st_launcher
Tonel-based Smalltalk shell scripts

```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
Hello world smalltalk script
"
Class {
	#name : 'StLauncher_Hello',
	#superclass : 'AbstractProgram',
	#category : 'st_launcher-Scripts'
}

{ #category : 'main' }
StLauncher_Hello >> main [

	^'hello world'
]
```

Dale Henrichs
GemTalk Systems
ESUG 2019

Motivation

- **No more bash scripts**
- https://github.com/GsDevKit/GsDevKit_home
  - 50+ bash scripts
  - reimplement using Smalltalk scripts
- `topaz solo` introduced in GemStone 3.5.0
  - No more excuses not to get started

GemStone Solo sessions

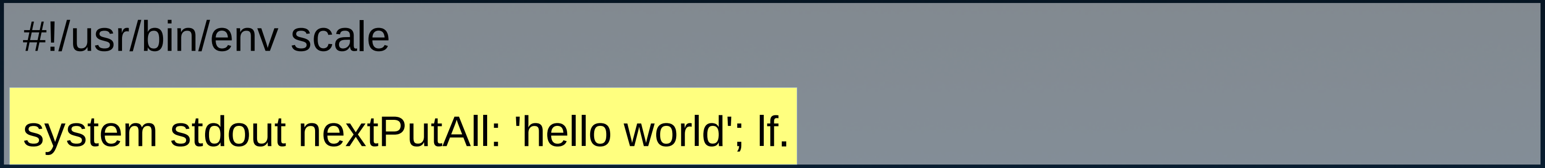
- #!/usr/bin/env topaz
  - `topaz` in $PATH
  - $GEMSTONE env var defined
  - `topaz` runs against extent file associated with $GEMSTONE

```
#!/usr/bin/env  topaz
set  u  DataCurator  p  swordfish
set sologin on
login
run
    GsFile stdout nextPutAll: ‘hello world’; lf
%
```

Pharo scale

https://github.com/guillep/Scale

- #!/usr/bin/env scale
  - `scale` in $PATH
  - Scripts run using the image associated with `scale` executable



```
#!/usr/bin/env scale

system stdout nextPutAll: 'hello world'; lf.
```

st_launcher scripts

- #!/usr/local/bin/smalltalk/gemstone/st_launcher

```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
Hello world smalltalk script
"
Class {
	#name : 'StLauncher_Hello',
	#superclass : 'AbstractProgram',
	#category : 'st_launcher-Scripts'
}
{ #category : 'main' }
StLauncher_Hello >> main [
	self stdout nextPutAll: 'hello world'; lf
]
```

“Image Names” for flexibility

- Scripts run in Pharo or Gemstone images defined by a unique “image name”
- Dialect, vm, and image information is registered by “image name”

```
hello.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
```

st_launcher
image name registry
~/.config/st_launcher/st_launcher.ston

```
StLauncherStructureProperties{
	#defaultImageName : 'gemstone',
	#homePath : '$ST_LAUNCHER_HOME',
	#imageMap : {
		'gemstone'		:	'$ST_LAUNCHER_HOME/images/gemstone',
		'pharo'			:	'$GS_HOME/shared/repos/st_launcher/home/images/pharo',
		'pharo-dev'		:	'$ST_LAUNCHER_HOME/images/pharo',
		'pharo-dev-ui'		:	'$ST_LAUNCHER_HOME/images/pharo',
		'pharo-ui'		:	'$ST_LAUNCHER_HOME/images/pharo',
		'rowan_dev_350'	:	'$GS_HOME/server/stones/rowan_dev_350',
		'st_launcher_350'	:	'$GS_HOME/server/stones/st_launcher_350'
	},
	#name : 'st_launcher_home',
	#productMap : {
		'gemstone' : {
			'3.5.0' : '$ST_LAUNCHER_HOME/platforms/gemstone/products/GemStone64Bit3.5.0',
		},
		'pharo' : {
			'70-64' : '$ST_LAUNCHER_HOME/platforms/pharo/70-64'
		} } }
```

st_launcher
image name registry
~/.config/st_launcher/st_launcher.ston

```
StLauncherStructureProperties {
        #defaultImageName : 'gemstone',
        #homePath : '$ST_LAUNCHER_HOME',
        #imageMap : {
                'gemstone'              :       '$ST_LAUNCHER_HOME/images/gemstone',
                'pharo'                 :       '$GS_HOME/shared/repos/st_launcher/home/images/pharo',
                'pharo-dev'             :       '$ST_LAUNCHER_HOME/images/pharo',
                'pharo-dev-ui'          :       '$ST_LAUNCHER_HOME/images/pharo',
                'pharo-ui'              :       '$ST_LAUNCHER_HOME/images/pharo',
                'rowan_dev_350'         :       '$GS_HOME/server/stones/rowan_dev_350',
                'st_launcher_350'       :       '$GS_HOME/server/stones/st_launcher_350'
        },
        #name : 'st_launcher_home',
        #productMap : {
                'gemstone' : {
                        '3.5.0' : '$ST_LAUNCHER_HOME/platforms/gemstone/products/GemStone64Bit3.5.0',
                },
                'pharo' : {
                        '70-64' : '$ST_LAUNCHER_HOME/platforms/pharo/70-64'
                } } }
```

st_launcher image properties (gemstone solo)

```
StLauncherGemStoneSoloImageProperties {
        #aliasProperties : {
                        'gemstone_DataCurator' : { 'topazini' : '.topazini_DataCurator' }
                },
        #binPath : 'bin',
        #comment : 'default GemStone image',
        #gemstoneVersion : '3.5.0',
        #imagePath : '$ST_LAUNCHER_HOME/images/gemstone',
        #logsPath : 'logs',
        #name : 'gemstone',
        #productPath : '$ST_LAUNCHER_HOME/platforms/gemstone/products/GemStone64Bit3.5.0',
        #snapshotsPath : 'snapshots',
        #soloSnapshot : '$ST_LAUNCHER_HOME/images/gemstone/snapshots/extent0.gemstone.dbf',
        #structureName : 'st_launcher_home'
}
```

st_launcher image properties (gemstone)

```
StLauncherGemStoneImageProperties {
	#aliasProperties : {
			'st_launcher_350' : { 'cacheName' : 'Zoro' }
		},
	#backupsPath : 'backups',
	#binPath : 'bin',
	#comment : 'development image for st_launcher',
	#extentsPath : 'extents',
	#gemstoneVersion : '3.5.0',
	#imagePath : '$GS_HOME/server/stones/st_launcher_350',
	#logsPath : 'logs',
	#name : 'st_launcher_350',
	#productPath : '$ST_LAUNCHER_HOME/platforms/gemstone/products/GemStone64Bit3.5.0’,
	#snapshotsPath : 'snapshots',
	#statsPath : 'stats',
	#structureName : 'st_launcher_home',
	#tranlogsPath : 'tranlogs'
}
```

st_launcher image properties (pharo)

```
StLauncherPharoImageProperties {
    #aliasProperties : {
            'pharo-ui' : { 'headless' : false },
            'pharo-dev' : { 'pharoImageName' : 'pharo.dev.image' },
            'pharo-dev-ui' : {
                'headless' : false,
                'pharoImageName' : 'pharo.dev.image'  }
        },
    #comment : 'default pharo image',
    #imagePath : '$ST_LAUNCHER_HOME/images/pharo',
    #pharoVersion : '70-64',
    #name : 'pharo',
    #productPath : '$ST_LAUNCHER_HOME/platforms/pharo/70-64',
    #structureName : 'st_launcher_home'
}
```

st_launcher image properties aliases

```
StLauncherPharoImageProperties {
    #aliasProperties : {
            'pharo-ui' : { 'headless' : false },
            'pharo-dev' : { 'pharoImageName' : 'pharo.dev.image' },
            'pharo-dev-ui' : {
                'headless' : false,
                'pharoImageName' : 'pharo.dev.image'  }
        },
    #comment : 'default pharo image',
    #imagePath : '$ST_LAUNCHER_HOME/images/pharo',
    #pharoVersion : '70-64',
    #name : 'pharo',
    #productPath : '$ST_LAUNCHER_HOME/platforms/pharo/70-64',
    #structureName : 'st_launcher_home'
}
```

Advantages of class-based scripts

- Structure for organizing script functionality (superior to workspace)
- Inherited behavior for
  - options and argument handling
  - Help/usage generation
- First class object in image
  - Create and maintain script using traditional browser and tools
  - Senders/implementors

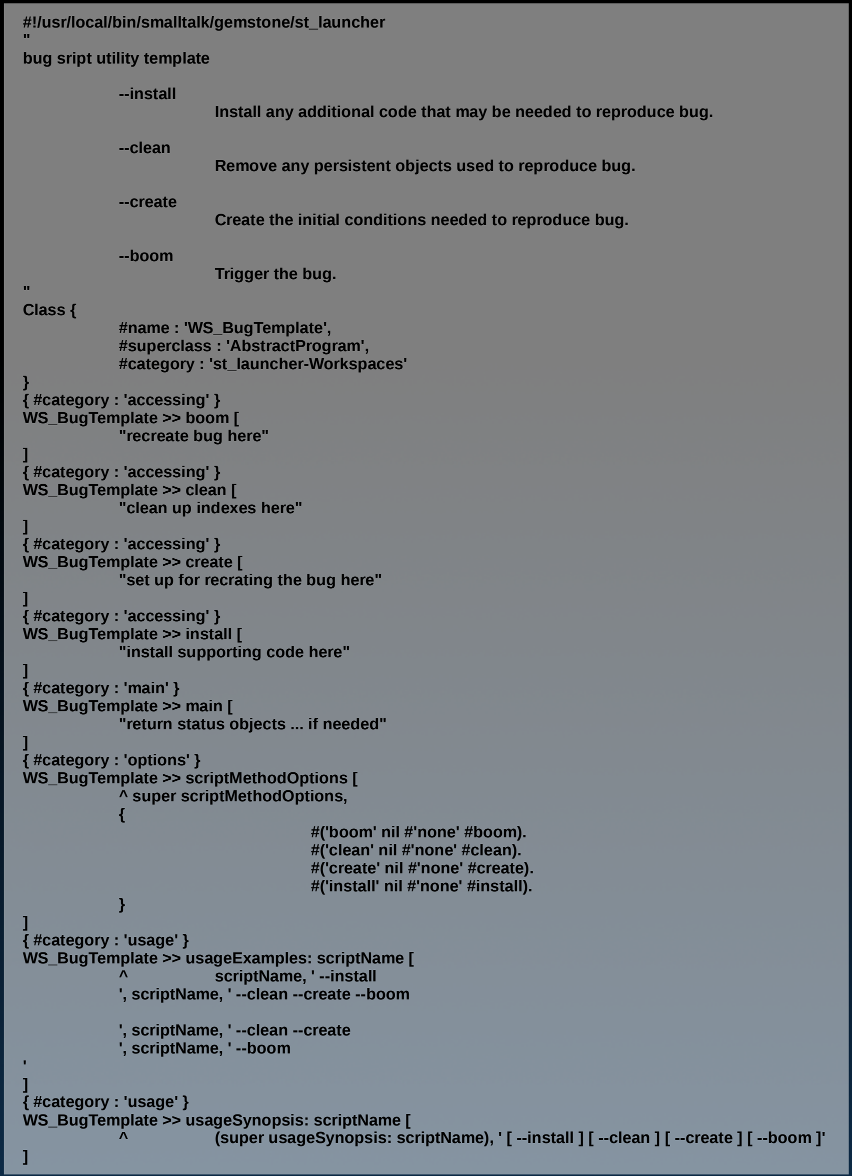
st_launcher help text

```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
Hello world smalltalk script
"
Class {
	#name : 'StLauncher_Hello',
	#superclass : 'AbstractProgram',
	#category : 'st_launcher-Scripts'
}
{ #category : 'main' }
StLauncher_Hello >> main [
	self stdout nextPutAll: 'hello world'; lf
]
```

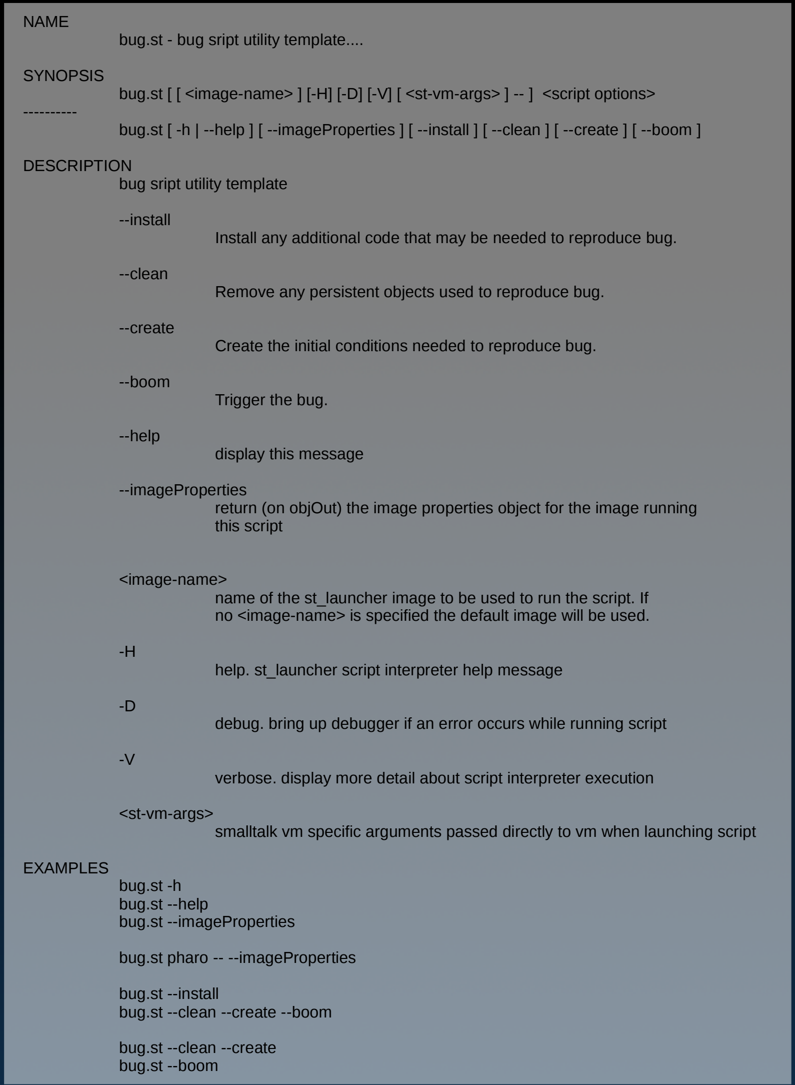
```
NAME
	hello.st - Hello world smalltalk script....

SYNOPSIS
	hello.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
----------
	hello.st [ -h | --help ] [ --imageProperties ]

DESCRIPTION
	Hello world smalltalk script

	--help
		display this message

	--imageProperties
		return (on objOut) the image properties object for the image running
		this script


	<image-name>
		name of the st_launcher image to be used to run the script. If
		no <image-name> is specified the default image will be used.

	-H
		help. st_launcher script interpreter help message

	-D
		debug. bring up debugger if an error occurs while running script

	-V
		verbose. display more detail about script interpreter execution

	<st-vm-args>
		smalltalk vm specific arguments passed directly to vm when launching script

EXAMPLES
	hello.st -h
	hello.st --help
	hello.st --imageProperties

	hello.st pharo -- --imageProperties

	hello.st
```

st_launcher

Usage
Options Declaration
Script Body



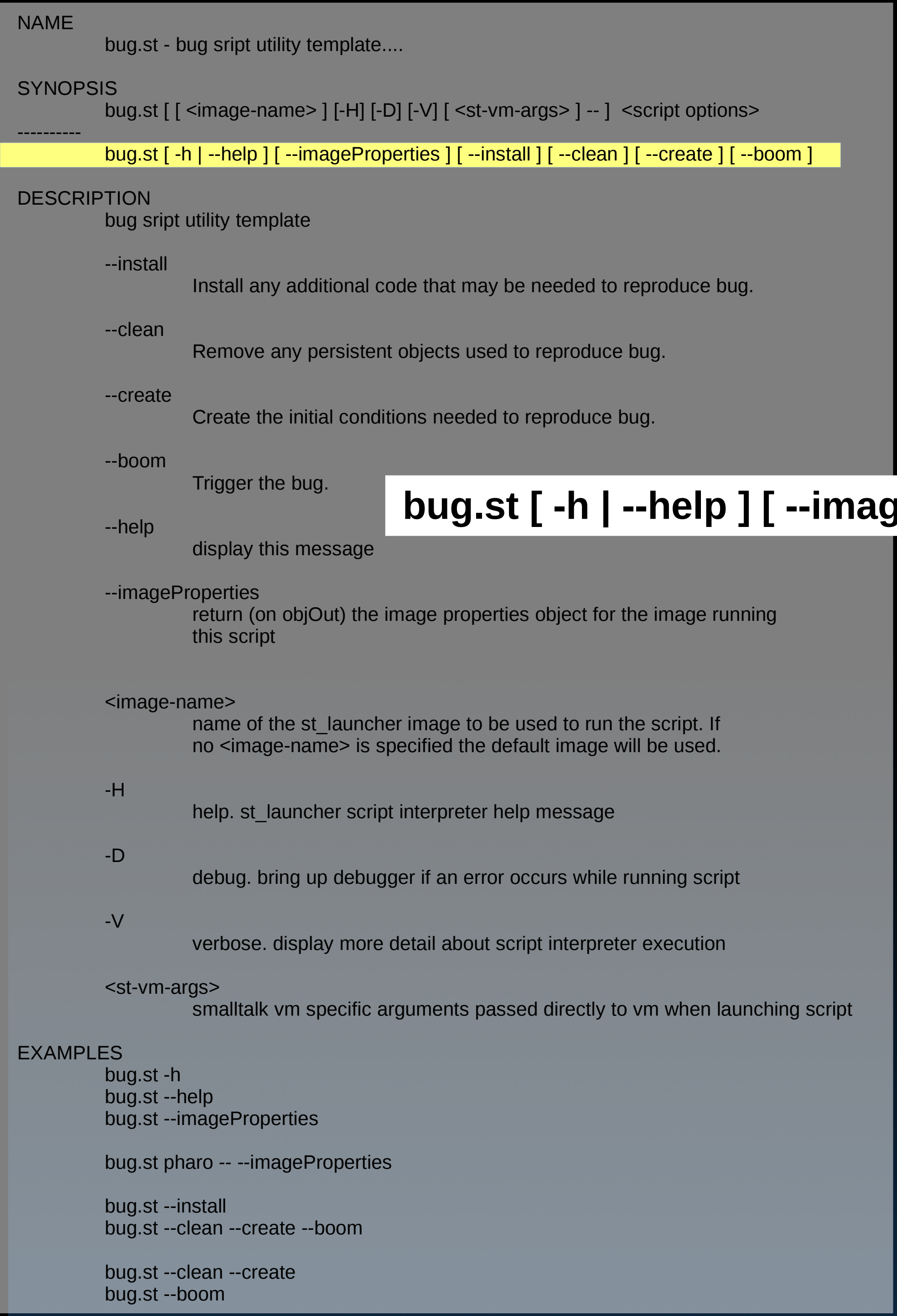
```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.
"
Class {
        #name : 'WS_BugTemplate',
        #superclass : 'AbstractProgram',
        #category : 'st_launcher-Workspaces'
}
{ #category : 'accessing' }
WS_BugTemplate >> boom [
        "recreate bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> clean [
        "clean up indexes here"
]
{ #category : 'accessing' }
WS_BugTemplate >> create [
        "set up for recrating the bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> install [
        "install supporting code here"
]
{ #category : 'main' }
WS_BugTemplate >> main [
        "return status objects ... if needed"
]
{ #category : 'options' }
WS_BugTemplate >> scriptMethodOptions [
        ^ super scriptMethodOptions,
        {
                        #('boom' nil #'none' #boom).
                        #('clean' nil #'none' #clean).
                        #('create' nil #'none' #create).
                        #('install' nil #'none' #install).
        }
]
{ #category : 'usage' }
WS_BugTemplate >> usageExamples: scriptName [
        ^       scriptName, ' --install
        ', scriptName, ' --clean --create --boom

        ', scriptName, ' --clean --create
        ', scriptName, ' --boom
'
]
{ #category : 'usage' }
WS_BugTemplate >> usageSynopsis: scriptName [
        ^       (super usageSynopsis: scriptName), ' [ --install ] [ --clean ] [ --create ] [ --boom ]'
]
```



```
NAME
        bug.st - bug sript utility template....

SYNOPSIS
        bug.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
----------
        bug.st [ -h | --help ] [ --imageProperties ] [ --install ] [ --clean ] [ --create ] [ --boom ]

DESCRIPTION
        bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.

        --help
                display this message

        --imageProperties
                return (on objOut) the image properties object for the image running
                this script


        <image-name>
                name of the st_launcher image to be used to run the script. If
                no <image-name> is specified the default image will be used.

        -H
                help. st_launcher script interpreter help message

        -D
                debug. bring up debugger if an error occurs while running script

        -V
                verbose. display more detail about script interpreter execution

        <st-vm-args>
                smalltalk vm specific arguments passed directly to vm when launching script

EXAMPLES
        bug.st -h
        bug.st --help
        bug.st --imageProperties

        bug.st pharo -- --imageProperties

        bug.st --install
        bug.st --clean --create --boom

        bug.st --clean --create
        bug.st --boom
```

st_launcher

Usage: SYNOPSIS

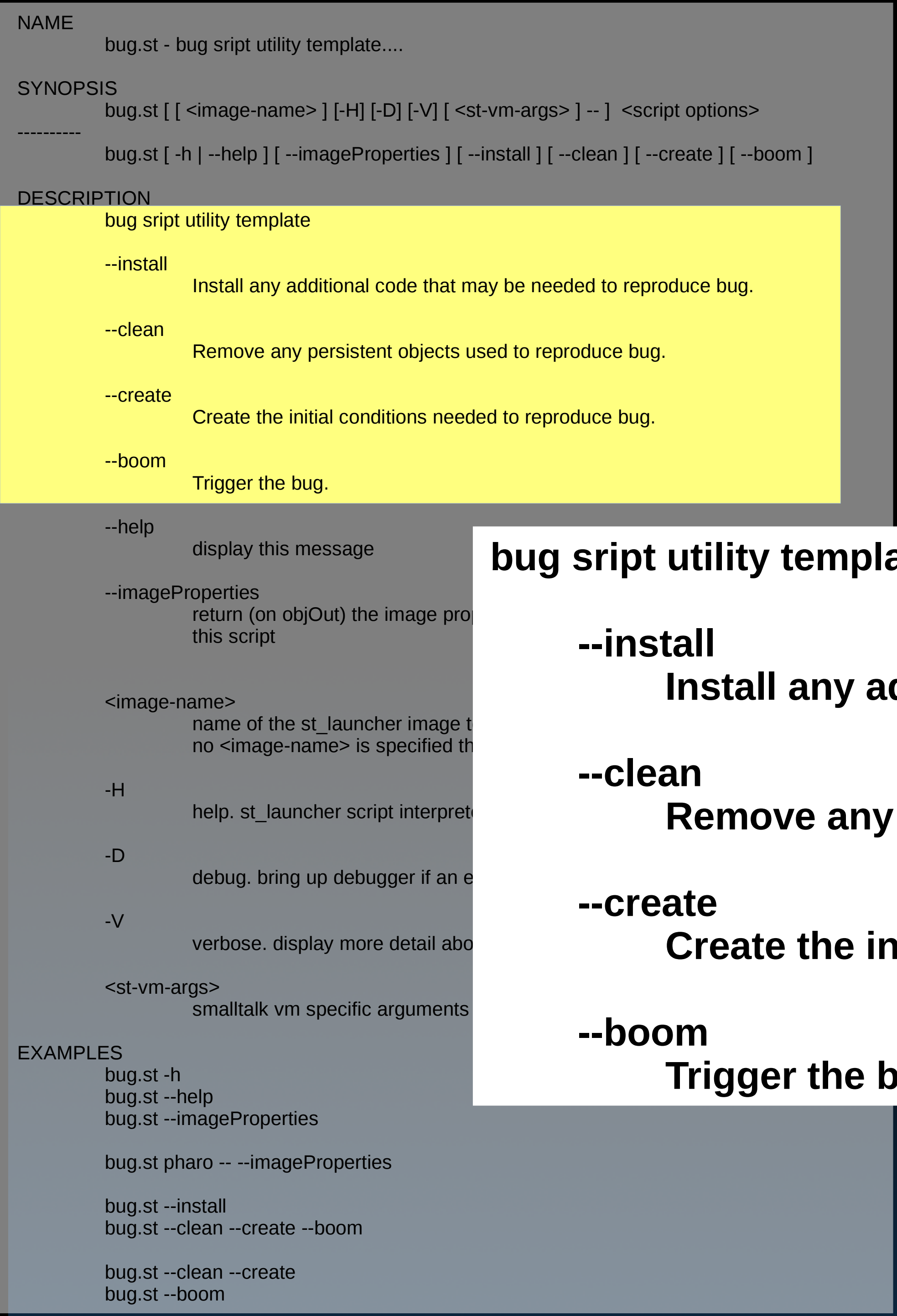


```
NAME
          bug.st - bug sript utility template....

SYNOPSIS
          bug.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
----------
          bug.st [ -h | --help ] [ --imageProperties ] [ --install ] [ --clean ] [ --create ] [ --boom ]

DESCRIPTION
          bug sript utility template

          --install
                    Install any additional code that may be needed to reproduce bug.

          --clean
                    Remove any persistent objects used to reproduce bug.

          --create
                    Create the initial conditions needed to reproduce bug.

          --boom
                    Trigger the bug.

          --help
                    display this message

          --imageProperties
                    return (on objOut) the image properties object for the image running
                    this script


          <image-name>
                    name of the st_launcher image to be used to run the script. If
                    no <image-name> is specified the default image will be used.

          -H
                    help. st_launcher script interpreter help message

          -D
                    debug. bring up debugger if an error occurs while running script

          -V
                    verbose. display more detail about script interpreter execution

          <st-vm-args>
                    smalltalk vm specific arguments passed directly to vm when launching script

EXAMPLES
          bug.st -h
          bug.st --help
          bug.st --imageProperties

          bug.st pharo -- --imageProperties

          bug.st --install
          bug.st --clean --create --boom

          bug.st --clean --create
          bug.st --boom
```

bug.st [-h | --help] [--imageProperties] [--install] [--clean] [--create] [--boom]

st_launcher

Usage: DESCRIPTION



```
NAME
        bug.st - bug sript utility template....

SYNOPSIS
        bug.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
----------
        bug.st [ -h | --help ] [ --imageProperties ] [ --install ] [ --clean ] [ --create ] [ --boom ]

DESCRIPTION
        bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.

        --help
                display this message

        --imageProperties
                return (on objOut) the image pro
                this script

        <image-name>
                name of the st_launcher image t
                no <image-name> is specified th

        -H
                help. st_launcher script interpret

        -D
                debug. bring up debugger if an e

        -V
                verbose. display more detail abo

        <st-vm-args>
                smalltalk vm specific arguments

EXAMPLES
        bug.st -h
        bug.st --help
        bug.st --imageProperties

        bug.st pharo -- --imageProperties

        bug.st --install
        bug.st --clean --create --boom

        bug.st --clean --create
        bug.st --boom
```

bug sript utility template

--install
Install any additional code that may be needed to reproduce bug.

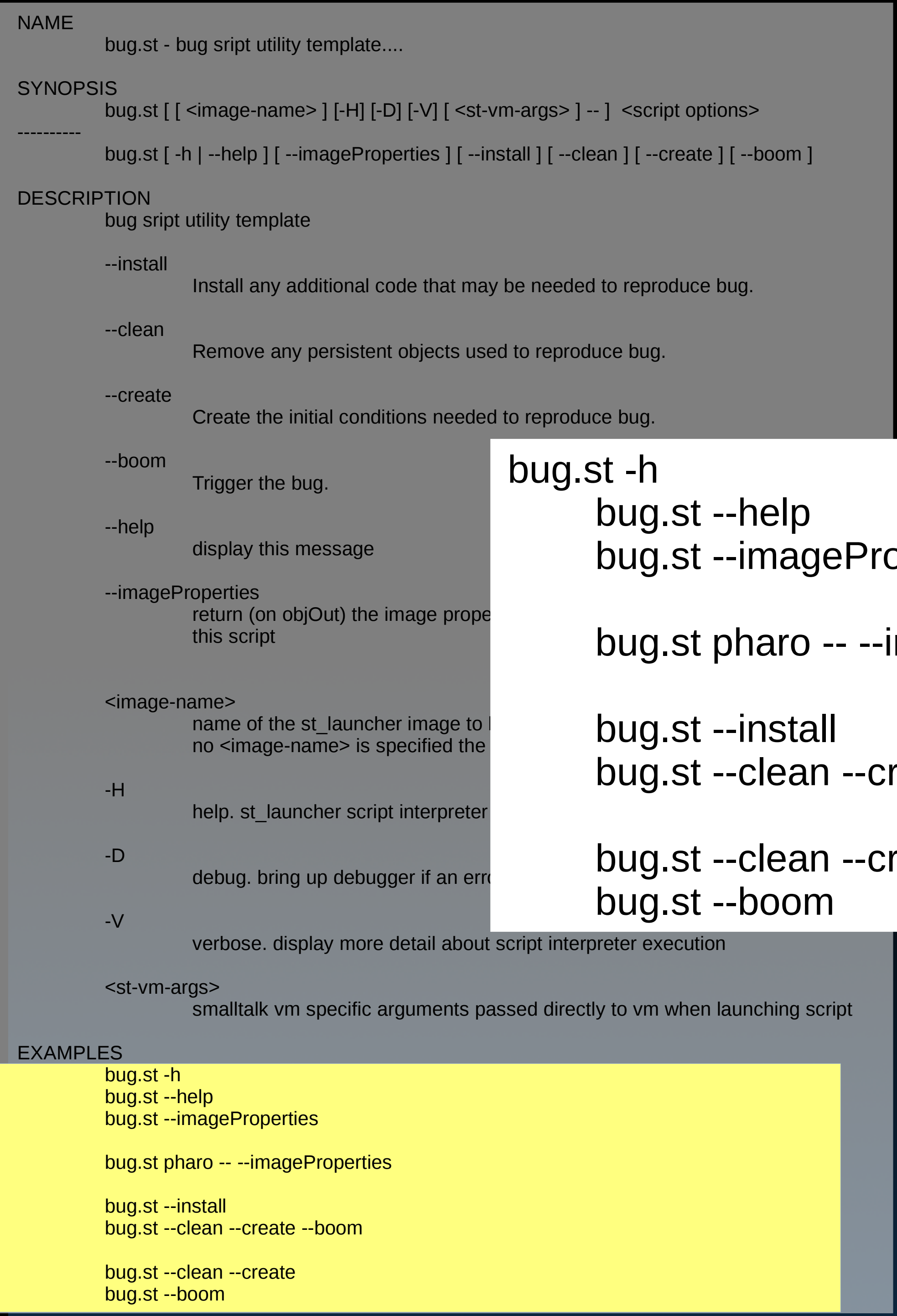
--clean
Remove any persistent objects used to reproduce bug.

--create
Create the initial conditions needed to reproduce bug.

--boom
Trigger the bug.

st_launcher

Usage: EXAMPLES



```
NAME
        bug.st - bug sript utility template....

SYNOPSIS
        bug.st [ [ <image-name> ] [-H] [-D] [-V] [ <st-vm-args> ] -- ]  <script options>
----------
        bug.st [ -h | --help ] [ --imageProperties ] [ --install ] [ --clean ] [ --create ] [ --boom ]

DESCRIPTION
        bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.

        --help
                display this message

        --imageProperties
                return (on objOut) the image prope
                this script


        <image-name>
                name of the st_launcher image to l
                no <image-name> is specified the

        -H
                help. st_launcher script interpreter

        -D
                debug. bring up debugger if an err

        -V
                verbose. display more detail about script interpreter execution

        <st-vm-args>
                smalltalk vm specific arguments passed directly to vm when launching script

EXAMPLES
        bug.st -h
        bug.st --help
        bug.st --imageProperties

        bug.st pharo -- --imageProperties

        bug.st --install
        bug.st --clean --create --boom

        bug.st --clean --create
        bug.st --boom
```

```
bug.st -h
        bug.st --help
        bug.st --imageProperties

        bug.st pharo -- --imageProperties

        bug.st --install
        bug.st --clean --create --boom

        bug.st --clean --create
        bug.st --boom
```

Usage

```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
bug sript utility template

        --install
                Install any additional code tha

        --clean
                Remove any persistent objects

        --create
                Create the initial conditions ne

        --boom
                Trigger the bug.
"
Class {
        #name : 'WS_BugTemplate',
        #superclass : 'AbstractProgram',
        #category : 'st_launcher-Workspaces'
}
{ #category : 'accessing' }
WS_BugTemplate >> boom [
        "recreate bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> clean [
        "clean up indexes here"
]
{ #category : 'accessing' }
WS_BugTemplate >> create [
        "set up for recrating the bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> install [
        "install supporting code here"
]
{ #category : 'main' }
WS_BugTemplate >> main [
        "return status objects ... if needed"
]
{ #category : 'options' }
WS_BugTemplate >> scriptMethodOptions [
        ^ super scriptMethodOptions,
        {
                        #('boom' nil #'none' #boom).
                        #('clean' nil #'none' #clean).
                        #('create' nil #'none' #create).
                        #('install' nil #'none' #install).
        }
]
{ #category : 'usage' }
WS_BugTemplate >> usageExamples: scriptName [
        ^       scriptName, ' --install
        ', scriptName, ' --clean --create --boom

        ', scriptName, '
        ', scriptName, '
'
]
{ #category : 'usage' }
WS_BugTemplate >> usageSynopsis: scriptName [
        ^       (super usageSynopsis: scriptName), ' [ --install ] [ --clean ] [ --create ] [ --boom ]'
]
```

```
“
bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.
“
```

```
WS_BugTemplate >> usageExamples: scriptName [
        ^       scriptName, ' --install
        ', scriptName, ' --clean --create --boom

        ', scriptName, ' --clean --create
        ', scriptName, ' --boom
'
]
```

```
WS_BugTemplate >> usageSynopsis: scriptName [
        ^       (super usageSynopsis: scriptName), ' [ --install ] [ --clean ] [ --create ] [ --boom ]'
]
```

st_launcher

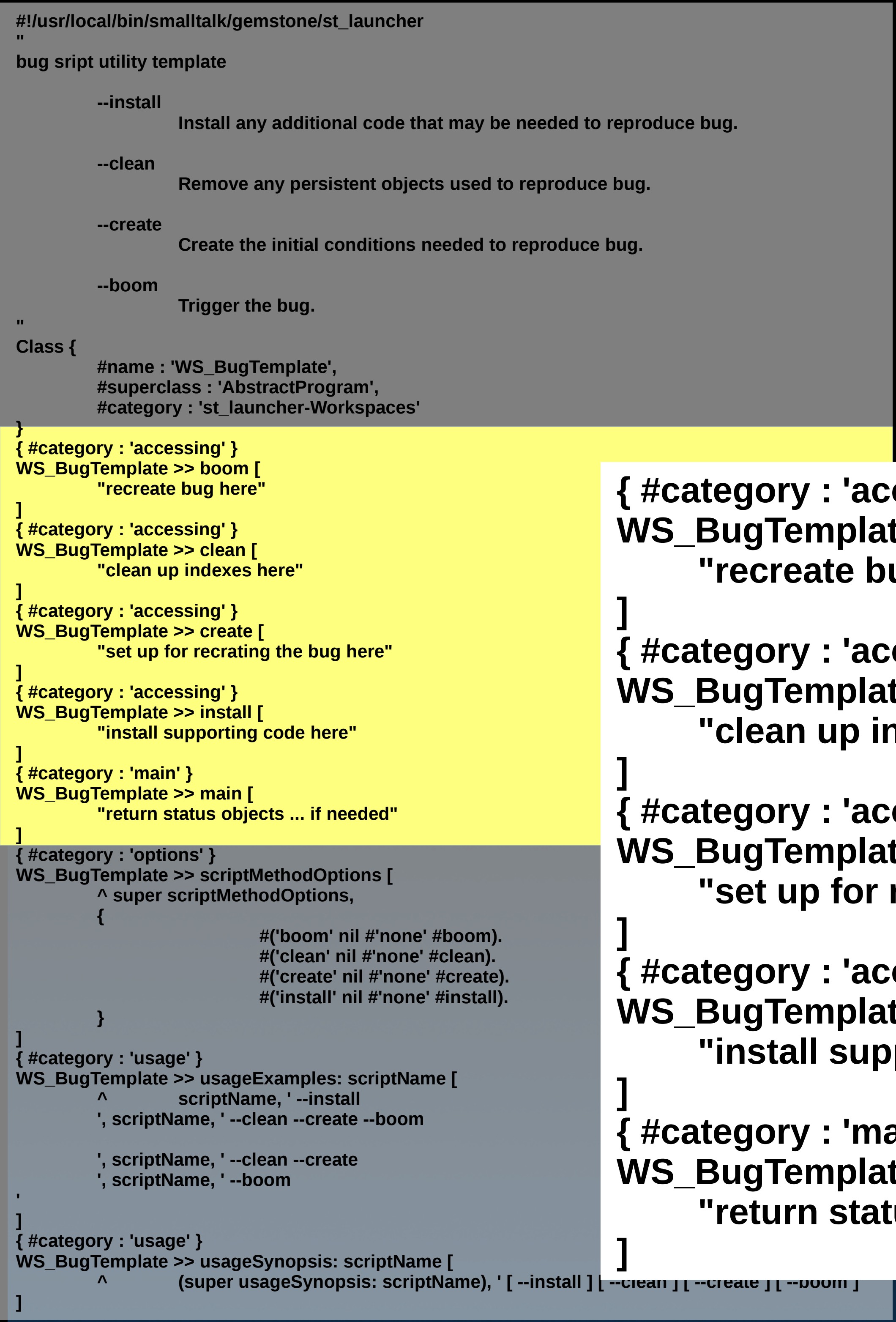
Options declaration

```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
bug sript utility template

		--install
			Install any additional code that may be needed to reproduce bug.

		--clean
			Remove any persistent objects used to reproduce bug.

		--create
			Create the initial conditions needed to reproduce bug.

		--boom
			Trigger the bug.
"
Class {
		#name : 'WS_BugTemplate',
		#superclass : 'AbstractProgram',
		#category : 'st_launcher-Workspaces'
}
{ #category : 'accessing' }
WS_BugTemplate >> boom [
		"recreate bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> clean [
		"clean up indexes here"
]
{ #category : 'accessing' }
WS_BugTemplate >> create [
		"set up for recrating the bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> install [
		"install supporting code here"
]
{ #category : 'main' }
WS_BugTemplate >> main [
		"return status objects ... if needed"
]
{ #category : 'options' }
WS_BugTemplate >> scriptMethodOptions [
		^ super scriptMethodOptions,
		{
					#('boom' nil #'none' #boom).
					#('clean' nil #'none' #clean).
					#('create' nil #'none' #create).
					#('install' nil #'none' #install).
		}
]
{ #category : 'usage' }
WS_BugTemplate >> usageExamples: scriptName [
		^		scriptName, ' --install
		', scriptName, ' --clean --create --boom

		', scriptName, ' --clean --create
		', scriptName, ' --boom
'
]
{ #category : 'usage' }
WS_BugTemplate >> usageSynopsis: scriptName [
		^		(super usageSynopsis: scriptName), ' [ --install ] [ --clean ] [ --create ] [ --boom ]'
]
```

```
WS_ >> scriptMethodOptions [
		^ super scriptMethodOptions,
		{
					#('boom' nil #'none' #boom).
					#('clean' nil #'none' #clean).
					#('create' nil #'none' #create).
					#('install' nil #'none' #install).
		}
]
```

st_launcher

Script body

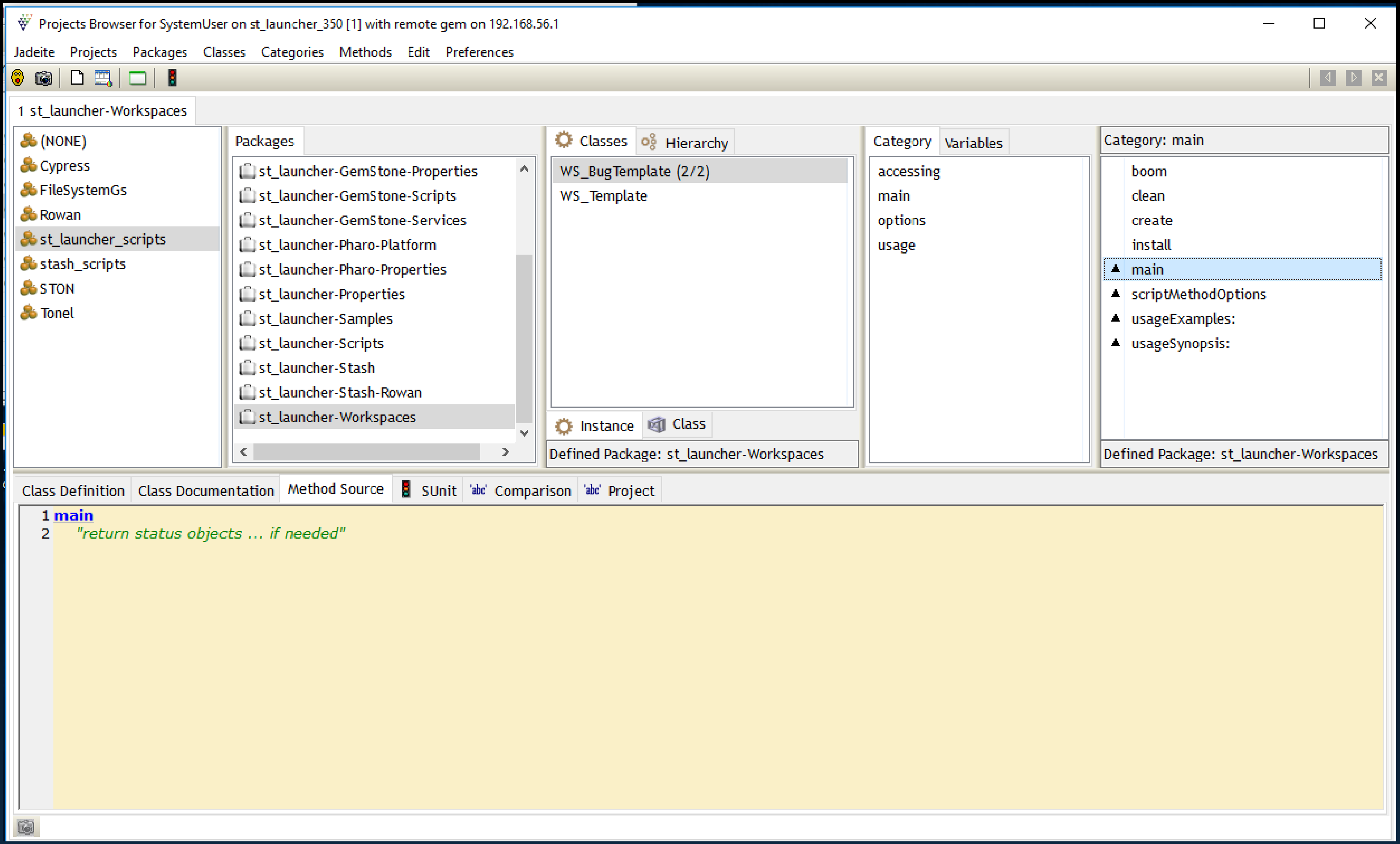


```
#!/usr/local/bin/smalltalk/gemstone/st_launcher
"
bug sript utility template

        --install
                Install any additional code that may be needed to reproduce bug.

        --clean
                Remove any persistent objects used to reproduce bug.

        --create
                Create the initial conditions needed to reproduce bug.

        --boom
                Trigger the bug.
"
Class {
        #name : 'WS_BugTemplate',
        #superclass : 'AbstractProgram',
        #category : 'st_launcher-Workspaces'
}
{ #category : 'accessing' }
WS_BugTemplate >> boom [
        "recreate bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> clean [
        "clean up indexes here"
]
{ #category : 'accessing' }
WS_BugTemplate >> create [
        "set up for recrating the bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> install [
        "install supporting code here"
]
{ #category : 'main' }
WS_BugTemplate >> main [
        "return status objects ... if needed"
]
{ #category : 'options' }
WS_BugTemplate >> scriptMethodOptions [
        ^ super scriptMethodOptions,
        {
                        #('boom' nil #'none' #boom).
                        #('clean' nil #'none' #clean).
                        #('create' nil #'none' #create).
                        #('install' nil #'none' #install).
        }
]
{ #category : 'usage' }
WS_BugTemplate >> usageExamples: scriptName [
        ^       scriptName, ' --install
        ', scriptName, ' --clean --create --boom

        ', scriptName, ' --clean --create
        ', scriptName, ' --boom
'
]
{ #category : 'usage' }
WS_BugTemplate >> usageSynopsis: scriptName [
        ^       (super usageSynopsis: scriptName), ' [ --install ] [ --clean ] [ --create ] [ --boom ]
]
```

```
{ #category : 'accessing' }
WS_BugTemplate >> boom [
        "recreate bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> clean [
        "clean up indexes here"
]
{ #category : 'accessing' }
WS_BugTemplate >> create [
        "set up for recreating the bug here"
]
{ #category : 'accessing' }
WS_BugTemplate >> install [
        "install supporting code here"
]
{ #category : 'main' }
WS_BugTemplate >> main [
        "return status objects ... if needed"
]
```

st_launcher

Jadeite (Rowan) browser



Projects Browser for SystemUser on st_launcher_350 [1] with remote gem on 192.168.56.1

Defined Package: st_launcher-Workspaces

Category: main

Defined Package: st_launcher-Workspaces

```
main
  "return status objects ... if needed"
```

Additional st_launcher Script Features

- objIn/objOut
- Live debugger for script errors

Demos

Shell interpreter

- Bash program (right now)
- Runs the default image to translate the imageName into a command line for the correct vm/image that are needed to run the script
- Executes the generated command line

Tonel class file loading for execution

- Parse Tonel class file
- Create class definition
  - MCClassDefinition in Pharo
  - RowanClassDefinition in GemStone
- Load class definition

Tonel class file loading as package

- Not implemented in Pharo
- GemStone
  - Rowan Tonel parser and loader preserve the shebang line in a class property on read/load
  - On write the shebang line is added back to the file

Project Status
https://github.com/dalehenrich/st_launcher

- Currently in alpha
- If you are interested, I invite collaborators to contribute ideas and code