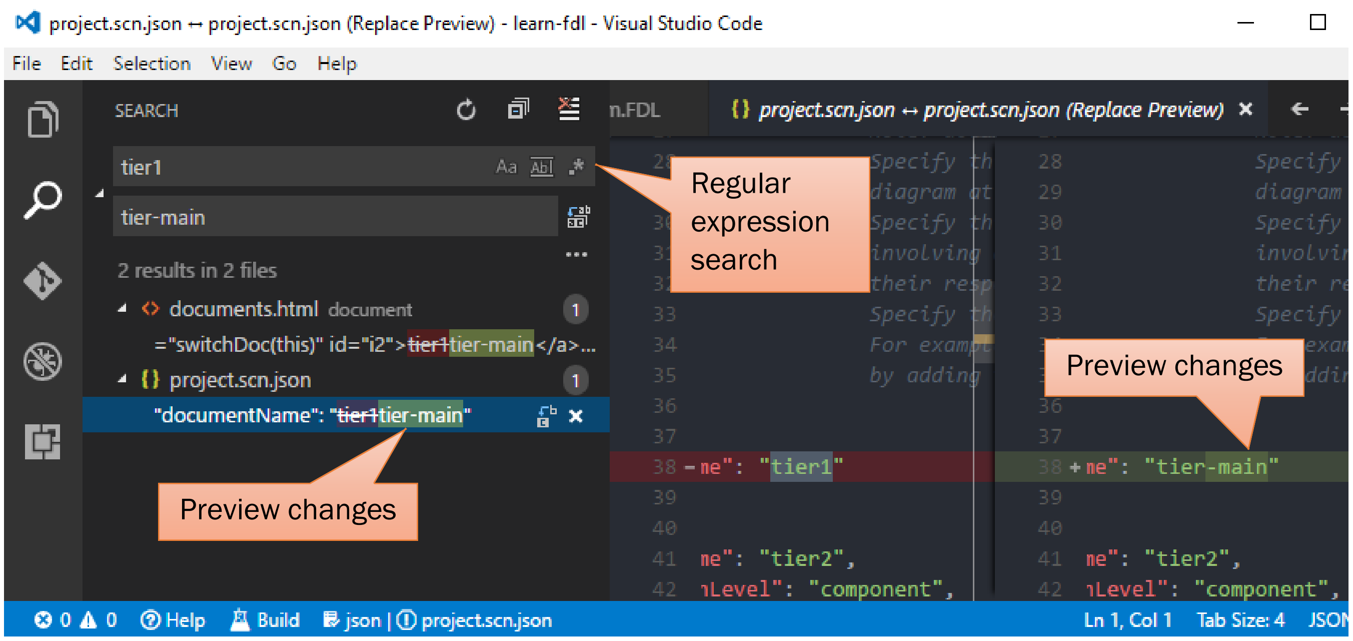
What's New in EventStudio 7

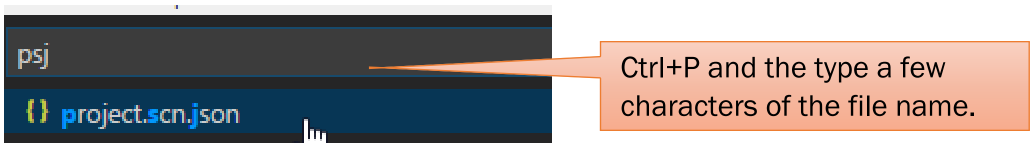
1 NEW USER INTERFACE BASED ON VISUAL STUDIO CODE

1.1 Project wide search and replace

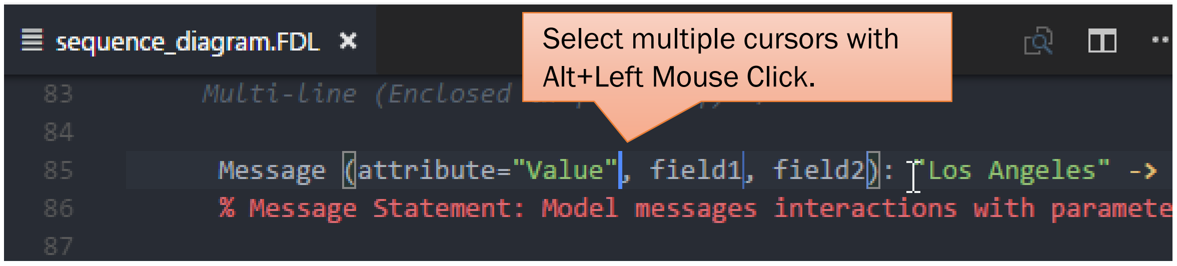


project.scn.json ↔ project.scn.json (Replace Preview) - learn-fdl - Visual Studio Code
File Edit Selection View Go Help
SEARCH
tier1
tier-main
2 results in 2 files
documents.html document
="switchDoc(this)" id="i2">~~tier1~~tier-main</a>...
project.scn.json
"documentName": "~~tier1~~tier-main"
Regular expression search
Preview changes
Preview changes
- ne": "tier1"
+ ne": "tier-main"
ne": "tier2",
Level": "component",
0 0 Help Build json | project.scn.json
Ln 1, Col 1 Tab Size: 4 JSON

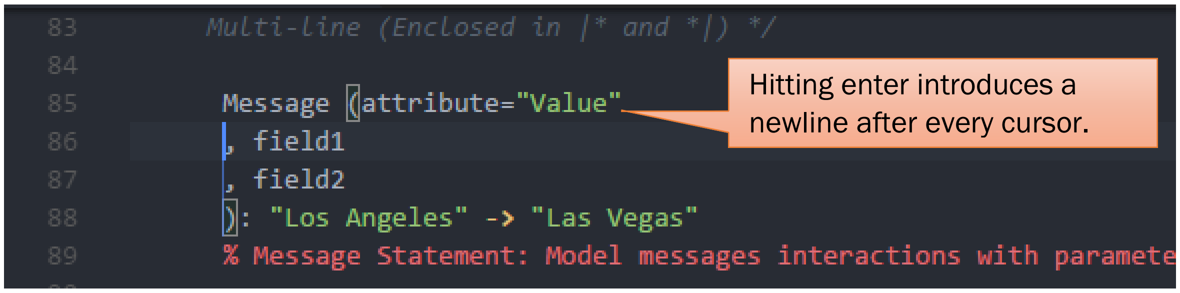
1.2 Quick file open with fuzzy search



1.3 Multi-cursor editing

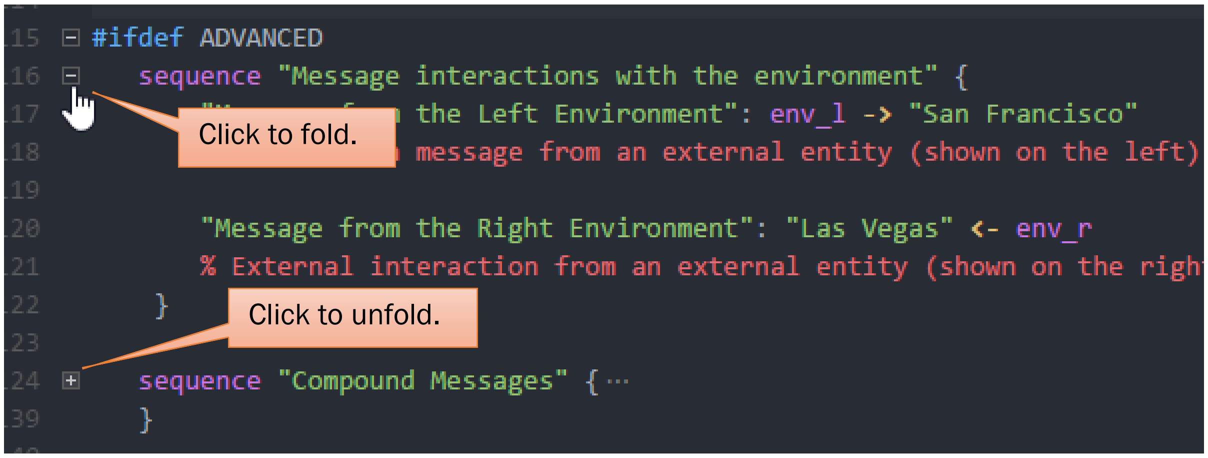


```
Multi-line (Enclosed
Message (attribute="Value", field1, field2): "Los Angeles" ->
% Message Statement: Model messages interactions with paramete
```



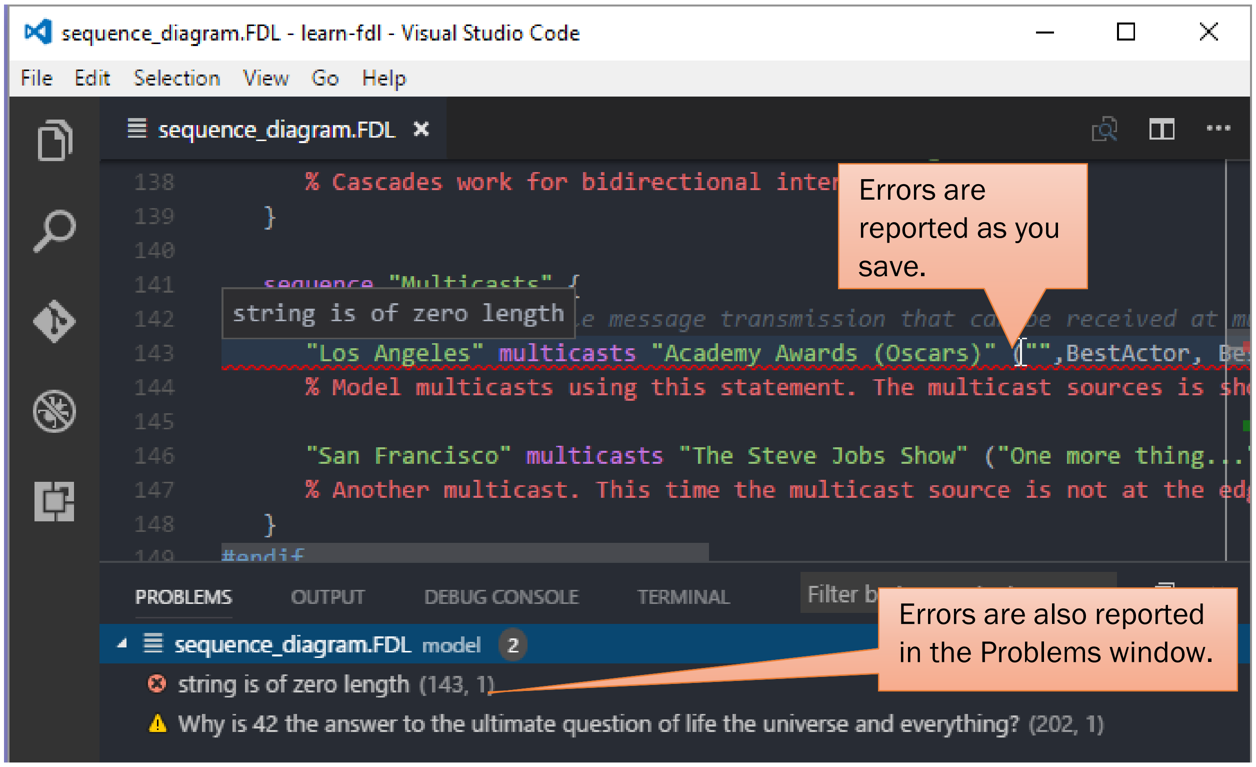
```
        Multi-line (Enclosed in |* and *|) */

        Message (attribute="Value"
        , field1
        , field2
        ): "Los Angeles" -> "Las Vegas"
        % Message Statement: Model messages interactions with paramete
```

1.4 Folding



```
#ifdef ADVANCED
    sequence "Message interactions with the environment" {
        "M              the Left Environment": env_l -> "San Francisco"
                        message from an external entity (shown on the left)

        "Message from the Right Environment": "Las Vegas" <- env_r
        % External interaction from an external entity (shown on the righ
    }

    sequence "Compound Messages" {···
    }
```

1.5 In-line error reporting



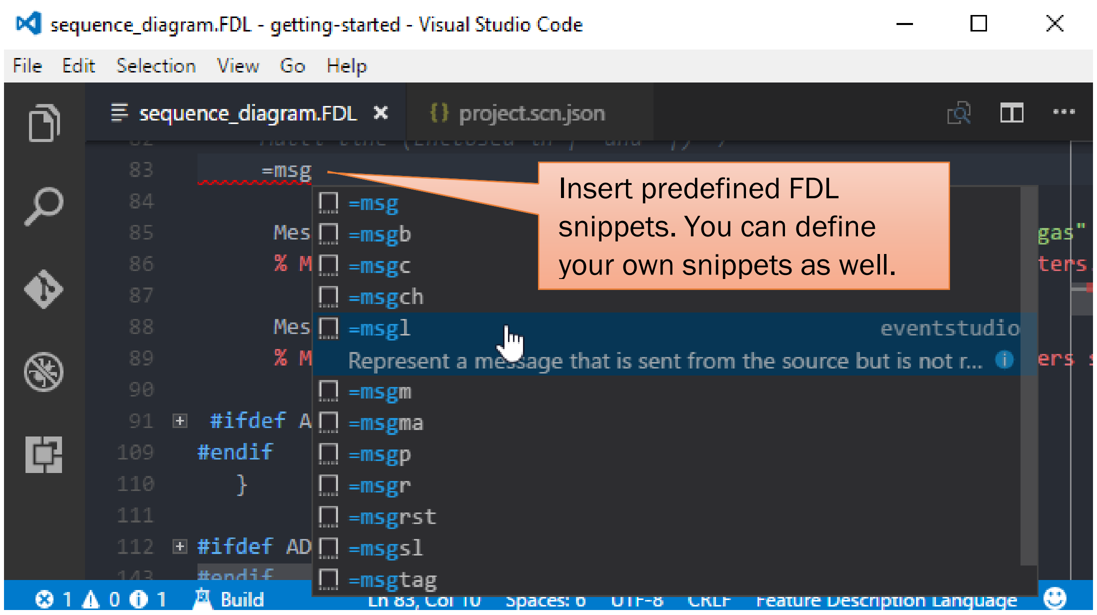
sequence_diagram.FDL - learn-fdl - Visual Studio Code
File Edit Selection View Go Help
sequence_diagram.FDL

```
138         % Cascades work for bidirectional inter
139     }
140
141     sequence "Multicasts" {
142     string is of zero length   e message transmission that ca    be received at m
143         "Los Angeles" multicasts "Academy Awards (Oscars)" ("",BestActor, Be
144         % Model multicasts using this statement. The multicast sources is sh
145
146         "San Francisco" multicasts "The Steve Jobs Show" ("One more thing...
147         % Another multicast. This time the multicast source is not at the ed
148     }
149 #endif
```

```
sequence_diagram.FDL model 2
  string is of zero length (143, 1)
  Why is 42 the answer to the ultimate question of life the universe and everything? (202, 1)
```

Errors are reported as you save.
Errors are also reported in the Problems window.

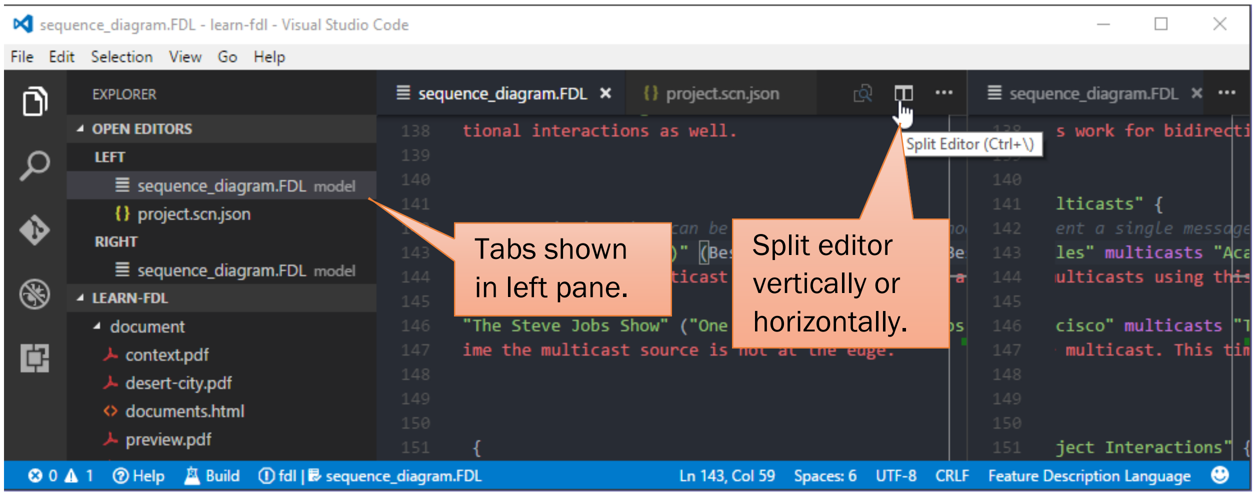
1.6 Snippets


sequence_diagram.FDL - getting-started - Visual Studio Code
File Edit Selection View Go Help
sequence_diagram.FDL
project.scn.json
Build
Ln 83, Col 10 Spaces: 6 UTF-8 CRLF Feature Description Language

Insert predefined FDL snippets. You can define your own snippets as well.

```
=msg
=msg
=msgb
=msgc
=msgch
=msgl                                        eventstudio
Represent a message that is sent from the source but is not r...
=msgm
=msgma
=msgp
=msgr
=msgrst
=msgsl
=msgtag
```

1.7 Tabs and split Editor

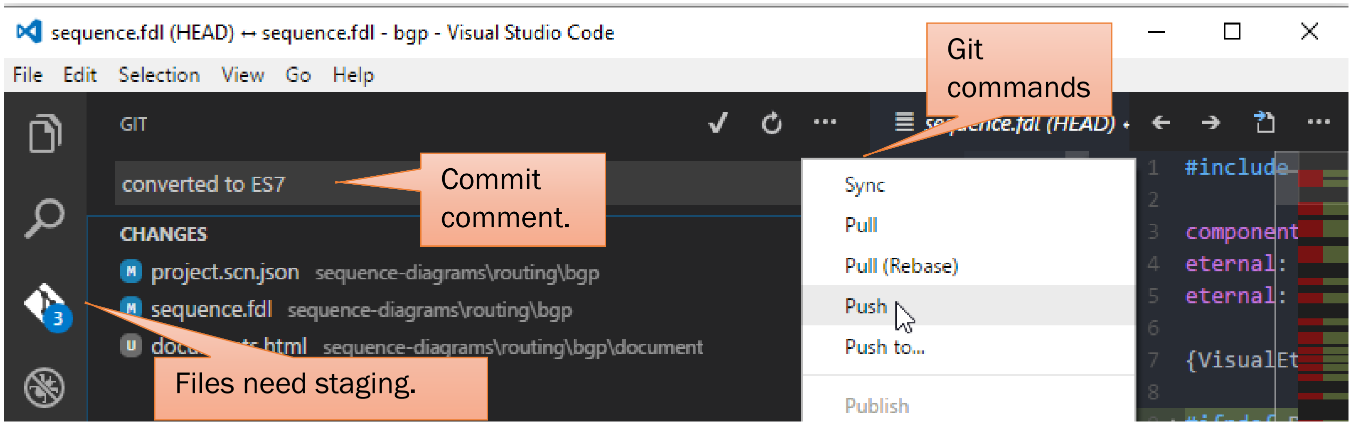

sequence_diagram.FDL - learn-fdl - Visual Studio Code
File Edit Selection View Go Help
EXPLORER
OPEN EDITORS
LEFT
sequence_diagram.FDL model
project.scn.json
RIGHT
sequence_diagram.FDL model
LEARN-FDL
document
context.pdf
desert-city.pdf
documents.html
preview.pdf
sequence_diagram.FDL
project.scn.json
Split Editor (Ctrl+\)
Help
Build
fdl | sequence_diagram.FDL
Ln 143, Col 59 Spaces: 6 UTF-8 CRLF Feature Description Language

Tabs shown in left pane.

Split editor vertically or horizontally.

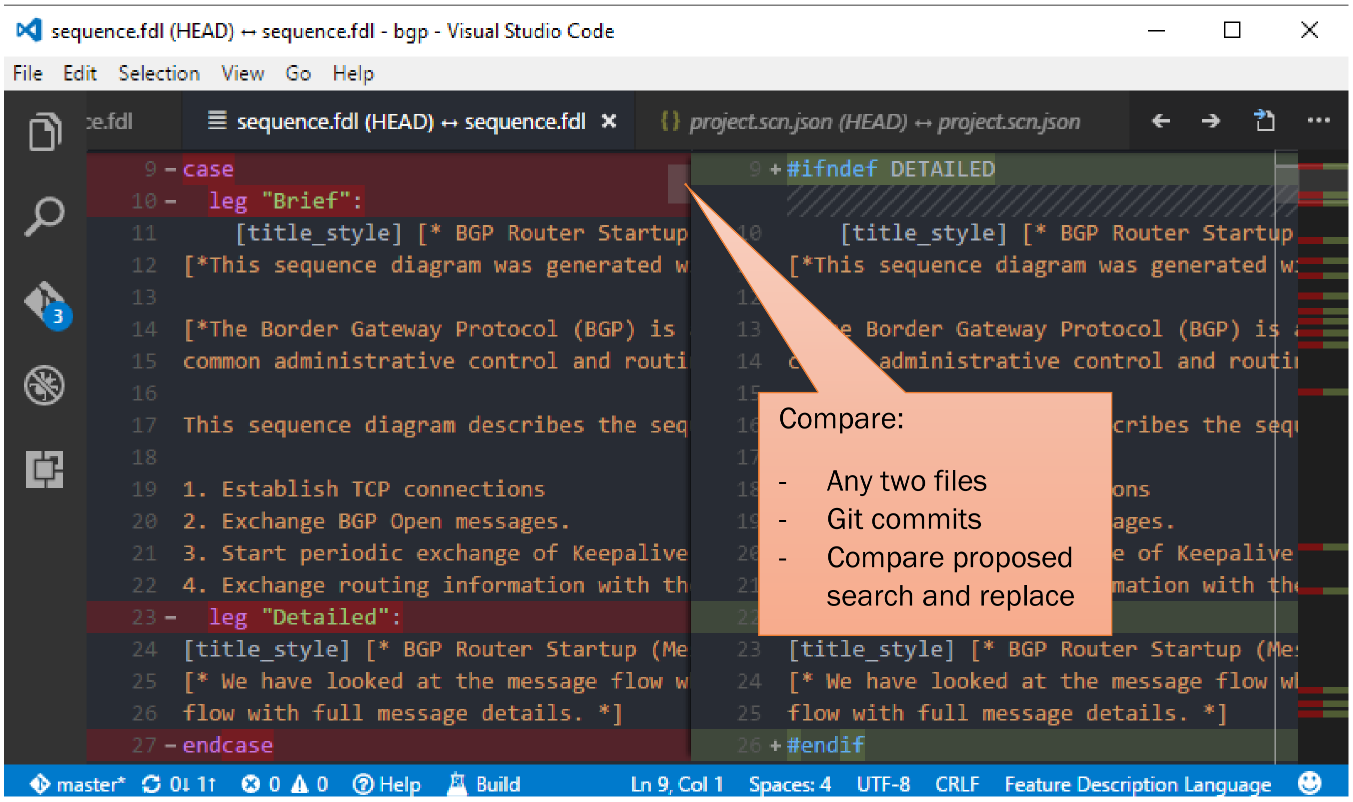
```
tional interactions as well.
"The Steve Jobs Show" ("One
ime the multicast source is not at the edge.
```

1.8 Built in Git support


sequence.fdl (HEAD) ↔ sequence.fdl - bgp - Visual Studio Code
File Edit Selection View Go Help
GIT
converted to ES7
Commit comment.
CHANGES
project.scn.json sequence-diagrams\routing\bgp
sequence.fdl sequence-diagrams\routing\bgp
document.html sequence-diagrams\routing\bgp\document
Files need staging.
Git commands
Sync
Pull
Pull (Rebase)
Push
Push to...
Publish

```
#include
component
eternal:
eternal:
{VisualEt
```

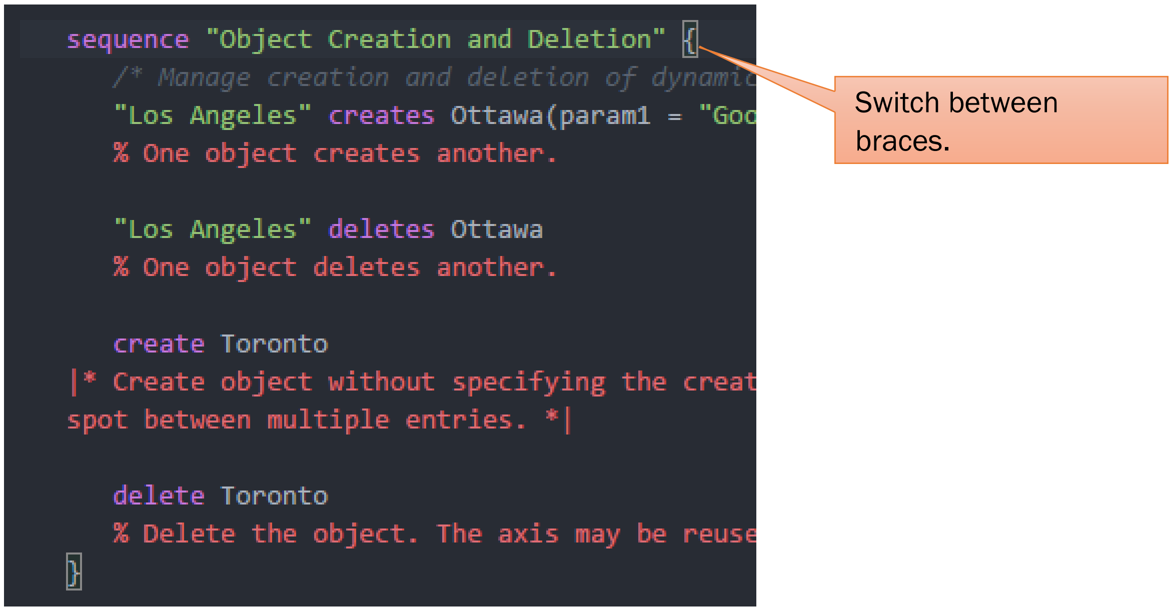
1.9 Compare files


sequence.fdl (HEAD) ↔ sequence.fdl - bgp - Visual Studio Code
File Edit Selection View Go Help
sequence.fdl (HEAD) ↔ sequence.fdl
project.scn.json (HEAD) ↔ project.scn.json
Compare:
- Any two files
- Git commits
- Compare proposed search and replace
master* 0↓ 1↑ 0 0 Help Build Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Feature Description Language

```
- case
-   leg "Brief":
      [title_style] [* BGP Router Startup
  [*This sequence diagram was generated w

  [*The Border Gateway Protocol (BGP) is
  common administrative control and routi

  This sequence diagram describes the seq

  1. Establish TCP connections
  2. Exchange BGP Open messages.
  3. Start periodic exchange of Keepalive
  4. Exchange routing information with th
-   leg "Detailed":
  [title_style] [* BGP Router Startup (Me
  [* We have looked at the message flow w
  flow with full message details. *]
- endcase
```

```
+ #ifndef DETAILED
      [title_style] [* BGP Router Startup
  [*This sequence diagram was generated w

  [*The Border Gateway Protocol (BGP) is
  common administrative control and routi

  This sequence diagram describes the seq

  1. Establish TCP connections
  2. Exchange BGP Open messages.
  3. Start periodic exchange of Keepalive
  4. Exchange routing information with the

  [title_style] [* BGP Router Startup (Mes
  [* We have looked at the message flow wi
  flow with full message details. *]
+ #endif
```

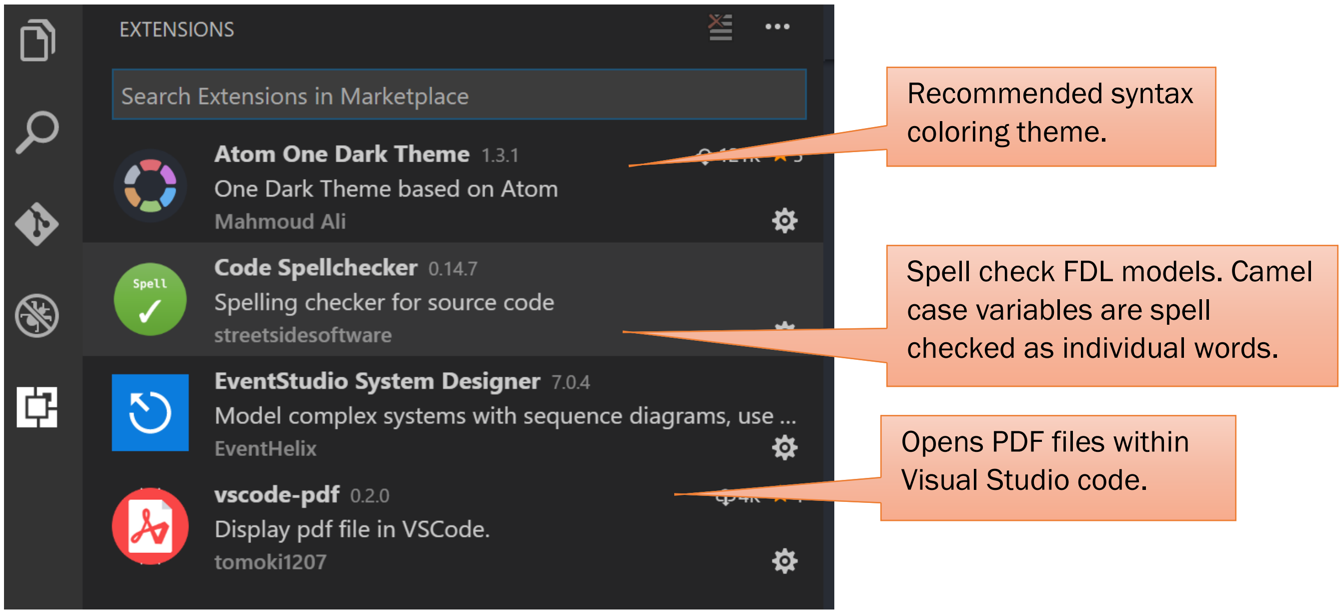
1.10 Brace and comment end matching



```
sequence "Object Creation and Deletion" {
    /* Manage creation and deletion of dynamic
    "Los Angeles" creates Ottawa(param1 = "Goo
    % One object creates another.

    "Los Angeles" deletes Ottawa
    % One object deletes another.

    create Toronto
|* Create object without specifying the creat
spot between multiple entries. *|

    delete Toronto
    % Delete the object. The axis may be reuse
}
```

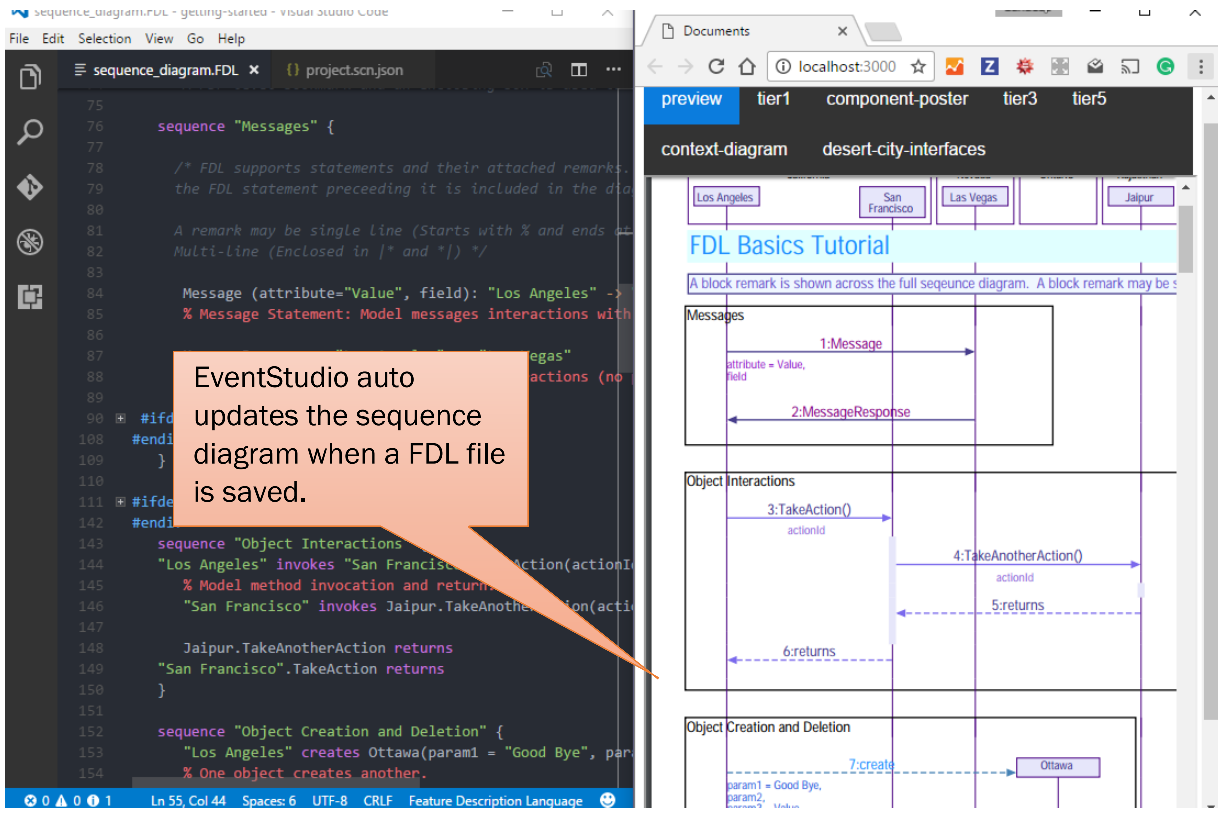
1.11 Extensions

Extend your EventStudio experience with Visual Studio Code extensions. Here are a few extensions that we have found useful.



EXTENSIONS
Search Extensions in Marketplace
Atom One Dark Theme 1.3.1
One Dark Theme based on Atom
Mahmoud Ali
Code Spellchecker 0.14.7
Spelling checker for source code
streetsidesoftware
EventStudio System Designer 7.0.4
Model complex systems with sequence diagrams, use ...
EventHelix
vscode-pdf 0.2.0
Display pdf file in VSCode.
tomoki1207
Recommended syntax coloring theme.
Spell check FDL models. Camel case variables are spell checked as individual words.
Opens PDF files within Visual Studio code.

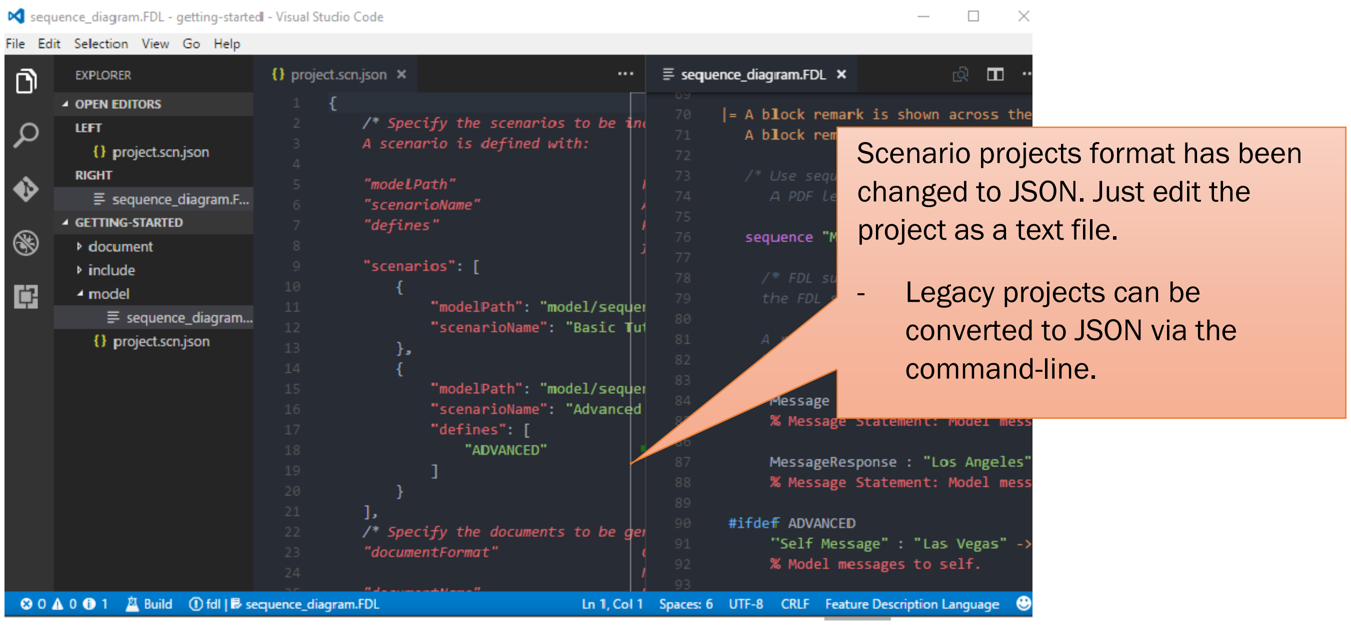
2 AUTO PREVIEW



```
     sequence "Messages" {

       /* FDL supports statements and their attached remarks.
       the FDL statement preceeding it is included in the dia

       A remark may be single line (Starts with % and ends at
       Multi-line (Enclosed in |* and *|) */

        Message (attribute="Value", field): "Los Angeles" ->
        % Message Statement: Model messages interactions with

                                                  egas"
                                                  actions (no

  #ifd
  #endi
     }

  #ifde
  #endi
     sequence "Object Interactions
     "Los Angeles" invokes "San Francisc          Action(actionI
        % Model method invocation and return
        "San Francisco" invokes Jaipur.TakeAnothe        ion(acti

        Jaipur.TakeAnotherAction returns
     "San Francisco".TakeAction returns
     }

     sequence "Object Creation and Deletion" {
        "Los Angeles" creates Ottawa(param1 = "Good Bye", par
        % One object creates another.
```

EventStudio auto updates the sequence diagram when a FDL file is saved.

3 JSON BASED PROJECT FILE


sequence_diagram.FDL - getting-started - Visual Studio Code
File Edit Selection View Go Help
EXPLORER
OPEN EDITORS
LEFT
project.scn.json
RIGHT
sequence_diagram.F...
GETTING-STARTED
document
include
model
sequence_diagram...
project.scn.json
project.scn.json
sequence_diagram.FDL
Build
fdl | sequence_diagram.FDL
Ln 1, Col 1 Spaces: 6 UTF-8 CRLF Feature Description Language

```
{
     /* Specify the scenarios to be inc
     A scenario is defined with:

     "modelPath"
     "scenarioName"
     "defines"

     "scenarios": [
          {
               "modelPath": "model/sequen
               "scenarioName": "Basic Tut
          },
          {
               "modelPath": "model/sequen
               "scenarioName": "Advanced
               "defines": [
                    "ADVANCED"
               ]
          }
     ],
     /* Specify the documents to be gen
     "documentFormat"
```

```
  |= A block remark is shown across the
     A block rem

     /* Use sequ
        A PDF le

     sequence "M

       /* FDL su
       the FDL

       A

        Message
        % Message Statement: Model mess

        MessageResponse : "Los Angeles"
        % Message Statement: Model mess

   #ifdef ADVANCED
        "Self Message" : "Las Vegas" ->
        % Model messages to self.
```

Scenario projects format has been changed to JSON. Just edit the project as a text file.

- Legacy projects can be converted to JSON via the command-line.

4 IMPROVED SUPPORT FOR MULTIPLE SCENARIOS

4.1 Preprocessor defines simplify multiple scenario definition

```
/* Specify the scenarios to be included in the project
A scenario is defined with:

"modelPath"                      FDL file that defines the model.
"scenarioName"                   A title that summarizes the scenario
"defines"                        Preprocessor defines that select the
                                 flow specific to this scenario. */
"scenarios": [
    {
        "modelPath": "model/sequence_diagram.FDL",
        "scenarioName": "Basic Tutorial"
    },
    {
        "modelPath": "model/sequence_diagram.FDL",
        "scenarioName": "Advanced Tutorial",
        "defines": ["ADVANCED"]
    }
],
```

Scenario is identified by the preprocessor symbol ADVANCED.

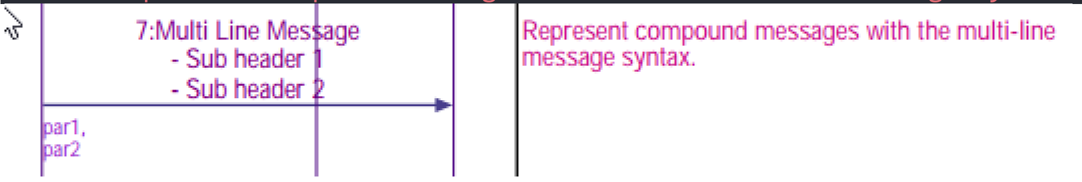
4.2 Specify preprocessor defines for documents

```
                                 their respective type specifications.
"defines"                        Specify the preprocessor defines that are specific
                                 for this document.
                                 For example, "POSTER" and "PRESENTATION" formats
                                 can be selected by adding them to the defines. */
"documents": [
    {
        "documentName": "tier1"
    },
    {
        "documentName": "tier2",
        "interactionLevel": "component",
        "defines": ["POSTER"]
    }
]
```

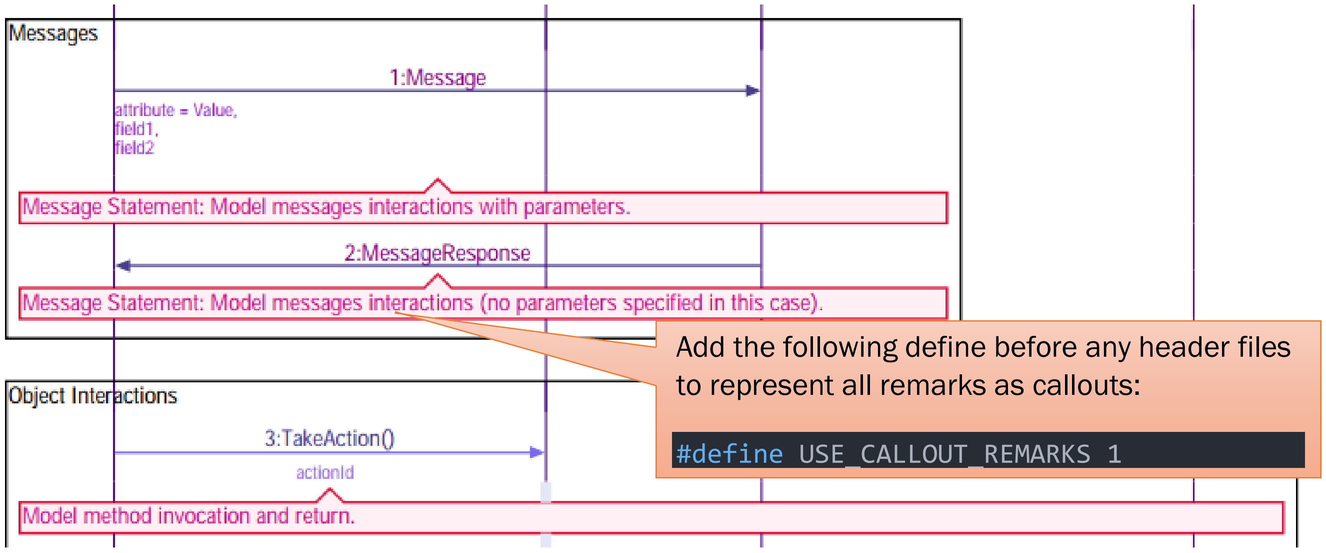
Document characteristics are identified by the preprocessor symbol POSTER.

5 MULTI-LINE MESSAGE OPCODES

```
    "Multi Line Message" +
       "- Sub header 1" +
       "- Sub header 2" (par1, par2):"Las Vegas" -> Johannesburg
    % Represent compound messages with the multi-line message syntax.
```


7:Multi Line Message
- Sub header 1
- Sub header 2
par1,
par2
Represent compound messages with the multi-line message syntax.

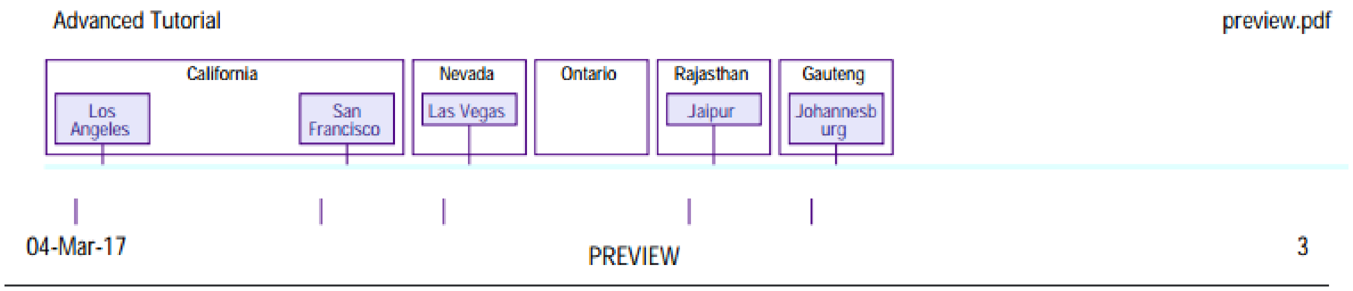
6 CALLOUT COMMENTS


Messages
1:Message
attribute = Value,
field1,
field2
Message Statement: Model messages interactions with parameters.
2:MessageResponse
Message Statement: Model messages interactions (no parameters specified in this case).
Object Interactions
3:TakeAction()
actionId
Model method invocation and return.

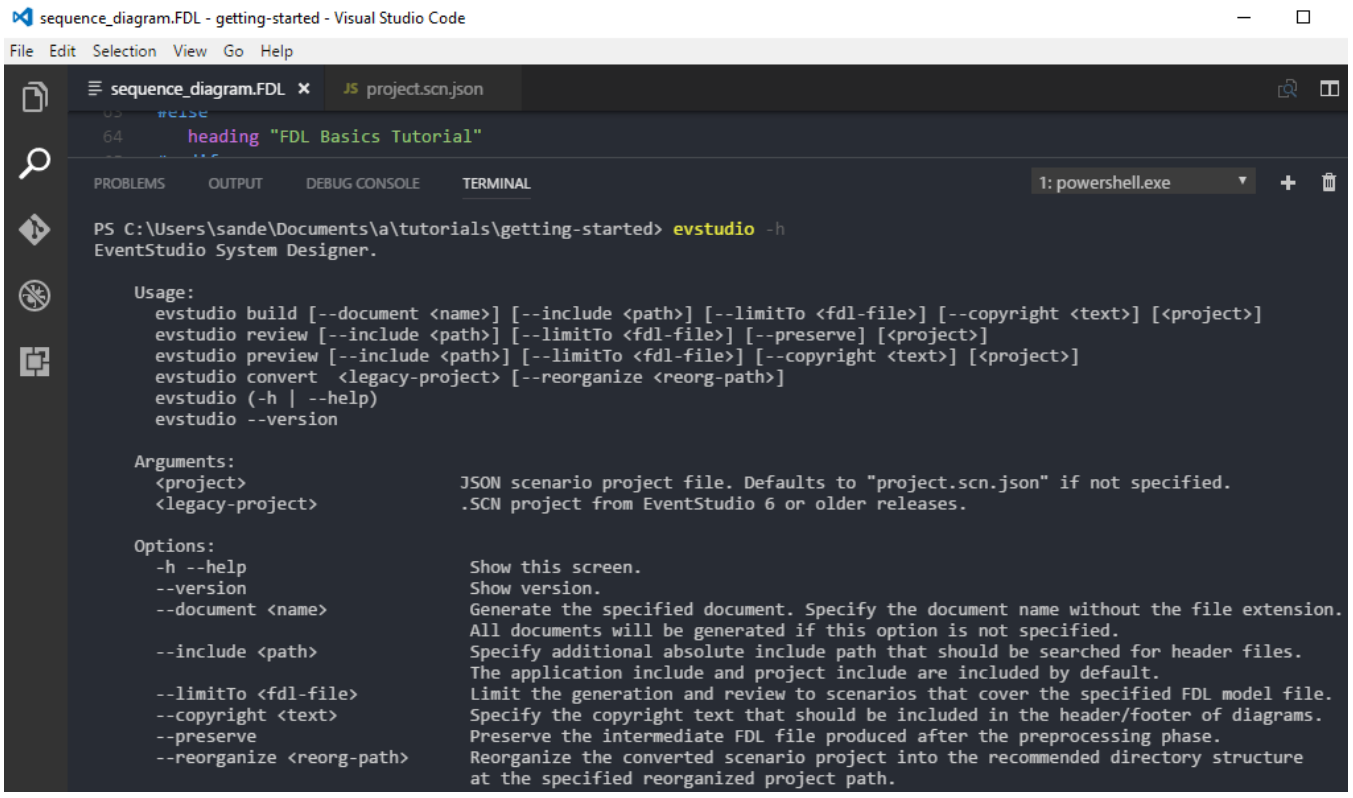
Add the following define before any header files to represent all remarks as callouts:

```
#define USE_CALLOUT_REMARKS 1
```

7 HEADER AND FOOTERS



8 IMPROVED COMMAND LINE SUPPORT


sequence_diagram.FDL - getting-started - Visual Studio Code
File Edit Selection View Go Help
sequence_diagram.FDL
project.scn.json
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell.exe

```
    #else
        heading "FDL Basics Tutorial"
```

```
PS C:\Users\sande\Documents\a\tutorials\getting-started> evstudio -h
EventStudio System Designer.

    Usage:
      evstudio build [--document <name>] [--include <path>] [--limitTo <fdl-file>] [--copyright <text>] [<project>]
      evstudio review [--include <path>] [--limitTo <fdl-file>] [--preserve] [<project>]
      evstudio preview [--include <path>] [--limitTo <fdl-file>] [--copyright <text>] [<project>]
      evstudio convert  <legacy-project> [--reorganize <reorg-path>]
      evstudio (-h | --help)
      evstudio --version

    Arguments:
      <project>                     JSON scenario project file. Defaults to "project.scn.json" if not specified.
      <legacy-project>              .SCN project from EventStudio 6 or older releases.

    Options:
      -h --help                      Show this screen.
      --version                      Show version.
      --document <name>              Generate the specified document. Specify the document name without the file extension.
                                     All documents will be generated if this option is not specified.
      --include <path>               Specify additional absolute include path that should be searched for header files.
                                     The application include and project include are included by default.
      --limitTo <fdl-file>           Limit the generation and review to scenarios that cover the specified FDL model file.
      --copyright <text>             Specify the copyright text that should be included in the header/footer of diagrams.
      --preserve                     Preserve the intermediate FDL file produced after the preprocessing phase.
      --reorganize <reorg-path>      Reorganize the converted scenario project into the recommended directory structure
                                     at the specified reorganized project path.
```

9 SYNTAX IMPROVEMENTS

9.1 Single line remarks

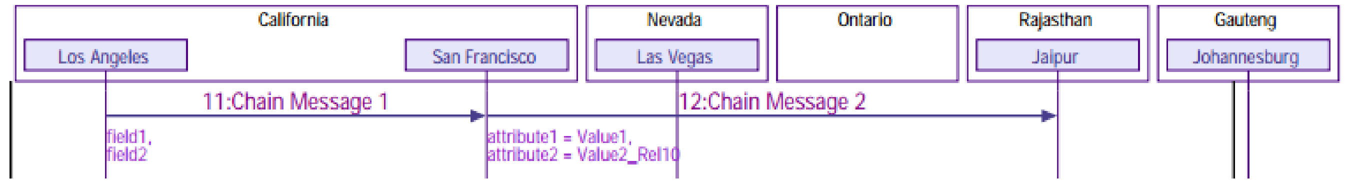
```
"Self Message" : "Las Vegas" -> "Las Vegas"
% Model messages to self.

"Message with Bold Arrow" : "Los Angeles" => Jaipur
% Use the => or <= to represent messages with bold arrows.

"Bidirectional Interaction" : "Los Angeles" <-> Jaipur
% Model bi-directional message interactions with <-> or <=>.
```

Single line remarks begin with the % sign.

9.2 Flexible syntax that permits scenario level customization

```
#define RELEASE_10 1
      chain {
        "Chain Message 1" (field1, field2): "Los Angeles" -> "San Francisco"
        "Chain Message 2"
        (attribute1=Value1,
#ifdef RELEASE_10
        attribute2=Value2_Rel10
#else
        attribute2=Value2
#endif
        ): "San Francisco" -> Jaipur
```

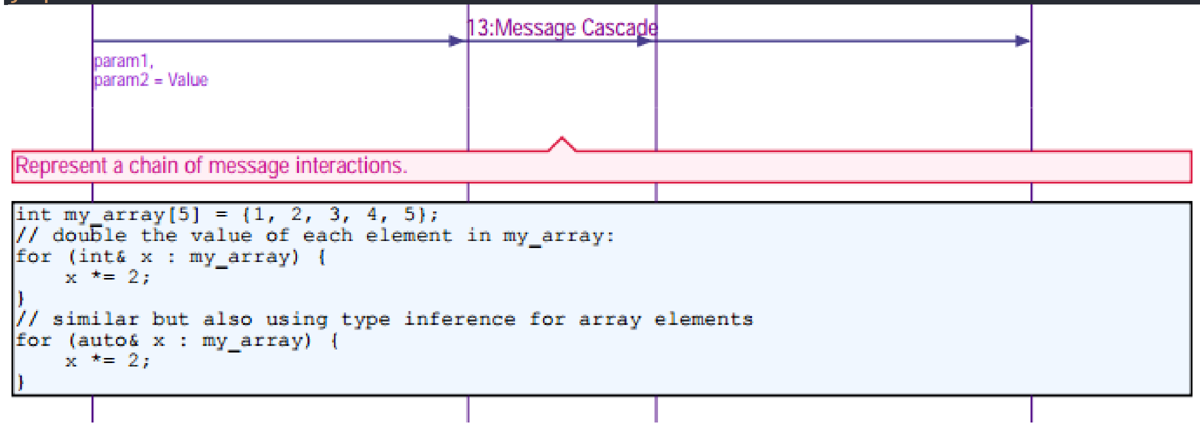
FDL syntax has been enhanced to allow blank lines in several statements. This allows you to define message parameters based on scenario or document defines.



9.3 Insert pseudo code fragments in block remarks

```
      "Message Cascade" (param1, param2="Value"): "Los Angeles" ->
      "San Francisco" -> "Las Vegas" -> Jaipur
      % Represent a chain of message interactions.
      [_code] |=
int my_array[5] = {1, 2, 3, 4, 5};
// double the value of each element in my_array:
for (int& x : my_array) {
    x *= 2;
}
// similar but also using type inference for array elements
for (auto& x : my_array) {
    x *= 2;
}=|
```

You can add pseudo code right into the sequence diagram.



```
int my_array[5] = {1, 2, 3, 4, 5};
// double the value of each element in my_array:
for (int& x : my_array) {
    x *= 2;
}
// similar but also using type inference for array elements
for (auto& x : my_array) {
    x *= 2;
}
```