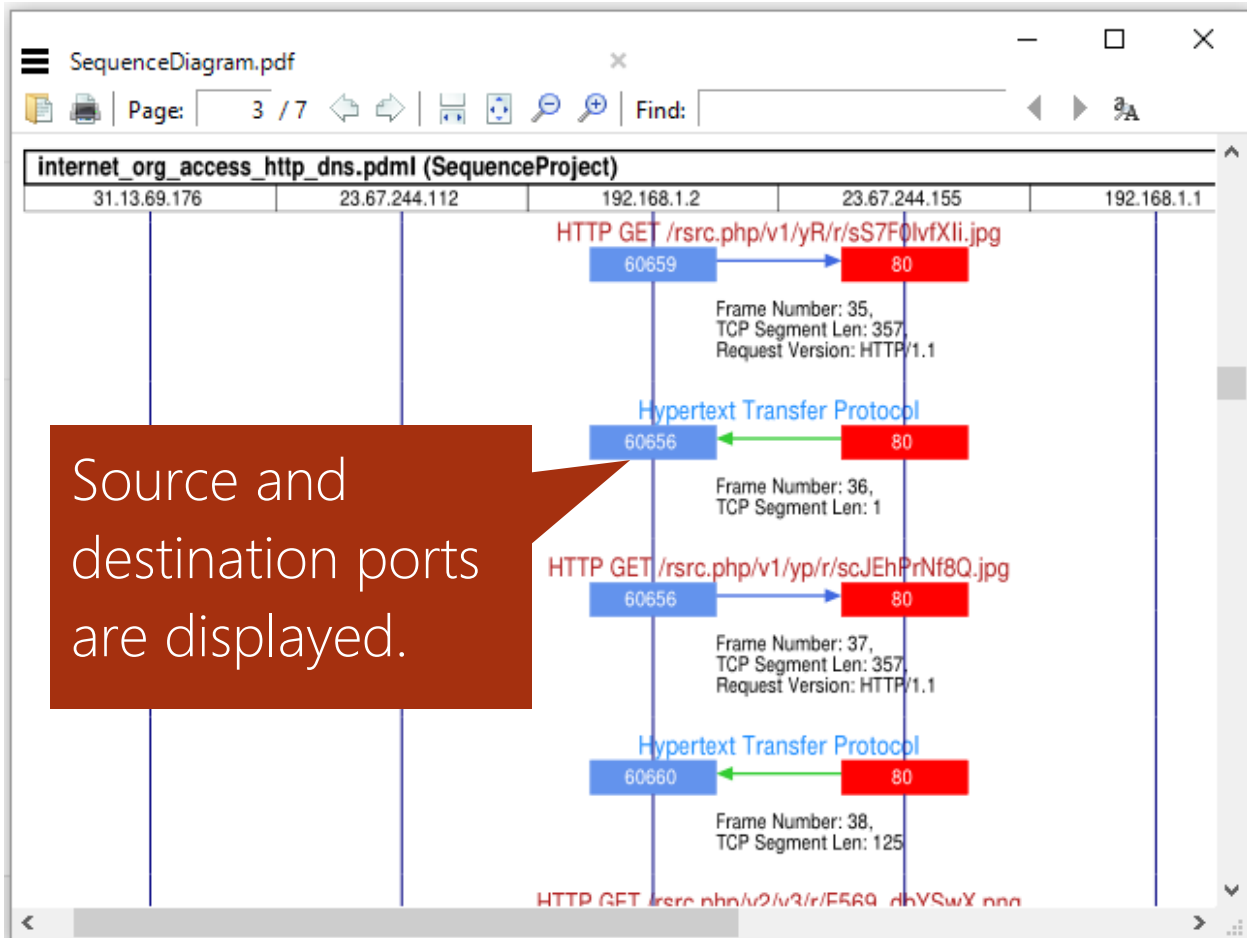
- TCP, UDP and SCTP port level
- IP Address level and display port numbers
- IP Address level



Draw instance axis at IP address level and display port numbers

Draw sequence diagram instance axis at:

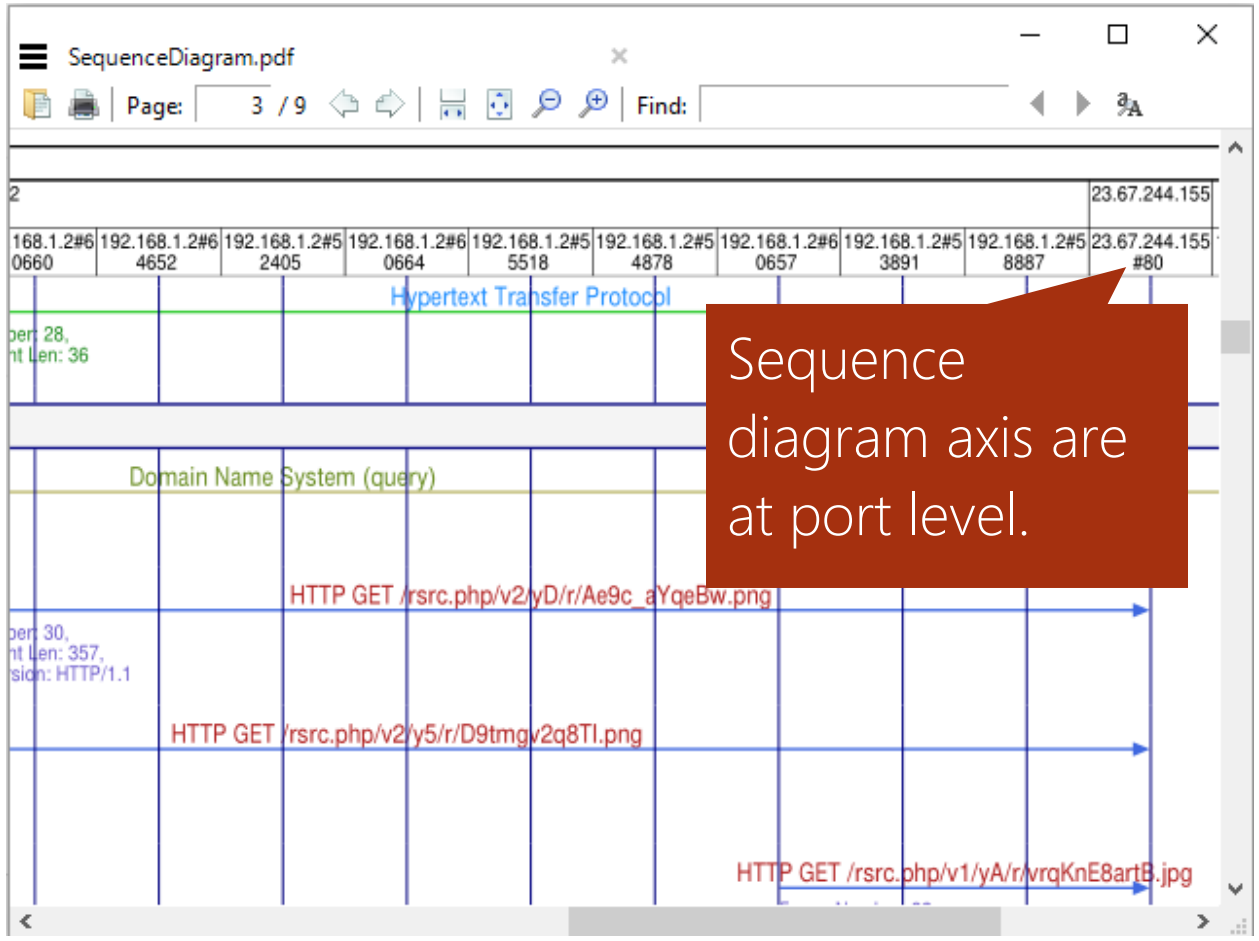
- TCP, UDP and SCTP port level
- IP Address level and display port numbers
- IP Address level



## Draw instance axis at TCP, UDP and SCTP port level

Draw sequence diagram instance axis at

- TCP, UDP and SCTP port level
- IP Address level and display port numbers
- IP Address level



# Filter out periodic and traffic messages

When capturing SIP and IMS calls, RTP and RTCP packets can crowd out the signaling handshakes. Periodic messages like the Wi-Fi beacon can also clutter the generated sequence diagram.

A filter attribute can be added to filter out periodic and traffic flow messages. When the filter attribute is set, only one message of the matching message type is displayed.

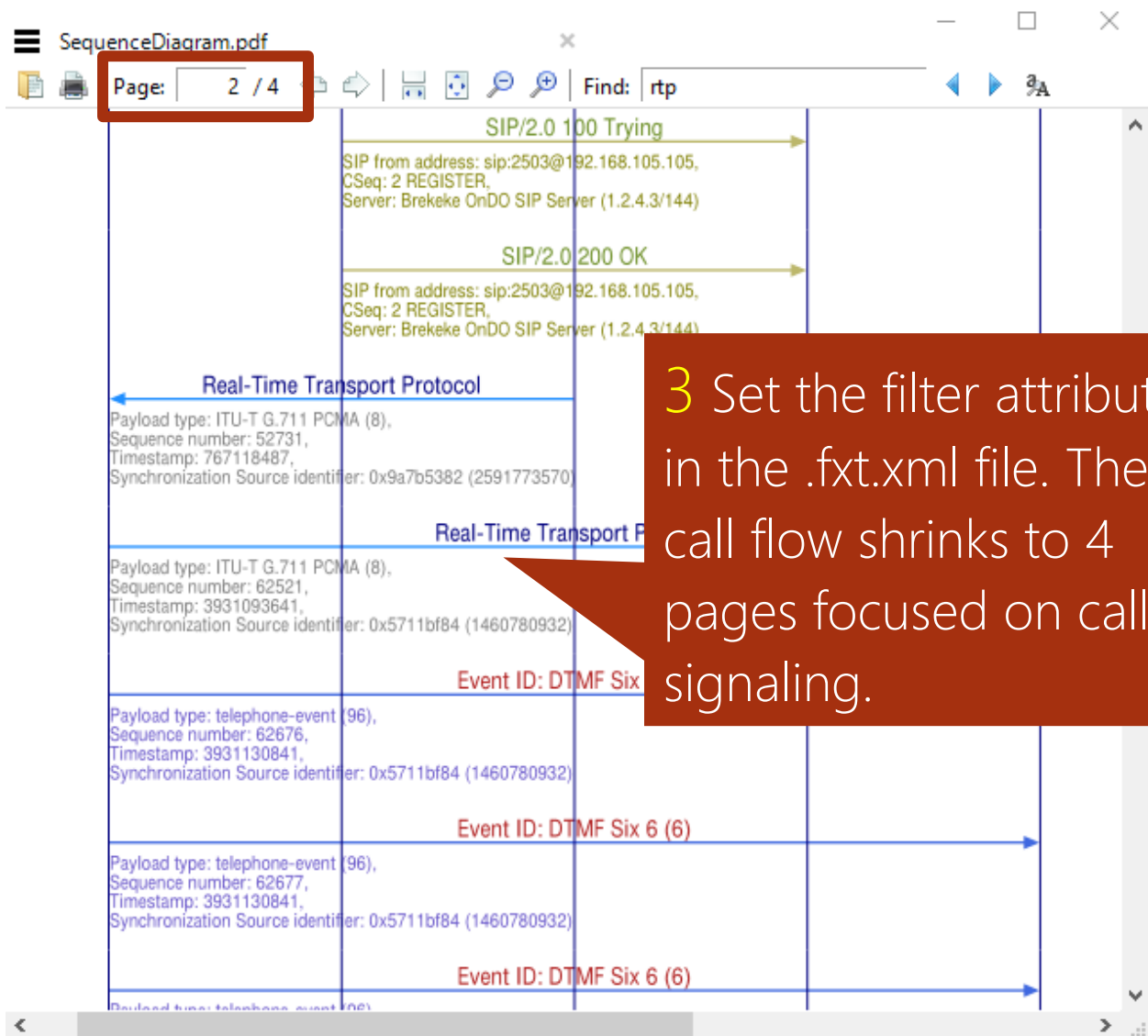
The screenshot shows a sequence diagram window titled "SequenceDiagram.pdf". At the top, there is a search bar with the text "Find: rtp" and a page indicator "Page: 2 / 76". The diagram displays several messages between two lifelines:

- SIP/2.0 100 Trying (green arrow pointing right)
- SIP/2.0 200 OK (green arrow pointing right)
- Real-Time Transport Protocol (blue arrows pointing left and right)

Each RTP message includes details such as "Payload type: ITU-T G.711 PCMA (8)", "Sequence number", "Timestamp", and "Synchronization Source identifier". A red callout box with a white border contains the text: "1 The SIP-RTP sample results in a 76-page call flow if all RTP packets are shown!".

```
<udp-message filter="true">  
  <opcode display="brief">rtp</opcode>  
  <param>rtp.p_type</param>  
  <param>rtp.ssrc</param>  
  <param>rtp.seq</param>  
  <param>rtp.timestamp</param>  
</udp-message>
```

2 Add a filter for RTP messages. This removes out all but one RTP message in each direction.



3 Set the filter attribute in the .fxt.xml file. The call flow shrinks to 4 pages focused on call signaling.



# Extract tunneled messages

When dealing with tunneling protocols like GTP you can choose between the outer and the inner message by specifying the skip attribute.

```

Frame 9: 174 bytes on wire (1392 bytes captured) on interface 0
Ethernet II, Src: 00:00:00_00:00:00, Dst: 00:00:00_00:00:00
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (Default)
  Total Length: 160
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (17)
  Header checksum: 0x3c4a [validation disabled]
  Source: 127.0.0.1 (127.0.0.1)
  Destination: 127.0.0.2 (127.0.0.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 3386 (3386), Dst Port: 3386
GPRS Tunneling Protocol
  Flags: 0x1e
  Message Type: T-PDU (0xff)
  Length: 112
  Sequence number: 0x0000
  Flow label: 0x0001
  SNDCP N-PDU LLC Number: 0xff
  TID: 2400101234567890
  T-PDU Data 112 bytes
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.3
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (Default)
  Total Length: 112
  Identification: 0xa9a2 (4342)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 64
  Protocol: ICMP (1)
  Header checksum: 0x7ed6 [validation disabled]
  Source: 192.168.0.1 (192.168.0.1)
  Destination: 192.168.0.3 (192.168.0.3)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]

```

By default, VisualEther will select the fields from the outer message

Fields from the inner message can be selected by using the skip="1" attribute

## Capturing the outer message

VisualEther defaults to capturing the outer message.

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>
  <message>
    <opcode>gtp.message</opcode>
    <param>gtp.length</param>
    <param>gtp.teid</param>
    <param>gtp.seq_number</param>
    <param>gtp.apn</param>
    <param>pap.code</param>
    <param>gtp.gsn_ipv4</param>
    <param>gsm_map.address.digits</param>

    <source>
      <address>ip.src</address>
    </source>
    <destination>
      <address>ip.dst</address>
    </destination>
  </message>
</FXT>
```

## Capturing the inner message

Adding a skip-attribute results in VisualEther ignoring the outer message fields and capturing the fields from the inner message.

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>
  <message>
    <opcode>icmp.type</opcode>
    <param skip="1">ip.len</param>
    <source>
      <address skip="1">ip.src</address>
    </source>
    <destination>
      <address skip="1">ip.dst</address>
    </destination>
  </message>
</FXT>
```

# Specify the color and style for messages

```

<?xml version="1.0" encoding="utf-8"?>
<FXT>
  <!-- Message Template for Domain Name System (DNS) Extraction -->
  <udp-message style="greenkhaki">
    <opcode regex-match="^Domain">dns</opcode>
    <param>dns.qry.name</param>
    <param>dns.resp.name</param>
    <param>dns.Addr</param>
  </udp-message>

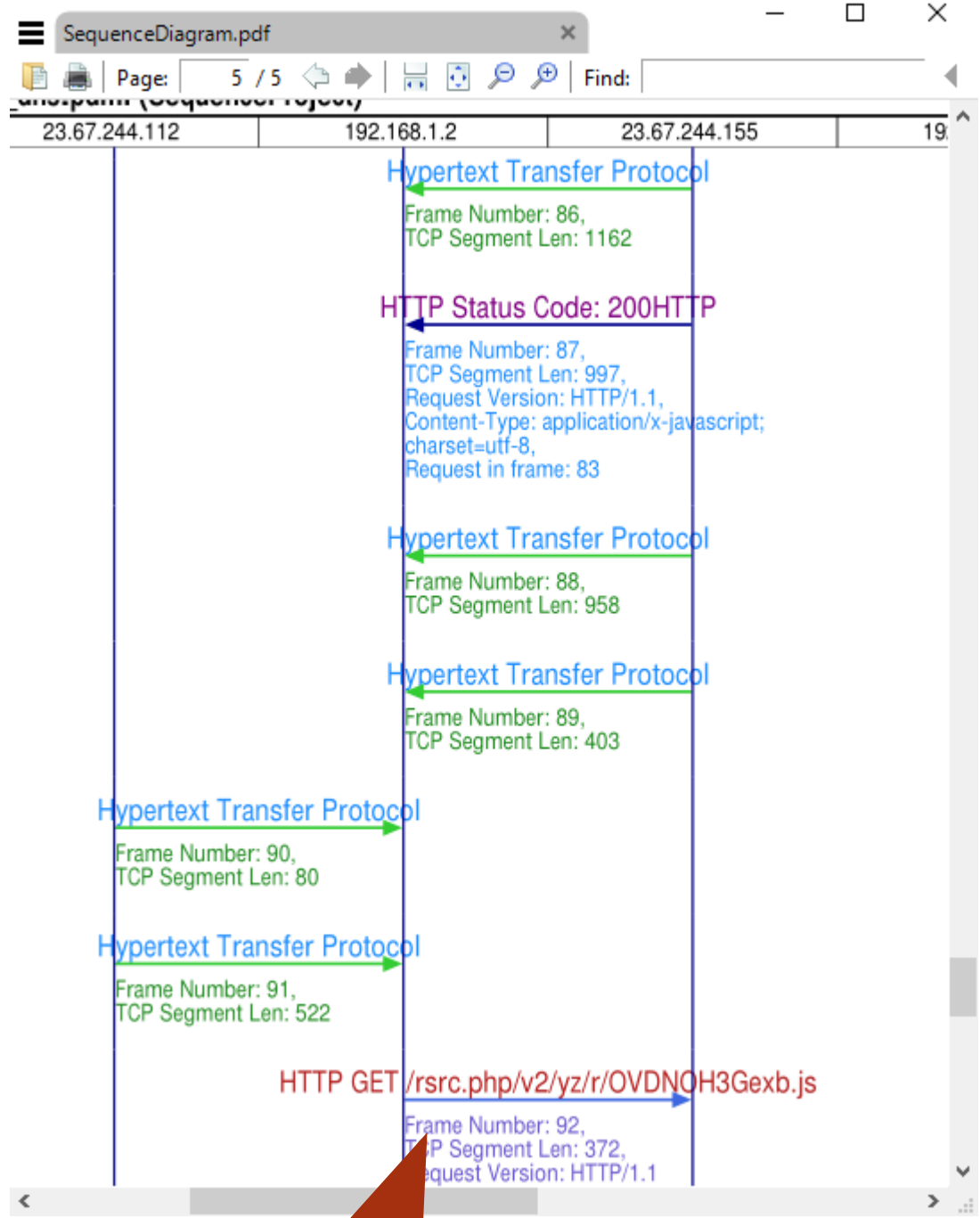
  <!-- Template for Hypertext Transfer Protocol (HTTP) Request Extraction -->
  <tcp-message style="redblue">
    <opcode>http.request.method</opcode>
    <param>http.request.uri</param>
    <param>http.request.version</param>
    <param>http.response.code</param>
    <param>http.If-Modified-Since</param>
    <param>tcp.len</param>
  </tcp-message>

  <!-- Template for Hypertext Transfer Protocol (HTTP) Response -->
  <tcp-message style="purpleblue">
    <opcode>http.response.code</opcode>
    <param>http.request.uri</param>
    <param>http.request.version</param>
    <param>tcp.len</param>
  </tcp-message>

  <!-- Default Message Template for Hypertext Transfer Protocol (HTTP) -->
  <tcp-message style="bluegreen">
    <opcode>http</opcode>
    <param>http.request.uri</param>
    <param>http.request.version</param>
    <param>http.response.code</param>
    <param>tcp.len</param>
  </tcp-message>
</FXT>

```

1 Add a style to a filter to choose the font, size and color.



2 Regenerate documents with selected styles.

VisualEther Options ×

Draw sequence diagram instance axis at:

- TCP, UDP and SCTP port level
- IP Address level and display port numbers
- IP Address level

Max parameters per message:

Packets per PCAP file:

Clicking on a message title in the PDF sequence diagram shows complete message contents at field level.

EventStudio path

Style and Theme File

Default HOSTS file

Limit diagram to nodes in the HOSTS file

3 Customize the themes by editing the VisualEther.fdl file.

### Sample styles

```
style redblue: textcolor=FIREBRICK, color=ROYALBLUE, paramcolor=SLATEBLUE
style bluegreen: textcolor=DODGERBLUE, color=LIMEGREEN, paramcolor=FORESTGREEN
style bluegrey: textcolor=MEDIUMBLUE, color=GREY, paramcolor=DIMGREY
style greenkhaki: textcolor=OLIVEDRAB, color=DARKKHAKI, paramcolor=OLIVE
style purpleblue: textcolor=PURPLE, color=DARKBLUE, paramcolor=DODGERBLUE
```

# Explore the examples



[Explore the examples](#)

Get started with examples that cover a range of protocols from ARP to X.509. The examples include PCAP files, extraction template files (.FXT.XML).

Some examples include Hosts.txt file that allows you to substitute IP address axis headings with host names.

The examples are installed in:

My Documents\VisualEther Documents\Examples

1 Click "Explore Examples" to browse the available examples.

VisualEther Protocol Analyzer 7.0

License Tools Help

Specify a Wireshark PCAP or PDML file

C:\Users\sande\Documents\VisualEther Documents\Examples\UMTS\Mobile Terminating Call(AMR).pcap

Explore Examples Browse View

Specify the Field Extraction Template (FXT) file

C:\Users\sande\Documents\VisualEther Documents\Examples\UMTS\UMTS.fxt.xml

Explore Examples New Browse Edit

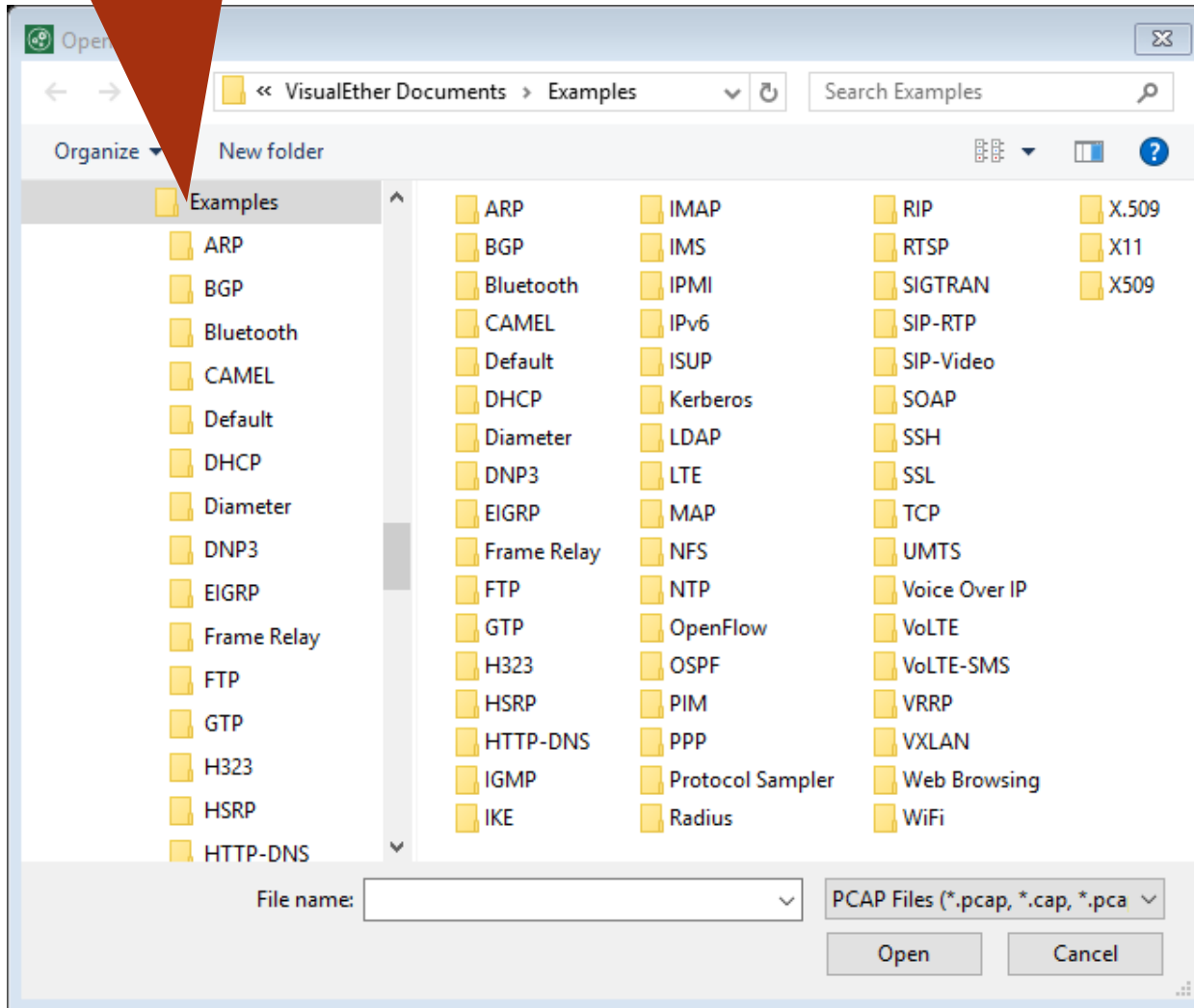
Options Help

Generate Diagrams

Your trial expires in 3 days.

VisualEther Protocol Analyzer 7.0 Getting Started

2 Choose from more than 45 examples.



## run-all.bat – Script diagram generation

```

11 start /wait vether BGP/bgp.pcap BGP/BGP.fxt.xml
12 start /wait vether BGP/BGP_MP_NLRI.cap BGP/BGP.fxt.xml
13 start /wait vether BGP/BGP_soft_reset.cap BGP/BGP.fxt.xml
14 start /wait vether DHCP/"DHCP_MessageType 10,11,12 and
15 start /wait vether Diameter/diameter-and-other-protocol
16 start /wait vether DNP3/DNP3ReadRequest.pcap DNP3/DNP3.
17 start /wait vether DNP3/DNP3SelectOperateRequest.pcap I
18 start /wait vether EIGRP/EIGRPv2_subnet_transition.cap
19 start /wait vether "Frame Relay"/ICMP_across_frame_rela
20 start /wait vether FTP/FTP-Telnet.pcap FTP/FTP-Telnet.f
21 start /wait vether Bluetooth/Bluetooth1.pcapng Blueoot
22 start /wait vether Bluetooth/bluetooth-l2ping.pcap Blue
23 start /wait vether CAMEL/camel2.pcap CAMEL/camel.fxt.xml
24 start /wait vether EIGRP/EIGRPv2_subnet_transition.cap
25 start /wait vether GPRS/Alcatel_Ericsson.pcap GPRS/GPRS

```

Add the VisualEther installation directory to the default search path. Use the **start /wait** primitive in batch files to invoke VisualEther via a command-line interface.

The **run-all.bat** sample batch file in the Examples directory generates diagrams from all the samples included with VisualEther.



# FXT reference

## Supported protocols

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<FXT>
```

```
<!-- Message Templates for Session Initiation Protocol (SIP) Extraction -->
```

```
<udp-message>
```

```
<opcode display="brief">sip.Request-Line</opcode>
<param display="brief">sip.from.addr</param>
<param display="brief">sdp.connection_info</param>
```

```
</udp-message>
```

UDP v4

```
<udp-message>
```

```
<opcode display="brief">sip.Request-Line</opcode>
<param display="brief">sip.from.addr</param>
<param display="brief">sdp.connection_info</param>
```

```
</udp-message>
```

UDP v6

```
<!-- Message Template for File Transfer Protocol (FTP) Extraction -->
```

```
<tcp-message>
```

```
<opcode display="brief">ftp</opcode>
<param display="brief">ftp.response.code</param>
<param display="brief">ftp.response.arg</param>
<param display="brief">ftp.request.command</param>
```

```
</tcp-message>
```

TCP v4

TCP v6

```
<tcpv6-message>
```

```
<opcode display="brief">ftp</opcode>
<param display="brief">ftp.response.code</param>
<param display="brief">ftp.response.arg</param>
<param display="brief">ftp.request.command</param>
```

```
</tcpv6-message>
```

```
<!-- Message Template for Transaction Capabilities Application Part (TCAP)
      Extraction -->
```

```
<sctp-message style="redblue">
  <opcode>tcap</opcode>
  <param>tcap.oid</param>
  <param>tcap.application_context_name</param>
  <param>tcap.otid</param>
  <param>tcap.msgtype</param>
</sctp-message>
```

SCTP v4 for telecom signaling

```
<sctpv6-message style="redblue">
```

```
  <opcode>tcap</opcode>
  <param>tcap.oid</param>
  <param>tcap.application_context_name</param>
  <param>tcap.otid</param>
  <param>tcap.msgtype</param>
</sctpv6-message>
```

SCTP v6 for telecom signaling

```
<!-- Message Template for Internet Control Message Protocol (ICMP)
      Extraction -->
```

```
<ip-message>
  <opcode>icmp.type</opcode>
  <param>icmp.seq</param>
</ip-message>
```

IPv4

```
<ipv6-message>
  <opcode>icmp.type</opcode>
  <param>icmp.seq</param>
</ipv6-message>
```

IPv6

```
<!-- Display the beacon message, but filter out duplicates -->
```

```
<wifi-message filter="true" style="grey">
  <opcode regex-match="Type/Subtype: Beacon (.*)" regex-
replace="IEEE 802.11: Beacon $1">wlan.fc.type_subtype</opcode>
  <param>wlan.seq</param>
  <param regex-match="Tag interpretation: (.*)" regex-
replace="$1">wlan_mgt.tag.interpretation</param>
  <param>data.len</param>
</wifi-message>
```

WiFi – 802.11 Wireless LAN

```

<!-- Display ARP messages -->
<mac-message style="purpleblue">
  <opcode regex-match="Opcode: (.*)" regex-replace="ARP $1">arp.opcode</opcode>
  <param regex-match="(.*)\((.*)\)"
    regex-replace="Who has $2?">arp.dst.proto_ipv4</param>
  <param regex-match="(.*)\((.*)\)"
    regex-replace="Tell $2">arp.src.proto_ipv4</param>
</mac-message>
</FXT>

```

Ethernet frames

## Define your own protocols

You are not limited to the predefined protocols. You can add filters for any custom protocol using the **<source>** and **<destination>** tags.

The following example demonstrates how the source and destination entities can be specified using the source and destination tags. These address tag in source and destination identifies the node. The port tag specifies the field that maps to the port number. Here the originating and destination point codes are used as the source and destination nodes. The SLS field is used as the port number.

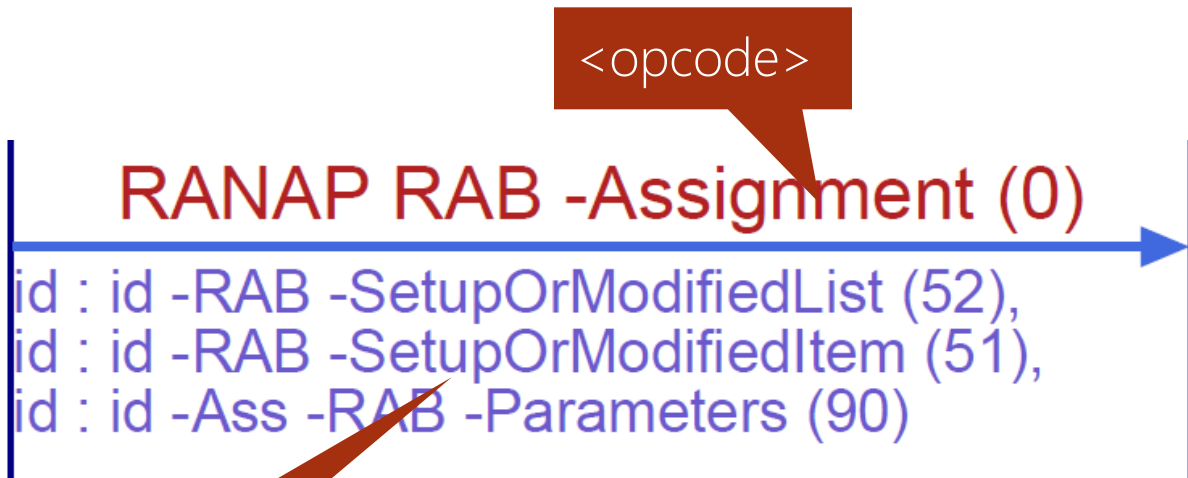
```

<?xml version="1.0" encoding="utf-8" ?>
<!-- -->
<FXT>
  <!-- MAP (Mobile Application Part) messages -->
  <message style="purpleblue">
    <opcode
      regex-match="private: \d{4} (.*)"
      regex-replace="MAP $1">ansi_tcap.private</opcode>
    <param>ansi_map.bcd_digits</param>
    <param>ansi_map.mscid</param>
    <param>ansi_map.serviceIndicator</param>
    <param>ansi_map.actionCode</param>
    <param>ansi_683.for_msg_type</param>
    <param>ansi_683.rev_msg_type</param>
    <param>ansi_tcap.ComponentPDU</param>
    <source>
      <address>mtp3.opc</address>
      <port>mtp3.sls</port>
    </source>
    <destination>
      <address>mtp3.dpc</address>
      <port>mtp3.sls</port>
    </destination>
  </message>
</FXT>

```

SS7 support added with point codes as addresses and SLS as port number.

Specify the message type and parameters



<param>

The <opcode> tag extracts the message name. A message is only included if a matching <opcode> tag is found.

```
<sctp-message style="redblue">
  <opcode regex-match="procedureCode: id-(*)"
    regex-replace="RANAP $1">ranap.procedureCode</opcode>
  <param>ranap.pLMNidentity</param>
  <param>ranap.id</param>
  <param>gsm_a.imsi</param>
</sctp-message>
```

Use the <param> tag to specify the parameters that should be included with the message.

# Include remarks



```
<sctp-message style="purpleblue" bookmark="true">
  <opcode regex-match=".*DTAP Mobility Management Message Type: (.*)"
  regex-replace="DTAP MM $1">gsm_a.dtap_msg_mm_type</opcode>
  <param>gsm_a.cld_party_bcd_num</param>
  <param>gsm_a.numbering_plan_id</param>
  <param>gsm_a_dtap.cause</param>
  <param>gsm_a.imsi</param>
  <remark>frame</remark>
</sctp-message>
```

You may also specify a <remark> tag to display a field next to the message. If no tag is specified, the time of message receive will be displayed.

# Attributes

## Bookmark messages

Color the messages in a combination of Green and Khaki colors.

Bookmark the message in PDF for easy access.

```
<udp-message style="greenkhaki" bookmark="true" >
  <opcode regex-match="\(query\)$" >dns</opcode>
  <param>dns.qry.name</param>
  <param>dns.resp.name</param>
  <param>dns.Addr</param>
</udp-message>
```

Only match DNS messages that end with the string "(query)"

## Substitute default Wireshark text with regular expressions

```
<sctp-message style="redblue">
  <opcode regex-match="procedureCode: id-(.*)"
  regex-replace="RANAP $1">ranap.procedureCode</opcode>
  <param>ranap.pLMNidentity</param>
  <param>ranap.id</param>
  <param>gsm_a.imsi</param>
</sctp-message>
```

Replace Wireshark text. The extracted part is substituted with \$1.

## Filter out periodic messages

```
<!-- Display the beacon message, but filter out duplicates -->
<wifi-message filter="true" style="grey">
  <opcode regex-match="Type/Subtype: Beacon (.*)"
    regex-replace="IEEE 802.11: Beacon $1">wlan.fc.type_subtype</opcode>
  <param>wlan.seq</param>
  <param regex-match="Tag interpretation: (.*)"
    regex-replace="$1">wlan.tag.interpretation</param>
  <param>data.len</param>
</wifi-message>
```

Filter out periodic and voice traffic by specifying the filter attribute.

## Choose the field to select from multiple occurrences in a message

Use the skip attribute to ignore the specified number of occurrences of a field code. Use the skip attribute to extract tunneled messages.

```
<?xml version="1.0" encoding="utf-8"?>
<FXT>
  <message>
    <opcode>icmp.type</opcode>
    <param skip="1">ip.len</param>
    <source>
      <address skip="1">ip.src</address>
    </source>
    <destination>
      <address skip="1">ip.dst</address>
    </destination>
  </message>
</FXT>
```

Use the skip attribute ignore the first occurrence of the field.

# Regular expressions

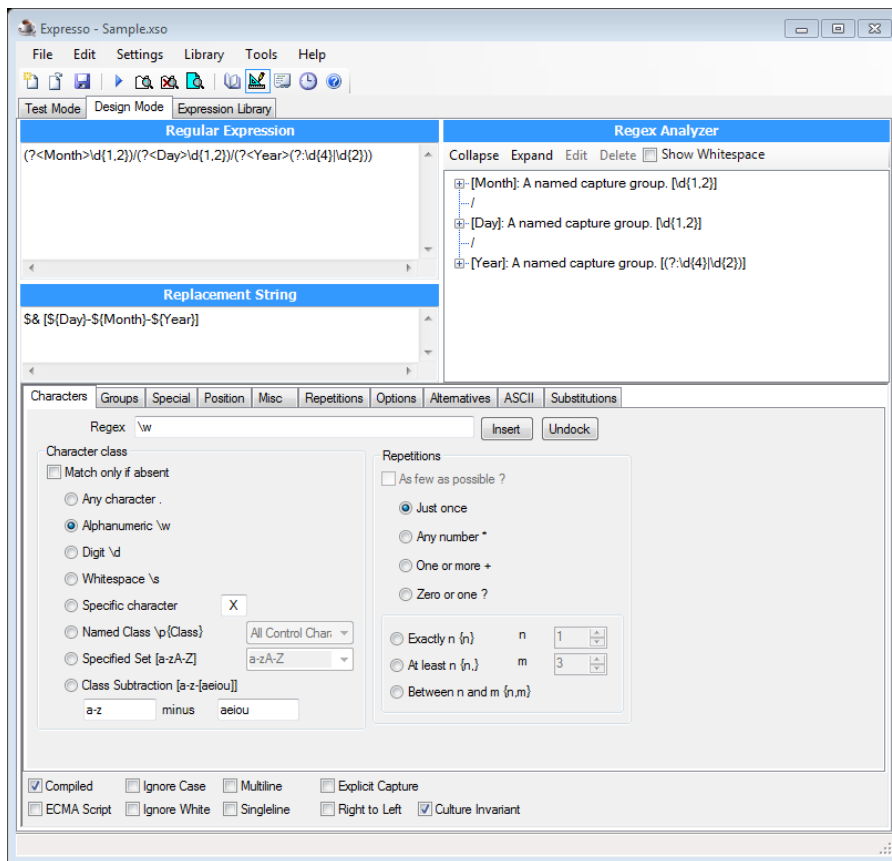
The samples included with VisualEther should be suitable for a large variety of matching and searching scenarios. For more complicated needs we recommend:

Regular expression – quick reference

<http://msdn.microsoft.com/en-us/library/az24scfc.aspx>

Free regular expression tool - Espresso

<http://www.ultrapico.com/expresso.htm>

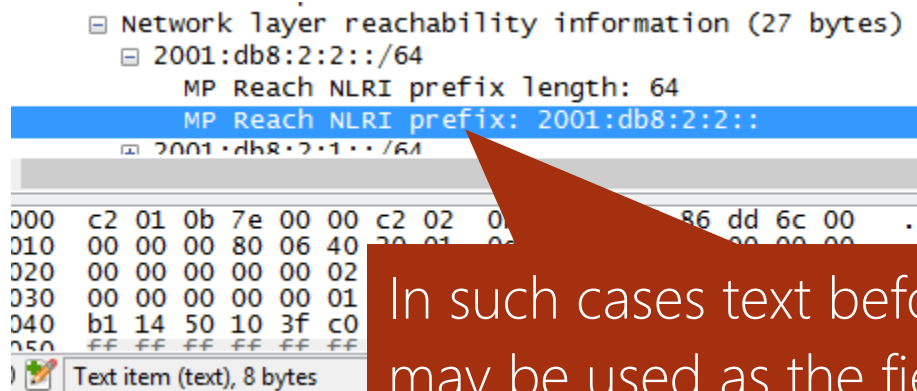




# Working around incomplete Wireshark field definitions

In rare cases, you will find that Wireshark does not have the correct field definition.

For example, the MP Reach NLRI fields do not have a field name (normally field name is displayed in the status bar).









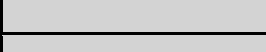
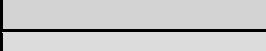

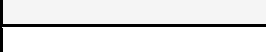



















In such cases text before the colon may be used as the field name.











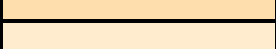








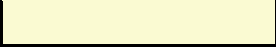


```
<param>bgp.MP Reach NLRI prefix length</param>
<param>bgp.MP Reach NLRI prefix</param>
```

# Colors










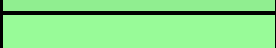


























A handy reference for predefined colors in EventStudio. Use these definitions to define your own styles in VisualEther.fdl file.

BLACK	"0.0,0.0,0.0"	
DIMGGRAY	"0.41,0.41,0.41"	
DIMGREY	"0.41,0.41,0.41"	
GRAY	"0.50,0.50,0.50"	
GREY	"0.50,0.50,0.50"	
DARKGREY	"0.66,0.66,0.66"	
DARKGRAY	"0.66,0.66,0.66"	
SILVER	"0.75,0.75,0.75"	
LIGHTGRAY	"0.83,0.83,0.83"	
LIGHTGREY	"0.83,0.83,0.83"	
GAINSBORO	"0.86,0.86,0.86"	
WHITESMOKE	"0.96,0.96,0.96"	
WHITE	"1.00,1.00,1.00"	
ROSYBROWN	"0.74,0.56,0.56"	
INDIANRED	"0.80,0.36,0.36"	
BROWN	"0.65,0.16,0.16"	
FIREBRICK	"0.70,0.13,0.13"	
LIGHTCORAL	"0.94,0.50,0.50"	
MAROON	"0.50,0.0,0.0"	
DARKRED	"0.55,0.0,0.0"	
RED	"1.00,0.0,0.0"	
SNOW	"1.00,0.98,0.98"	
SALMON	"0.98,0.50,0.45"	
MISTYROSE	"1.00,0.89,0.88"	
TOMATO	"1.00,0.39,0.28"	
DARKSALMON	"0.91,0.59,0.48"	
ORANGERED	"1.00,0.27,0.0"	
CORAL	"1.00,0.50,0.31"	
LIGHTSALMON	"1.00,0.63,0.48"	




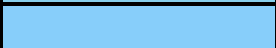







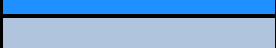
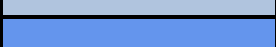























VISUALETHER PROTOCOL ANALYZER 7.2

SIENNA	"0.63,0.32,0.18"	
CHOCOLATE	"0.82,0.41,0.12"	
SADDLEBROWN	"0.55,0.27,0.7"	
SEASHELL	"1.00,0.96,0.93"	
SANDYBROWN	"0.96,0.64,0.38"	
PEACHPUFF	"1.00,0.85,0.73"	
PERU	"0.80,0.52,0.25"	
LINEN	"0.98,0.94,0.90"	
DARKORANGE	"1.00,0.55,0.0"	
BISQUE	"1.00,0.89,0.77"	
TAN	"0.82,0.71,0.55"	
BURLYWOOD	"0.87,0.72,0.53"	
ANTIQUWHITE	"0.98,0.92,0.84"	
NAVAJOWHITE	"1.00,0.87,0.68"	
BLANCHEDALMOND	"1.00,0.92,0.80"	
PAPAYAWHIP	"1.00,0.94,0.84"	
MOCCASIN	"1.00,0.89,0.71"	
WHEAT	"0.96,0.87,0.70"	
OLDLACE	"0.99,0.96,0.90"	
ORANGE	"1.00,0.65,0.0"	
FLORALWHITE	"1.00,0.98,0.94"	
GOLDENROD	"0.85,0.65,0.13"	
DARKGOLDENROD	"0.72,0.53,0.4"	
CORNSILK	"1.00,0.97,0.86"	
GOLD	"1.00,0.84,0.0"	
KHAKI	"0.94,0.90,0.55"	
LEMONCHIFFON	"1.00,0.98,0.80"	
PALEGOLDENROD	"0.93,0.91,0.67"	
DARKKHAKI	"0.74,0.72,0.42"	
BEIGE	"0.96,0.96,0.86"	
LIGHTGOLDENRODYELLOW	"0.98,0.98,0.82"	
OLIVE	"0.50,0.50,0.0"	
YELLOW	"1.00,1.00,0.0"	
LIGHTYELLOW	"1.00,1.00,0.88"	
IVORY	"1.00,1.00,0.94"	
OLIVEDRAB	"0.42,0.56,0.14"	










VISUALETHER PROTOCOL ANALYZER 7.2

YELLOWGREEN	"0.60,0.80,0.20"	
DARKOLIVEGREEN	"0.33,0.42,0.18"	
GREENYELLOW	"0.68,1.00,0.18"	
LAWNGREEN	"0.49,0.99,0.0"	
CHARTREUSE	"0.50,1.00,0.0"	
DARKSEAGREEN	"0.56,0.74,0.56"	
FORESTGREEN	"0.13,0.55,0.13"	
LIMEGREEN	"0.20,0.80,0.20"	
LIGHTGREEN	"0.56,0.93,0.56"	
PALEGREEN	"0.60,0.98,0.60"	
DARKGREEN	"0.0,0.39,0.0"	
GREEN	"0.0,0.50,0.0"	
LIME	"0.0,1.00,0.0"	
HONEYDEW	"0.94,1.00,0.94"	
SEAGREEN	"0.18,0.55,0.34"	
MEDIUMSEAGREEN	"0.24,0.70,0.44"	
SPRINGGREEN	"0.0,1.00,0.50"	
MINTCREAM	"0.96,1.00,0.98"	
MEDIUMSPRINGGREEN	"0.0,0.98,0.60"	
MEDIUMAQUAMARINE	"0.40,0.80,0.67"	
AQUAMARINE	"0.50,1.00,0.83"	
TURQUOISE	"0.25,0.88,0.82"	
LIGHTSEAGREEN	"0.13,0.70,0.67"	
MEDIUMTURQUOISE	"0.28,0.82,0.80"	
DARKSLATEGRAY	"0.18,0.31,0.31"	
DARKSLATEGREY	"0.18,0.31,0.31"	
PALETURQUOISE	"0.69,0.93,0.93"	
TEAL	"0.0,0.50,0.50"	
DARKCYAN	"0.0,0.55,0.55"	
AQUA	"0.0,1.00,1.00"	
CYAN	"0.0,1.00,1.00"	
LIGHTCYAN	"0.88,1.00,1.00"	
AZURE	"0.94,1.00,1.00"	
DARKTURQUOISE	"0.0,0.81,0.82"	
CADETBLUE	"0.37,0.62,0.63"	
POWDERBLUE	"0.69,0.88,0.90"	

VISUALETHER PROTOCOL ANALYZER 7.2

LIGHTBLUE	"0.68,0.85,0.90"	
DEEPSKYBLUE	"0.0,0.75,1.00"	
SKYBLUE	"0.53,0.81,0.92"	
LIGHTSKYBLUE	"0.53,0.81,0.98"	
STEELBLUE	"0.27,0.51,0.71"	
ALICEBLUE	"0.94,0.97,1.00"	
SLATEGREY	"0.44,0.50,0.56"	
SLATEGRAY	"0.44,0.50,0.56"	
LIGHTSLATEGREY	"0.47,0.53,0.60"	
LIGHTSLATEGRAY	"0.47,0.53,0.60"	
DODGERBLUE	"0.12,0.56,1.00"	
LIGHTSTEELBLUE	"0.69,0.77,0.87"	
CORNFLOWERBLUE	"0.39,0.58,0.93"	
ROYALBLUE	"0.25,0.41,0.88"	
MIDNIGHTBLUE	"0.10,0.10,0.44"	
LAVENDER	"0.90,0.90,0.98"	
NAVY	"0.0,0.0,0.50"	
DARKBLUE	"0.0,0.0,0.55"	
MEDIUMBLUE	"0.0,0.0,0.80"	
BLUE	"0.0,0.0,1.00"	
GHOSTWHITE	"0.97,0.97,1.00"	
DARKSLATEBLUE	"0.28,0.24,0.55"	
SLATEBLUE	"0.42,0.35,0.80"	
MEDIUMSLATEBLUE	"0.48,0.41,0.93"	
MEDIUMPURPLE	"0.58,0.44,0.86"	
BLUEVIOLET	"0.54,0.17,0.89"	
INDIGO	"0.29,0.0,0.51"	
DARKORCHID	"0.60,0.20,0.80"	
DARKVIOLET	"0.58,0.0,0.83"	
MEDIUMORCHID	"0.73,0.33,0.83"	
THISTLE	"0.85,0.75,0.85"	
PLUM	"0.87,0.63,0.87"	
VIOLET	"0.93,0.51,0.93"	
PURPLE	"0.50,0.0,0.50"	
DARKMAGENTA	"0.55,0.0,0.55"	
FUCHSIA	"1.00,0.0,1.00"	

VISUALETHER PROTOCOL ANALYZER 7.2

MAGENTA	"1.00,0.0,1.00"	
ORCHID	"0.85,0.44,0.84"	
MEDIUMVIOLETRED	"0.78,0.08,0.52"	
DEEPPINK	"1.00,0.08,0.58"	
HOTPINK	"1.00,0.41,0.71"	
PALEVIOLETRED	"0.86,0.44,0.58"	
LAVENDERBLUSH	"1.00,0.94,0.96"	
CRIMSON	"0.86,0.08,0.24"	
PINK	"1.00,0.75,0.80"	
LIGHTPINK	"1.00,0.71,0.76"	