Main UI Docking

Antidoc v3.0.0, Ron Dexter

Table of Contents

1. Project description . . . 1
2. Libraries . . . 2
    2.1. Docking.lvlib . . . 2
3. Classes . . . 3
    3.1. Classes overview . . . 3
    3.2. Adjust Splitter Msg.lvclass . . . 3
4. Actors (AF) . . . 5
    4.1. Preamble . . . 5
    4.2. Actors overview . . . 5
    4.3. BottomDock.lvclass . . . 5
    4.4. Docking.lvclass . . . 6
    4.5. PrimeDock.lvclass . . . 7
    4.6. RightDock.lvclass . . . 8
5. Legal Information . . . 10
    5.1. Document creation . . . 10
    5.2. Product used in the project . . . 12

Chapter 1. Project description

No description found (add content in project description)

Chapter 2. Libraries

This section describes the libraries contained in the project.

2.1. Docking.lvlib

Responsibility: The Docking library provides the basic defintions for panel position, scale and adjustment methods.

this library inherits from the managed actors to provide a means to manage subordinate classes if necessary.

Version: 1.0.0.0

Table 1. Nested libraries

Name	Type
Adjust Splitter Msg.lvclass	LVClass
BottomDock.lvclass	LVClass
Docking.lvclass	LVClass
PrimeDock.lvclass	LVClass
RightDock.lvclass	LVClass

2.1.1. Functions

This library has no functions set to non private scope.

2.1.2. Library Constant VIs

NOTE	No Constant VIs Found

Chapter 3. Classes

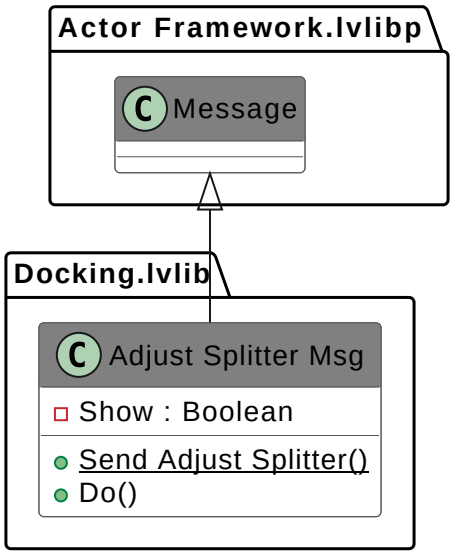
This section describes the classes contained in the project.

3.1. Classes overview

This project contains 1 classe and 0 interface.

Table 2. Classes list

Classes	Interfaces
Adjust Splitter Msg.lvclass	

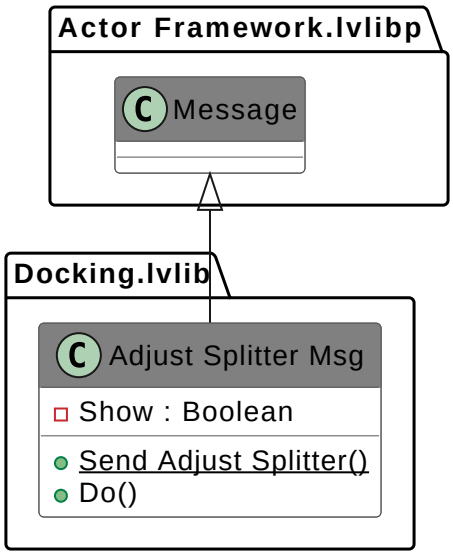


3.2. Adjust Splitter Msg.lvclass

Responsibility: This is an override message that the docked stations implement when they are commanded to show or hide. This provides the pop-up and pop-down capabilities of the software.

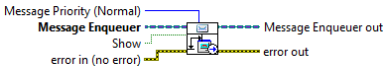
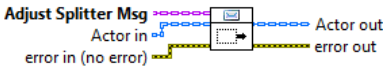
Version: 1.0.0.0

3.2.1. Diagram



3.2.2. Methods

Table 3. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Send Adjust Splitter		This VI sends the message to an actor.			
Do		This VI delivers the message to the actor by calling the appropriate method(s) on the actor.			

Scope: → Protected | → Community

Reentrancy: → Preallocated reentrancy | → Shared reentrancy

Inlining: → Inlined

3.2.3. Class Constant VIs

NOTE	No Constant VIs Found

Chapter 4. Actors (AF)

This section describes AF framework usage in the project

4.1. Preamble

Add anything that could be interseting to describe AF concepts and help the reader to understand the AF section

4.2. Actors overview

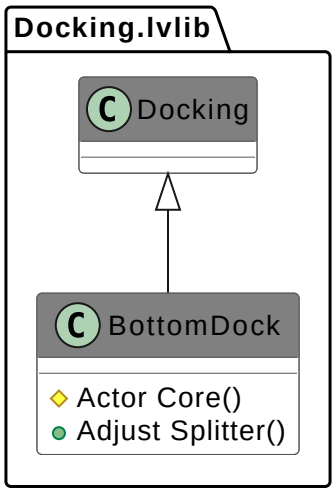
4.3. BottomDock.lvclass

Responsibility: This is the class for ancesor classes that must dock to the Bottom subpanel of the main application.

It provides the scale and color information for the actor core, and manages the splitter adjustment necessary to show or hide the panel.

Version: 1.0.0.0

4.3.1. Diagram



4.3.2. Methods

Table 4. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Actor Core		Base Class for the bottom subpanel. This provides the basic implementation for show/hide operations. ancestor classes should override this, but maintain proportions.			

Name	Connector pane	Description	S.	R.	I.
Adjust Splitter		This class is designed so that ancestor classes can rescale the UI to fit the actor core of the ancestor UI.			

Scope: → Protected | → Community

Reentrancy: → Preallocated reentrancy | → Shared reentrancy

Inlining: → Inlined

4.3.3. Class Constant VIs

NOTE	No Constant VIs Found

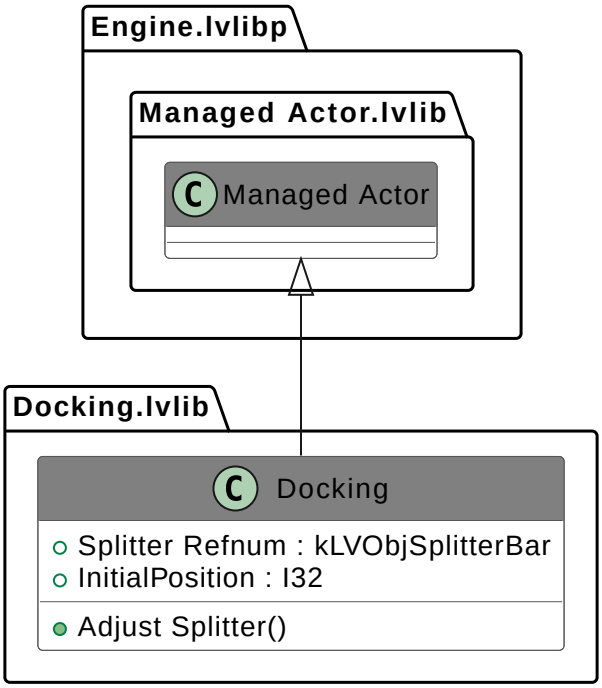
4.4. Docking.lvclass

Responsibility: This is the ancestor class for all docked classes in the main application. it provides the override and actor messages for adjusting splitters.

By default, this class inherits from the Actor Manager so that subordinate classes can administer their own panels if so desired. (typically this behavior is reserved for the prime panel)

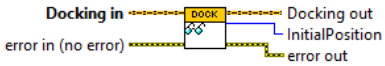
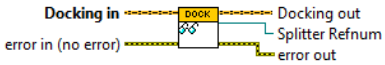
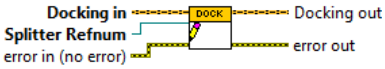
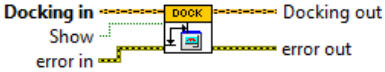
Version: 1.0.0.0

4.4.1. Diagram



4.4.2. Methods

Table 5. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Read InitialPosition		Returns the default position of the splitter no matter how much it has been adjusted			
Read Splitter Refnum		Returns a refnum to the splitter that is adjusted to show/hide the docked actor.			
Write Splitter Refnum		Writes the splitter refnum to private data for access/update. When writing the splitter refnum, this routine also reads the initial position of the splitter and stores the initial position to private data.			
Adjust Splitter		This class is designed so that ancestor classes can rescale the UI to fit the actor core of the ancestor UI.			

Scope: → Protected | → Community

Reentrancy: → Preallocated reentrancy | → Shared reentrancy

Inlining: → Inlined

4.4.3. Class Constant VIs

NOTE	No Constant VIs Found

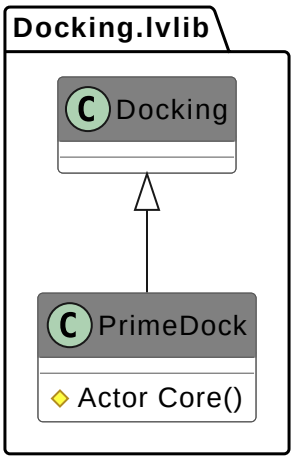
4.5. PrimeDock.lvclass

Responsibility: This is the class for ancesor classes that must dock to the Prime subpanel of the main application.

It provides the scale and color information for the actor core - Prime panel does not reposition splitters because it is a fixed size.

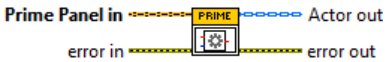
Version: 1.0.0.0

4.5.1. Diagram



4.5.2. Methods

Table 6. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Actor Core		Base Class for the right subpanel. This provides the basic implementation for show/hide operations. ancestor classes should override this, but maintain proportions.			

Scope: → Protected | → Community

Reentrancy: → Preallocated reentrancy | → Shared reentrancy

Inlining: → Inlined

4.5.3. Class Constant VIs

NOTE	No Constant VIs Found

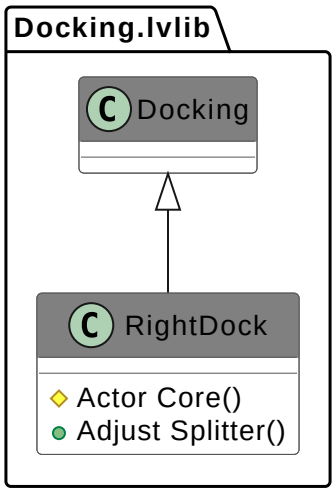
4.6. RightDock.lvclass

Responsibility: This is the class for ancesor classes that must dock to the right subpanel of the main application.

It provides the scale and color information for the actor core, and manages the splitter adjustment necessary to show or hide the panel.

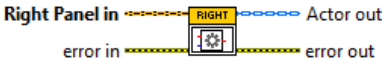
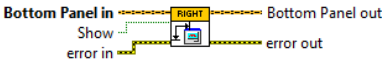
Version: 1.0.0.0

4.6.1. Diagram



4.6.2. Methods

Table 7. Functions (non private scope only)

Name	Connector pane	Description	S.	R.	I.
Actor Core		Base Class for the right subpanel. This provides the basic implementation for show/hide operations. ancestor classes should override this, but maintain proportions.			
Adjust Splitter		This class is designed so that ancestor classes can rescale the UI to fit the actor core of the ancestor UI.			

Scope: → Protected | → Community

Reentrancy: → Preallocated reentrancy | → Shared reentrancy

Inlining: → Inlined

4.6.3. Class Constant VIs

NOTE	No Constant VIs Found

Chapter 5. Legal Information

5.1. Document creation

This document has been generated using the following tools.

5.1.1. Antidoc

Project website: Antidoc

Maintainer website: Wovalab

5.1.2. Asciidoc for LabVIEW™

Project website: Asciidoc toolkit

Maintainer website: Wovalab

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5.1.3. classy Diagram Viewer

Project website: classy Diagram Viewer

BSD 3-Clause License

Copyright © 2021, Tatiana Boyé All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5.2. Product used in the project

Antidoc hasn’t been able to detect third party products in the project. This is the author’s responsibility to list any of the missing product used.