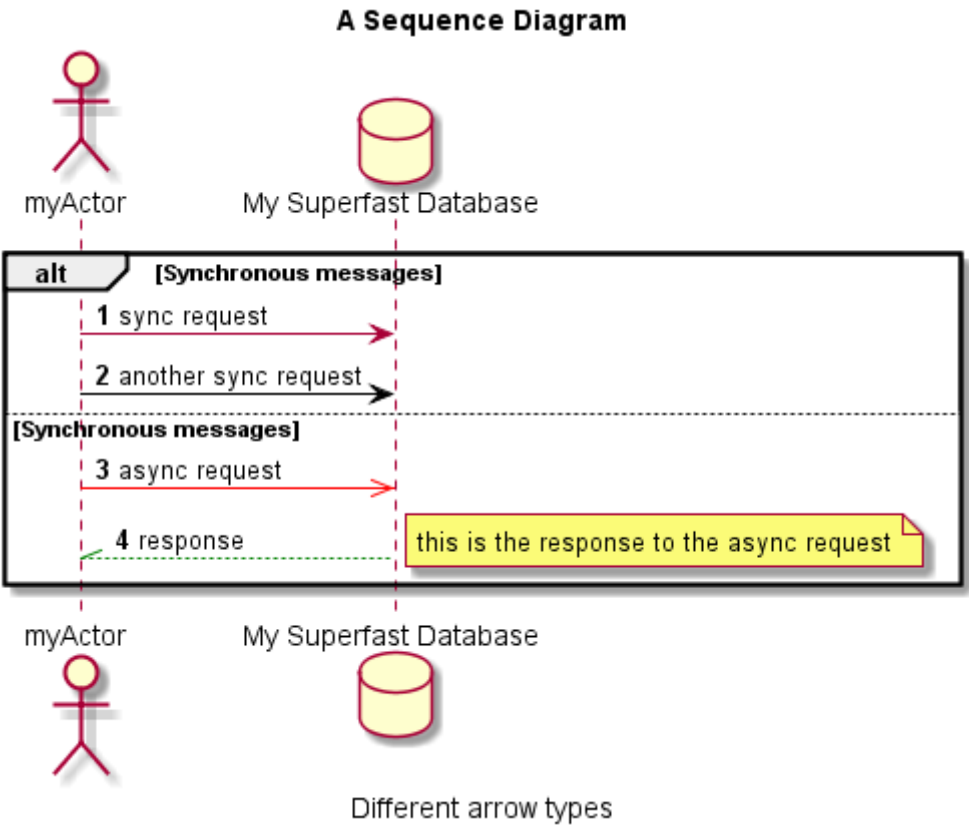


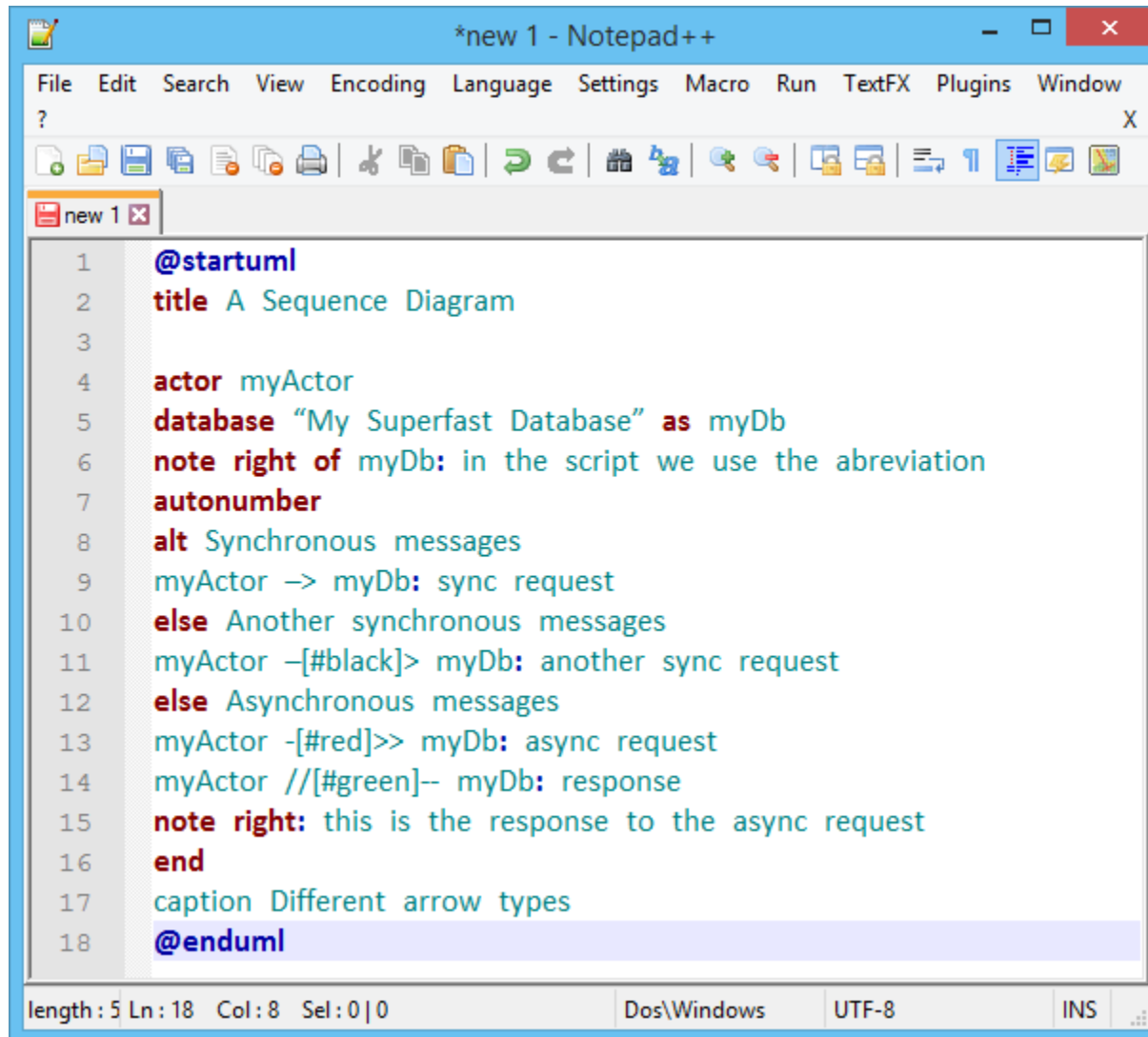
PlantUML

Scripting UML Diagrams

UML Sequence Diagram...



... from a Script



```
*new 1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window
?
new 1 x
1 @startuml
2 title A Sequence Diagram
3
4 actor myActor
5 database "My Superfast Database" as myDb
6 note right of myDb: in the script we use the abbreviation
7 autonumber
8 alt Synchronous messages
9 myActor -> myDb: sync request
10 else Another synchronous messages
11 myActor -[#black]> myDb: another sync request
12 else Asynchronous messages
13 myActor -[#red]>> myDb: async request
14 myActor //[#green]- myDb: response
15 note right: this is the response to the async request
16 end
17 caption Different arrow types
18 @enduml
length: 5 Ln: 18 Col: 8 Sel: 0|0 Dos\Windows UTF-8 INS
```

Supported Diagram Types

Structural Diagrams

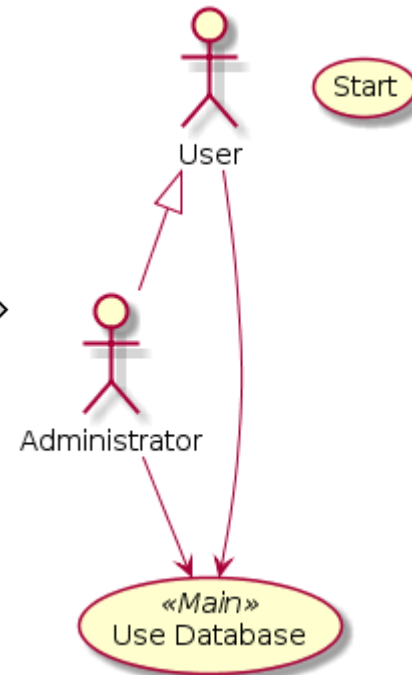
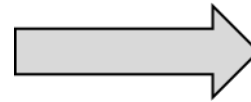
- Component
- Class
- Deployment
- Object

Behavioural Diagrams

- Use Case
- Sequence
- Activity
- State

Use Case Diagram

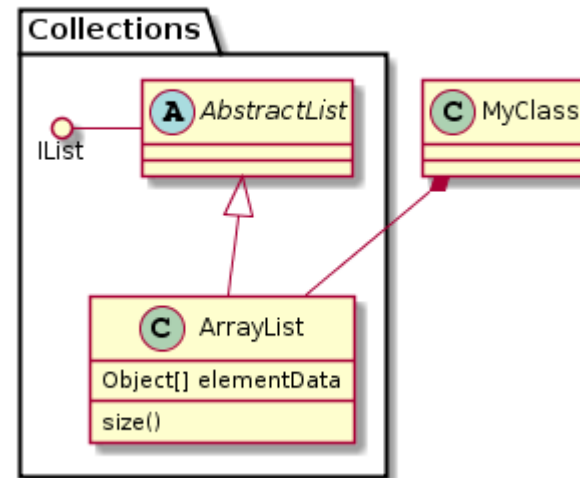
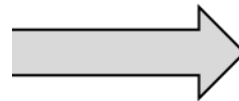
```
@startuml
actor User
actor Administrator as admin
User <|-- admin
(Use Database) as (Use) << Main >>
User --> (Use)
admin --> (Use)
@enduml
```



Class Diagram

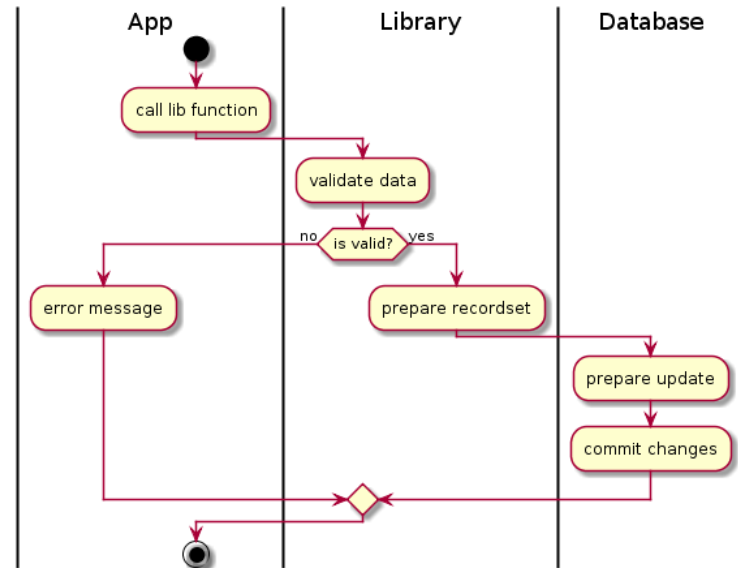
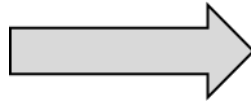
```
@startuml
package "Collections" {
  abstract class AbstractList
  IList ()- AbstractList
  AbstractList <|-- ArrayList
  class ArrayList {
    Object[] elementData
    size()
  }
}

class MyClass
MyClass *- ArrayList
@enduml
```



Activity Diagram

```
@startuml
|App|
start
:call lib function;
|Library|
:validate data;
if(is valid?) then (no)
  |App|
  :error message;
else (yes)
  |Library|
  :prepare recordset;
  |Database|
  :prepare update;
  :commit changes;
endif
|App|
stop
@enduml
```



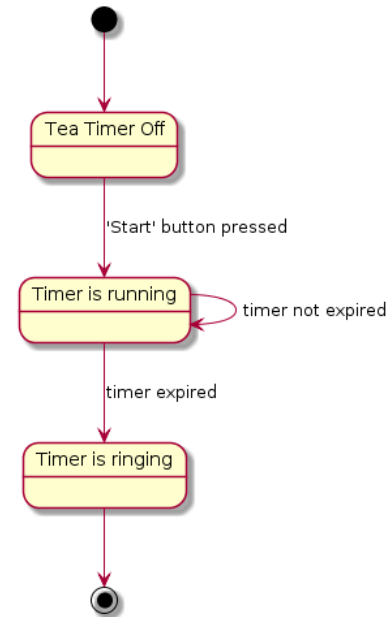
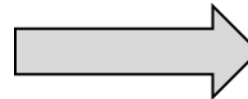
State Diagram

@startuml

```
state "Tea Timer Off" as Toff
state "Timer is running" as Trun
state "Timer is ringing" as Tring
```

```
[*] --> Toff
Toff --> Trun : 'Start' button pressed
Trun --> Trun : timer not expired
Trun --> Tring : timer expired
Tring --> [*]
```

@enduml



Applications using PlantUML



Consult <http://plantuml.com/running.html> for updated list and links to the different tools.

Architecture

Online Editors
(e.g. PlantText)

Doc Generators
(e.g. Doxygen)

Word Template
(e.g. Doxygen)

```

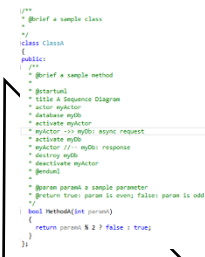
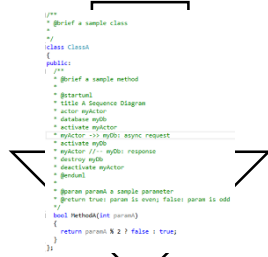
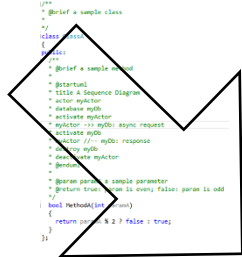
/**
 * @brief a sample class
 */
class MyClass {
public:
    /**
     * @brief a sample method
     */
    MyClass();
    // ...
};
    
```

```

/**
 * @brief a sample class
 */
class MyClass {
public:
    /**
     * @brief a sample method
     */
    MyClass();
    // ...
};
    
```

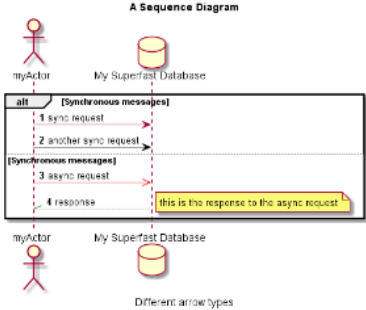
```

/**
 * @brief a sample class
 */
class MyClass {
public:
    /**
     * @brief a sample method
     */
    MyClass();
    // ...
};
    
```



plantuml.jar

Graphviz
dot.exe



Online Editor

The screenshot shows the PlantText online editor interface. On the left is a code editor with the following PlantUML script:

```
1 @startuml
2 title A Sequence Diagram
3
4 actor myActor
5 database "My Superfast Database" as myDb
6 note right of myDb: in the script we use the abbreviation
7 autonumber
8 alt Synchronous messages
9 myActor->>myDb: sync request
10 else Another synchronous messages
11 myActor->>myDb: another sync request
12 else Asynchronous messages
13 myActor->>myDb: async request
14 myDb-->>myActor: response
15 note right: this is the response to the async request
16 end
17 caption Different arrow types
18 @enduml
```

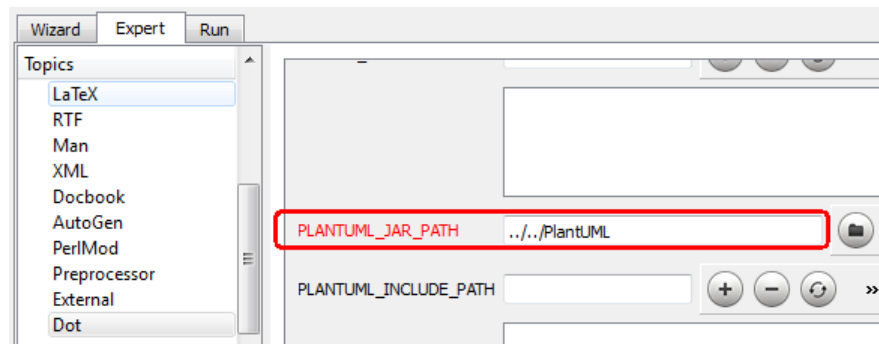
On the right is a sequence diagram titled "A Sequence Diagram" showing the interaction between "myActor" and "My Superfast Database". The diagram includes an alt block with three sections: "[Synchronous messages]", "[Another synchronous messages]", and "[Asynchronous messages]". The messages are: 1 sync request, 2 another sync request, 3 async request, and 4 response. A note on the right says "in the script we use the abbreviation" and another note says "this is the response to the async request". Below the diagram is the caption "Different arrow types" and a row of links: PNG | SVG | TXT | Edit.

Thanks to [PlantUML](#), [Graphviz](#), [Ace Editor](#), [Google App Engine](#), [Python](#), [WebApp2](#), [Jinja2](#), as well as [Steven Nichols](#) and [Brian Folts](#).

© Copyright 2015 [Arwen Vaughan](#).

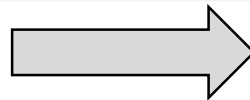
Doxygen

- PlantUML script placed within Doxygen comments in source code
- Doxygen executes PlantUML and places images in generated doc
- ! It is no longer necessary to parse the PlantUML code separately and add references to images in Doxygen comments !
- path to plantuml.jar must be specified in Doxygen configuration



Doxygen

```
/**  
 * @brief a sample class  
 */  
class ClassA  
{  
public:  
    /**  
     * @brief a sample method  
     *  
     * @startuml  
     * title A Sequence Diagram  
     * actor myActor  
     * database myDb  
     * activate myActor  
     * myActor ->> myDb: async request  
     * activate myDb  
     * myActor /-- myDb: response  
     * destroy myDb  
     * deactivate myActor  
     * @enduml  
     *  
     * @param paramA a sample parameter  
     * @return true: param is even; false: param is odd  
     */  
    bool MethodA(int paramA)  
    {  
        return paramA % 2 ? false : true;  
    }  
};
```



A Doxygen Sample: ClassA Class Reference - Mozilla Firefox

A Doxygen Sample: ClassA Clas... x +

file:///J:/Projects/UML_Present/Dox doxygen dowr

Public Member Functions | List of all members

ClassA Class Reference

a sample class

Public Member Functions

bool **MethodA** (int paramA)
a sample method More...

Member Function Documentation

bool ClassA::MethodA (int paramA) inline

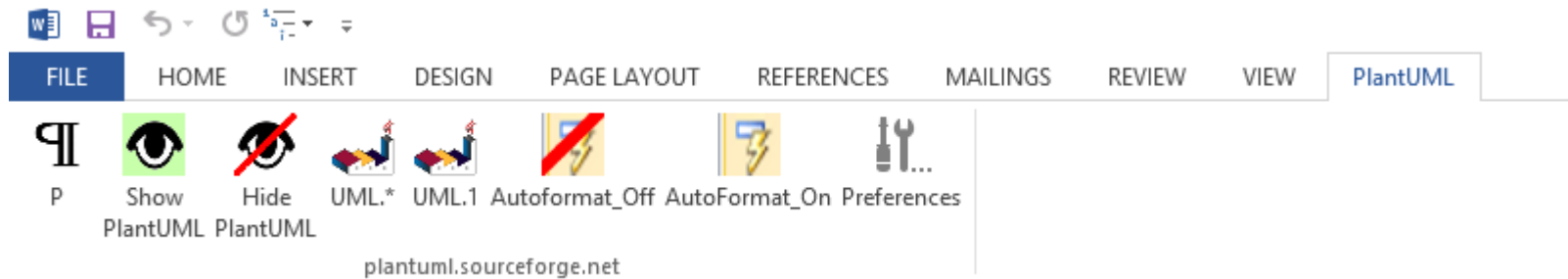
A Sequence Diagram

```
sequenceDiagram
    actor myActor
    participant myDb as myDb
    myActor->>myDb: async request
    activate myDb
    myDb-->>myActor: response
    deactivate myDb
```

Parameters

Word Macros

- Download Word template ,PlantUML_Template_v30.dotm‘
- ! current template not suitable for 64bit version of Word !
- Copy *.dotm file to %appdata%\Microsoft\Word\STARTUP
- Note: run 'echo %appdata%' to check value of ‘appdata’
- restart Word -> menu item ‘PlantUML’ should appear



Word Macros

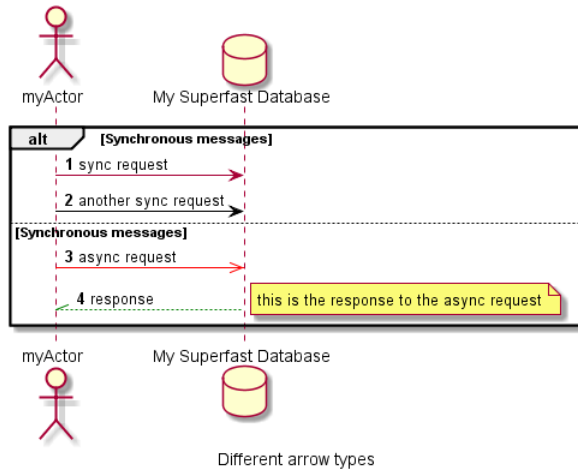
- PlantUML script within Word document; can be hidden for final doc



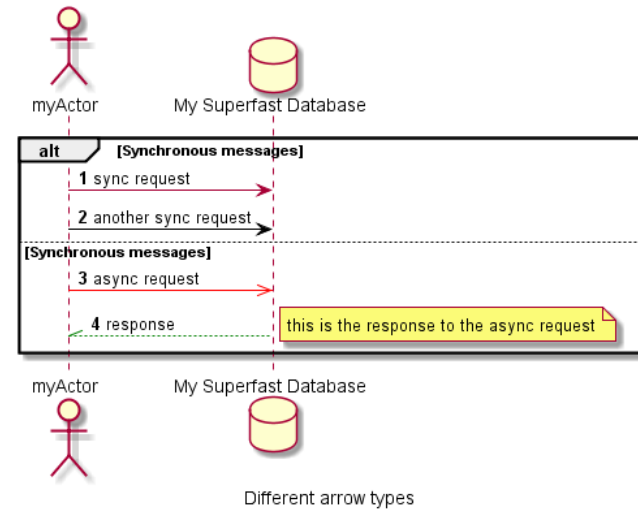
```
@startuml
title A Sequence Diagram

actor myActor
database "My Superfast Database" as myDb
note right of myDb: in the script we use the abbreviation
autonumber
alt Synchronous messages
myActor -> myDb: sync request
else Another synchronous messages
myActor -[#black]> myDb: another sync request
else Asynchronous messages
myActor -[#red]>> myDb: async request
myActor //[#green]-- myDb: response
note right: this is the response to the async request
end
caption Different arrow types
@enduml
```

A Sequence Diagram



A Sequence Diagram



Resources and Links

- PlantUML: <http://plantuml.com/>
- Word Template:
https://sourceforge.net/projects/plantuml/files/PlantUML_Template_v30.dotm/download
- Graphviz (for dot.exe): <http://www.graphviz.org/>
- Doxygen: www.doxygen.org/
- Website for editing and testing scripts: <http://www.planttext.com>

The End

