

*Changed text!*

My Project

Generated by Doxygen 1.8.11

A Bit More Sample Text!

Added text.



# Contents



Sample Text!

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Hierarchical Index</b>   | <b>1</b> |
| 1.1      | Class Hierarchy . . . . .   | 1        |
| <b>2</b> | <b>Class Index</b>  | <b>3</b> |
| 2.1      | Class List . . . . .  | 3        |
| <b>3</b> | <b>Class Documentation</b>  | <b>5</b> |
| 3.1      | air.AbstractDocumentCommandWrapper.AbstractDocumentCommand Class Reference . . . . .    | 5        |
| 3.1.1    | Member Function Documentation . . . . .   | 7        |
| 3.1.1.1  | execute() . . . . .   | 7        |
| 3.1.1.2  | getName() . . . . .   | 7        |
| 3.2      | air.AbstractDocumentCommandWrapper Class Reference . . . . .                            | 8        |
| 3.2.1    | Detailed Description . . . . .  | 10       |
| 3.3      | air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand Class Reference . . . . .    | 10       |
| 3.3.1    | Member Function Documentation . . . . .   | 11       |
| 3.3.1.1  | execute() . . . . .   | 11       |
| 3.3.1.2  | getName() . . . . .   | 11       |
| 3.4      | air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand Class Reference . . . . . | 12       |
| 3.4.1    | Member Function Documentation . . . . .   | 13       |
| 3.4.1.1  | execute() . . . . .   | 13       |
| 3.4.1.2  | getName() . . . . .   | 13       |
| 3.5      | air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand Class Reference . . . . .   | 14       |

|          |  |    |
|----------|--|----|
| 3.5.1    | Member Function Documentation  | 15 |
| 3.5.1.1  | execute()  | 15 |
| 3.5.1.2  | getName()  | 15 |
| 3.6      | air.AIRMain Class Reference  | 16 |
| 3.6.1    | Detailed Description   | 16 |
| 3.6.2    | Member Function Documentation  | 16 |
| 3.6.2.1  | test(String[] args)  | 16 |
| 3.7      | air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand Class Reference       | 17 |
| 3.7.1    | Member Function Documentation  | 18 |
| 3.7.1.1  | execute()  | 18 |
| 3.7.1.2  | getName()  | 18 |
| 3.8      | air.DocumentCommandWrapper Class Reference                                       | 19 |
| 3.8.1    | Detailed Description   | 21 |
| 3.8.2    | Member Function Documentation  | 21 |
| 3.8.2.1  | loadDocumentAtPath(String aPath)   | 21 |
| 3.8.2.2  | save(String aPath)   | 21 |
| 3.9      | air.AbstractDocumentCommandWrapper.makeCommand Interface Reference               | 22 |
| 3.10     | air.DocumentCommandWrapper.MoveAnnotationDocumentCommand Class Reference         | 23 |
| 3.10.1   | Member Function Documentation  | 24 |
| 3.10.1.1 | execute()  | 24 |
| 3.10.1.2 | getName()  | 24 |
| 3.11     | air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand Class Reference | 25 |
| 3.11.1   | Member Function Documentation  | 26 |
| 3.11.1.1 | execute()  | 26 |
| 3.11.1.2 | getName()  | 27 |
| 3.12     | air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand Class Reference      | 27 |
| 3.12.1   | Member Function Documentation  | 28 |
| 3.12.1.1 | execute()  | 28 |

|              |  |           |
|--------------|--|-----------|
| 3.12.1.2     | <a href="#">getName()</a>  | 29        |
| 3.13         | <a href="#">air.DocumentCommandWrapper.SaveDocumentCommand Class Reference</a>                   | 29        |
| 3.13.1       | <a href="#">Member Function Documentation</a>  | 30        |
| 3.13.1.1     | <a href="#">execute()</a>  | 30        |
| 3.13.1.2     | <a href="#">getName()</a>  | 31        |
| 3.14         | <a href="#">air.DocumentCommandWrapper.SaveTextDocumentCommand Class Reference</a>               | 31        |
| 3.14.1       | <a href="#">Member Function Documentation</a>  | 32        |
| 3.14.1.1     | <a href="#">execute()</a>  | 32        |
| 3.14.1.2     | <a href="#">getName()</a>  | 33        |
| 3.15         | <a href="#">air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand Class Reference</a> | 33        |
| 3.15.1       | <a href="#">Member Function Documentation</a>  | 34        |
| 3.15.1.1     | <a href="#">execute()</a>  | 34        |
| 3.15.1.2     | <a href="#">getName()</a>  | 35        |
| <b>Index</b> |  | <b>37</b> |

Some edited text.



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

|  |    |
|--|----|
| air.AbstractDocumentCommandWrapper.AbstractDocumentCommand       | 5  |
| air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand | 25 |
| air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand | 33 |
| air.DocumentCommandWrapper.SaveDocumentCommand                   | 29 |
| air.DocumentCommandWrapper.SaveTextDocumentCommand               | 31 |
| AbstractDocumentCommand  |    |
| air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand       | 10 |
| air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand    | 12 |
| air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand      | 14 |
| air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand       | 17 |
| air.DocumentCommandWrapper.MoveAnnotationDocumentCommand         | 23 |
| air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand      | 27 |
| air.AbstractDocumentCommandWrapper                               | 8  |
| air.DocumentCommandWrapper                                       | 19 |
| air.AIRMain  | 16 |
| air.AbstractDocumentCommandWrapper.makeCommand                   | 22 |





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

|  |    |
|--|----|
| <a href="#">air.AbstractDocumentCommandWrapper.AbstractDocumentCommand</a>       | 5  |
| <a href="#">air.AbstractDocumentCommandWrapper</a>                               | 8  |
| <a href="#">air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand</a>       | 10 |
| <a href="#">air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand</a>    | 12 |
| <a href="#">air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand</a>      | 14 |
| <a href="#">air.AIRMain</a>  | 16 |
| <a href="#">air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand</a>       | 17 |
| <a href="#">air.DocumentCommandWrapper</a>                                       | 19 |
| <a href="#">air.AbstractDocumentCommandWrapper.makeCommand</a>                   | 22 |
| <a href="#">air.DocumentCommandWrapper.MoveAnnotationDocumentCommand</a>         | 23 |
| <a href="#">air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand</a> | 25 |
| <a href="#">air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand</a>      | 27 |
| <a href="#">air.DocumentCommandWrapper.SaveDocumentCommand</a>                   | 29 |
| <a href="#">air.DocumentCommandWrapper.SaveTextDocumentCommand</a>               | 31 |
| <a href="#">air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand</a> | 33 |

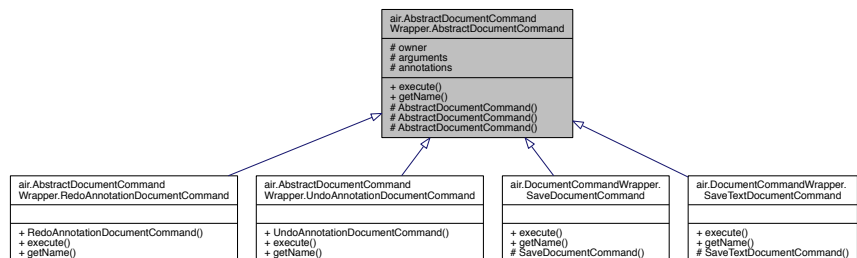


## Chapter 3

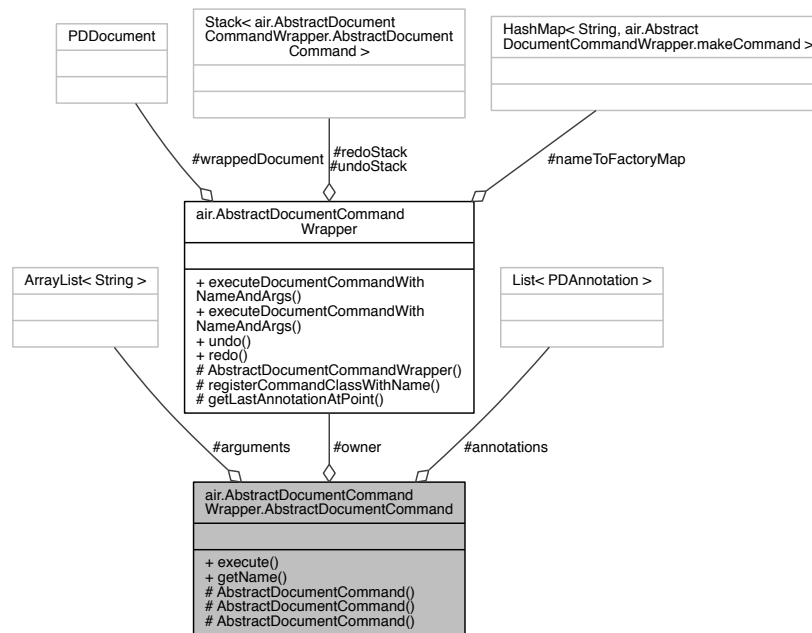
# Class Documentation

### 3.1 air.AbstractDocumentCommandWrapper.AbstractDocumentCommand Class Reference

Inheritance diagram for air.AbstractDocumentCommandWrapper.AbstractDocumentCommand:



Collaboration diagram for `air.AbstractDocumentCommandWrapper.AbstractDocumentCommand`:



## Public Member Functions

- abstract `AbstractDocumentCommand execute ()`
- abstract `String getName ()`

## Protected Member Functions

- **AbstractDocumentCommand** (`AbstractDocumentCommandWrapper` anOwner, `ArrayList< String >` args)
- **AbstractDocumentCommand** (`AbstractDocumentCommandWrapper` anOwner, `List< PDAnnotation >` someAnnotations)
- **AbstractDocumentCommand** (`AbstractDocumentCommandWrapper` anOwner, `List< PDAnnotation >` someAnnotations, `ArrayList< String >` args)

## Protected Attributes

- final `AbstractDocumentCommandWrapper` **owner**
- final `ArrayList< String >` **arguments**
- `List< PDAnnotation >` **annotations**

### 3.1.1 Member Function Documentation

3.1.1.1 **abstract AbstractDocumentCommand** air.AbstractDocumentCommandWrapper.AbstractDocumentCommand.execute ( ) [abstract]

#### Returns

If [execute\(\)](#) succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.

3.1.1.2 **abstract String** air.AbstractDocumentCommandWrapper.AbstractDocumentCommand.getName ( ) [abstract]

#### Returns

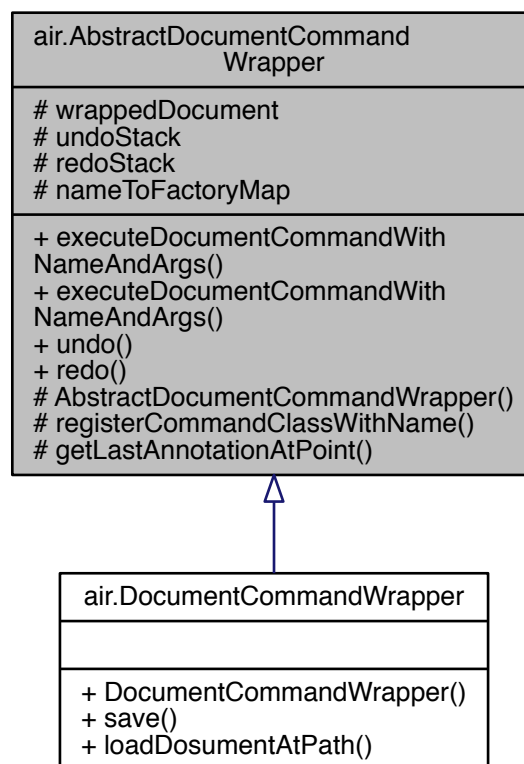
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

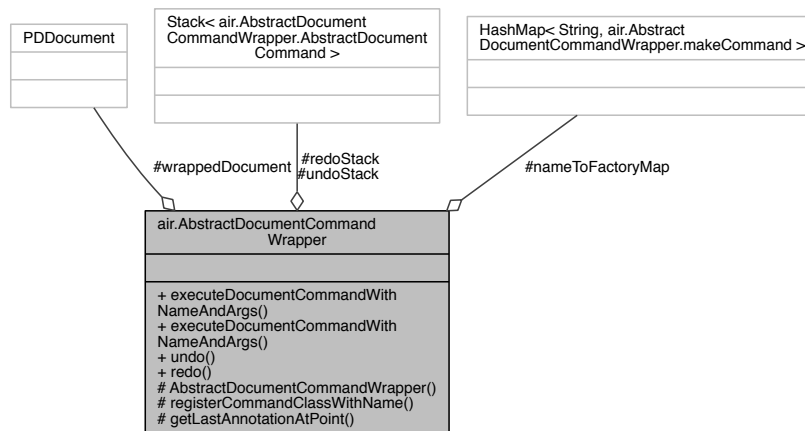
- src/air/AbstractDocumentCommandWrapper.java

## 3.2 air.AbstractDocumentCommandWrapper Class Reference

Inheritance diagram for air.AbstractDocumentCommandWrapper:



Collaboration diagram for air.AbstractDocumentCommandWrapper:



## Classes

- class [AbstractDocumentCommand](#)
- interface [makeCommand](#)
- class [RedoAnnotationDocumentCommand](#)
- class [UndoAnnotationDocumentCommand](#)

## Public Member Functions

- boolean **executeDocumentCommandWithNameAndArgs** (String aName, ArrayList< String > args)
- boolean **executeDocumentCommandWithNameAndArgs** (String aName, String[] args)
- boolean **undo** ()
- boolean **redo** ()

## Protected Member Functions

- **AbstractDocumentCommandWrapper** (PDDocument aDocument)
- void **registerCommandClassWithName** ([makeCommand](#) command, String aName)
- PDAnnotation **getLastAnnotationAtPoint** (List< PDAnnotation > annotations, float x, float y)

## Protected Attributes

- final PDDocument **wrappedDocument**
- final Stack< [AbstractDocumentCommand](#) > **undoStack**
- final Stack< [AbstractDocumentCommand](#) > **redoStack**
- final HashMap< String, [makeCommand](#) > **nameToFactoryMap**

### 3.2.1 Detailed Description

**Author**

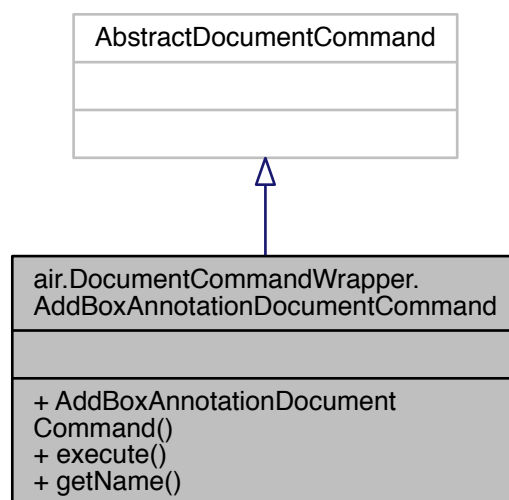
erik

The documentation for this class was generated from the following file:

- `src/air/AbstractDocumentCommandWrapper.java`

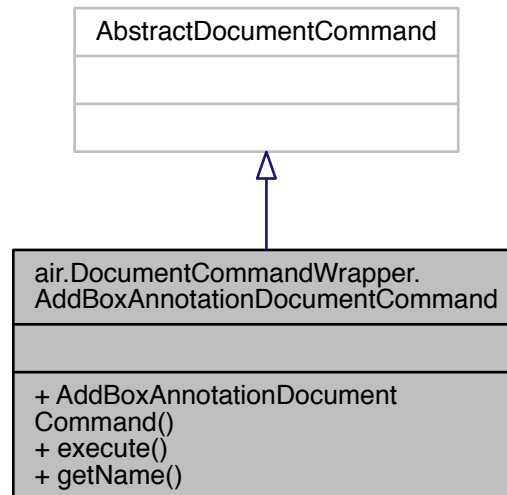
### 3.3 `air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand` Class Reference

Inheritance diagram for `air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand`:





Collaboration diagram for air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand:



## Public Member Functions

- **AddBoxAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, `ArrayList< String >` args)
- `AbstractDocumentCommand` [execute](#) ()
- `String` [getName](#) ()

### 3.3.1 Member Function Documentation

#### 3.3.1.1 `AbstractDocumentCommand` `air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand.execute ( )`

##### Returns

If [execute\(\)](#) succeeds, a `Command` that is the reciprocal of the receiver is returned. Otherwise, `null` is returned.

#### 3.3.1.2 `String` `air.DocumentCommandWrapper.AddBoxAnnotationDocumentCommand.getName ( )`

##### Returns

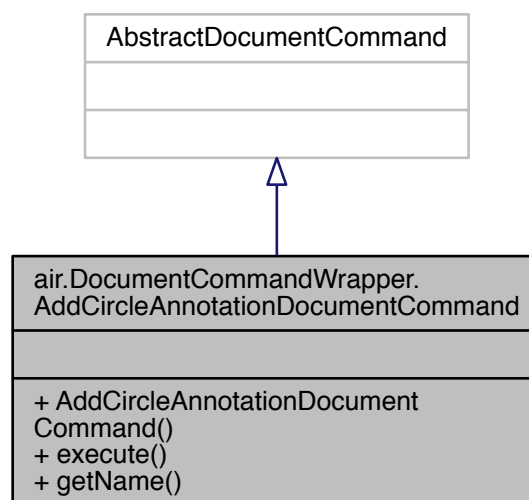
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

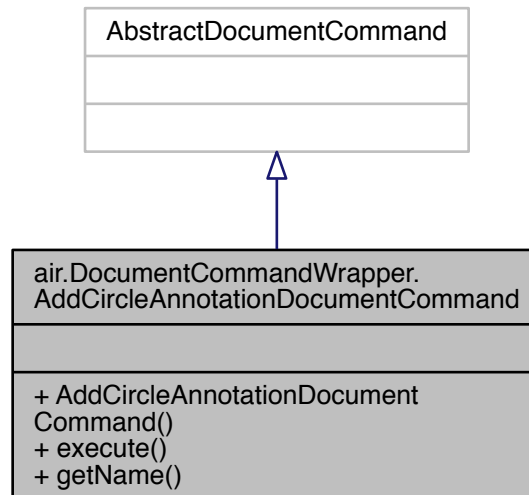
- `src/air/DocumentCommandWrapper.java`

### 3.4 air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand:



Collaboration diagram for air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand:



## Public Member Functions

- **AddCircleAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, `ArrayList< String > args`)
- `AbstractDocumentCommand` [execute](#) ()
- `String` [getName](#) ()

### 3.4.1 Member Function Documentation

#### 3.4.1.1 `AbstractDocumentCommand` `air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand.execute ( )`

##### Returns

If [execute\(\)](#) succeeds, a `Command` that is the reciprocal of the receiver is returned. Otherwise, `null` is returned.

#### 3.4.1.2 `String` `air.DocumentCommandWrapper.AddCircleAnnotationDocumentCommand.getName ( )`

##### Returns

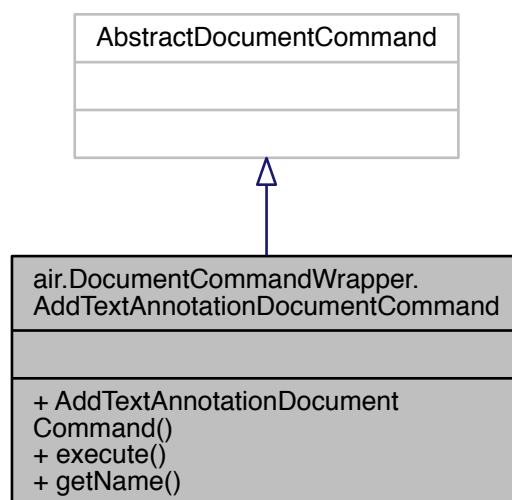
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

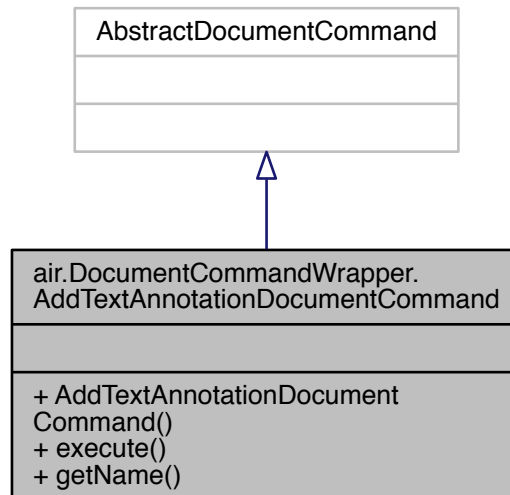
- `src/air/DocumentCommandWrapper.java`

### 3.5 air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand:



Collaboration diagram for air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand:



## Public Member Functions

- **AddTextAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, `ArrayList< String >` args)
- `AbstractDocumentCommand` [execute](#) ()
- `String` [getName](#) ()

### 3.5.1 Member Function Documentation

#### 3.5.1.1 `AbstractDocumentCommand` `air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand.execute` ( )

##### Returns

If [execute\(\)](#) succeeds, a `Command` that is the reciprocal of the receiver is returned. Otherwise, `null` is returned.

#### 3.5.1.2 `String` `air.DocumentCommandWrapper.AddTextAnnotationDocumentCommand.getName` ( )

##### Returns

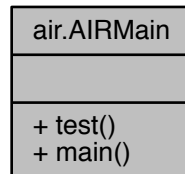
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

- `src/air/DocumentCommandWrapper.java`

## 3.6 air.AIRMain Class Reference

Collaboration diagram for air.AIRMain:



### Static Public Member Functions

- static void `test` (String[] args) throws IOException
- static void `main` (String[] args) throws IOException

### 3.6.1 Detailed Description

Author

erik

### 3.6.2 Member Function Documentation

3.6.2.1 static void `air.AIRMain.test` ( String[] args ) throws IOException [static]

Parameters

|             |                            |
|-------------|----------------------------|
| <i>args</i> | the command line arguments |
|-------------|----------------------------|

Exceptions

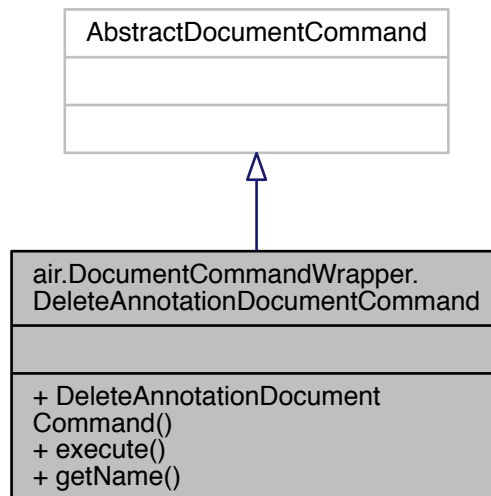
|                            |  |
|----------------------------|--|
| <i>java.io.IOException</i> |  |
|----------------------------|--|

The documentation for this class was generated from the following file:

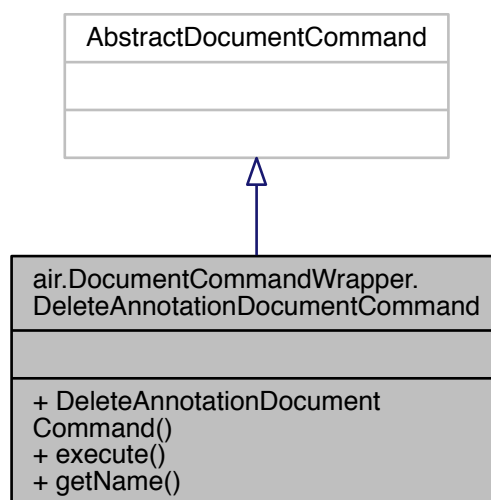
- `src/air/AIRMain.java`

### 3.7 air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand:



Collaboration diagram for air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand:



## Public Member Functions

- **DeleteAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, ArrayList< String > args)
- AbstractDocumentCommand [execute](#) ()
- String [getName](#) ()

### 3.7.1 Member Function Documentation

#### 3.7.1.1 AbstractDocumentCommand air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand.execute ( )

##### Returns

If [execute\(\)](#) succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.

#### 3.7.1.2 String air.DocumentCommandWrapper.DeleteAnnotationDocumentCommand.getName ( )

##### Returns

The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

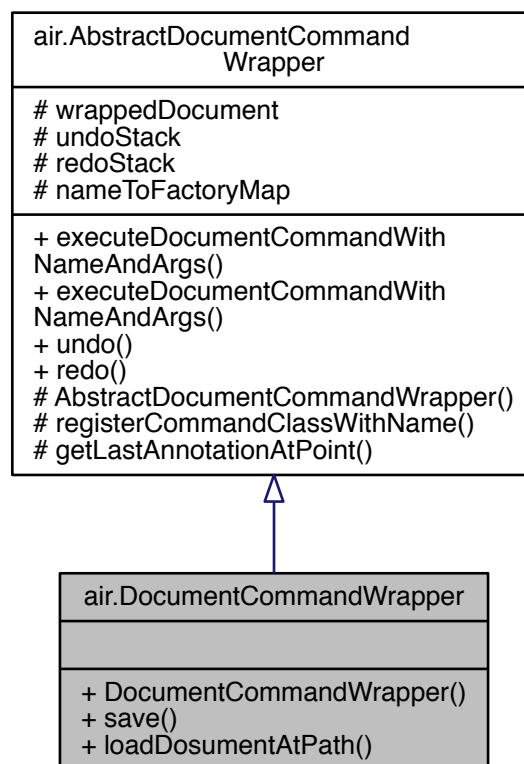
The documentation for this class was generated from the following file:

- src/air/DocumentCommandWrapper.java

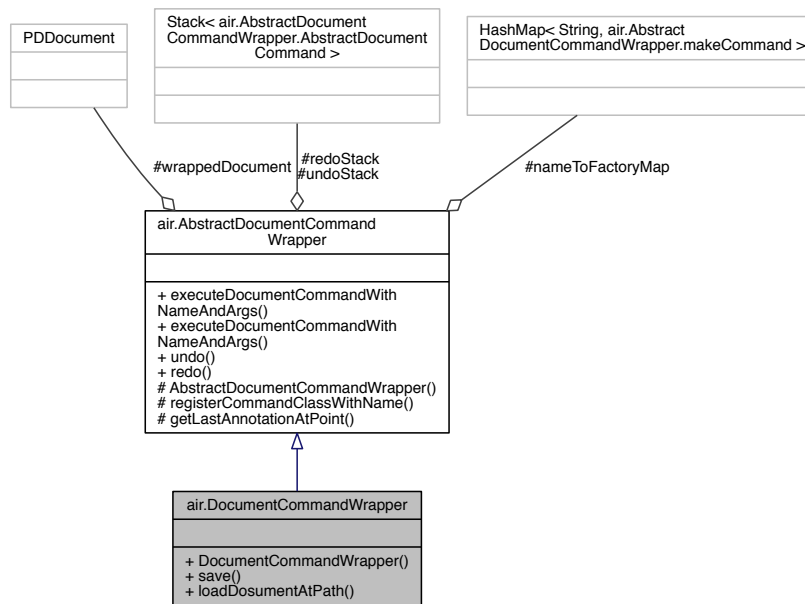


## 3.8 air.DocumentCommandWrapper Class Reference

Inheritance diagram for air.DocumentCommandWrapper:



Collaboration diagram for `air.DocumentCommandWrapper`:



## Classes

- class [AddBoxAnnotationDocumentCommand](#)
- class [AddCircleAnnotationDocumentCommand](#)
- class [AddTextAnnotationDocumentCommand](#)
- class [DeleteAnnotationDocumentCommand](#)
- class [MoveAnnotationDocumentCommand](#)
- class [ReplaceAnnotationDocumentCommand](#)
- class [SaveDocumentCommand](#)
- class [SaveTextDocumentCommand](#)

## Public Member Functions

- **DocumentCommandWrapper** (`PDDocument aDocument`, `String aPath`)
- void [save](#) (`String aPath`) throws `IOException`  
*Save the PDF information encapsulated by the receiver to a file at the specified path.*

## Static Public Member Functions

- static [DocumentCommandWrapper loadDocumentAtPath](#) (`String aPath`) throws `IOException`  
*Reads information from the valid PDF file if any at `aPath` and returns a [DocumentCommandWrapper](#) instance ready to add or delete annotations to the loaded information.*

## Additional Inherited Members

### 3.8.1 Detailed Description

#### Author

erik

### 3.8.2 Member Function Documentation

#### 3.8.2.1 static DocumentCommandWrapper air.DocumentCommandWrapper.loadDocumentAtPath ( String aPath ) throws IOException [static]

Reads information from the valid PDF file if any at aPath and returns a [DocumentCommandWrapper](#) instance ready to add or delete annotations to the loaded information.

Trace: REQ-041000, REQ-041060

#### Parameters

|              |   |
|--------------|---|
| <i>aPath</i> | A valid file system path to a valid PDF document. |
|--------------|---|

#### Returns

An instance of [DocumentCommandWrapper](#) configured to apply Commands to the document at aPath.

#### Exceptions

|                    |  |
|--------------------|--|
| <i>IOException</i> |  |
|--------------------|--|

#### 3.8.2.2 void air.DocumentCommandWrapper.save ( String aPath ) throws IOException

Save the PDF information encapsulated by the receiver to a file at the specified path.

The saved information contains the same annotation and non-annotation information as the file (if any) from which information was loaded to initialize the receiver with the following exceptions: - Annotations added via the receiver since information was loaded are saved in the specified file. - Annotations deleted via the receiver since information was loaded are not saved in the specified file. - Annotations whose attributes have been changed via the receiver since information was loaded are saved with changed attributes in the specified file. Trace: REQ-041000, REQ-041050

#### Parameters

|              |  |
|--------------|--|
| <i>aPath</i> | A valid complete file system path including the name of the document to save. The path must be writable in a file system with sufficient available storage capacity to store the information encapsulated by the receiver. |
|--------------|--|

### Exceptions

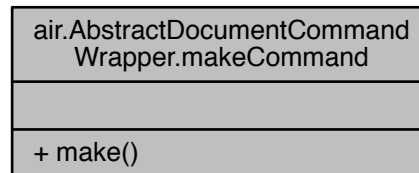
|                                  |  |
|----------------------------------|--|
| <code>java.io.IOException</code> |  |
|----------------------------------|--|

The documentation for this class was generated from the following file:

- `src/air/DocumentCommandWrapper.java`

## 3.9 `air.AbstractDocumentCommandWrapper.makeCommand` Interface Reference

Collaboration diagram for `air.AbstractDocumentCommandWrapper.makeCommand`:



### Public Member Functions

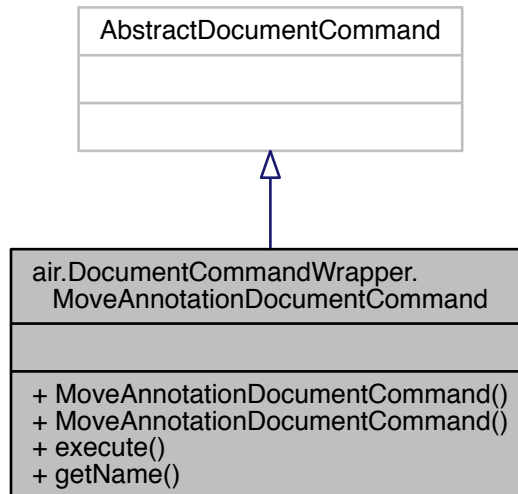
- `AbstractDocumentCommand` **make** (`AbstractDocumentCommandWrapper` anOwner, `ArrayList< String >` args)

The documentation for this interface was generated from the following file:

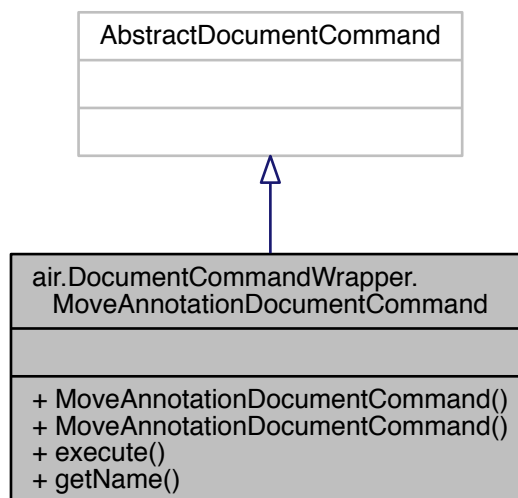
- `src/air/AbstractDocumentCommandWrapper.java`

### 3.10 air.DocumentCommandWrapper.MoveAnnotationDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.MoveAnnotationDocumentCommand:



Collaboration diagram for air.DocumentCommandWrapper.MoveAnnotationDocumentCommand:



## Public Member Functions

- **MoveAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, ArrayList< String > args)
- **MoveAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, List< PDAnnotation > annotations, ArrayList< String > args)
- AbstractDocumentCommand [execute](#) ()
- String [getName](#) ()

### 3.10.1 Member Function Documentation

#### 3.10.1.1 AbstractDocumentCommand air.DocumentCommandWrapper.MoveAnnotationDocumentCommand.execute ( )

##### Returns

If [execute\(\)](#) succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.

#### 3.10.1.2 String air.DocumentCommandWrapper.MoveAnnotationDocumentCommand.getName ( )

##### Returns

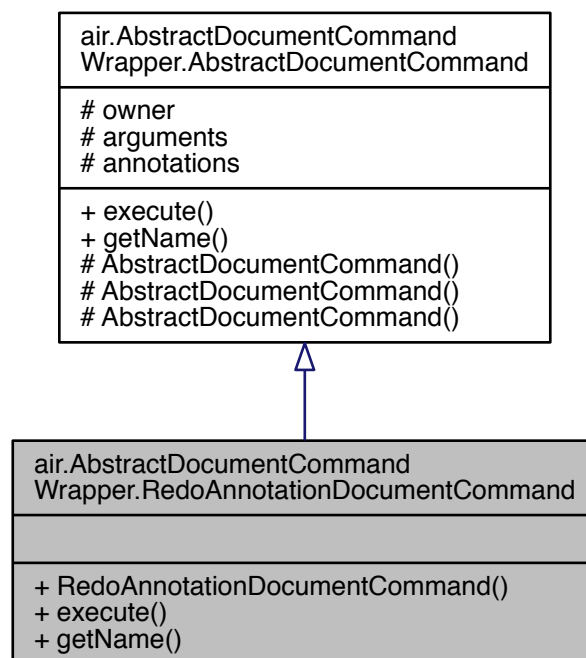
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

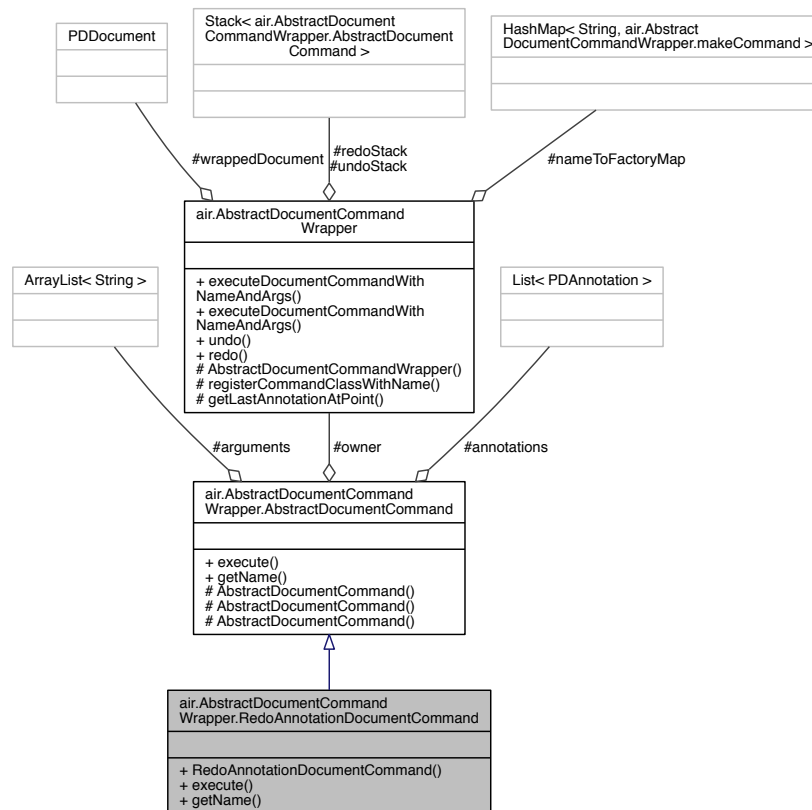
- src/air/DocumentCommandWrapper.java

### 3.11 air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand Class Reference

Inheritance diagram for air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand:



Collaboration diagram for `air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand`:



## Public Member Functions

- **RedoAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, `ArrayList< String >` args)
- [AbstractDocumentCommand](#) `execute` ()
- `String` `getName` ()

## Additional Inherited Members

### 3.11.1 Member Function Documentation

#### 3.11.1.1 `AbstractDocumentCommand` `air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand.execute` ( )

#### Returns

If `execute()` succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.



### 3.11.1.2 String air.AbstractDocumentCommandWrapper.RedoAnnotationDocumentCommand.getName ( )

#### Returns

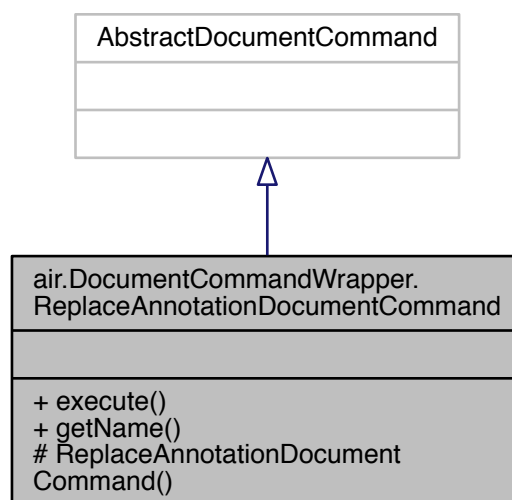
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

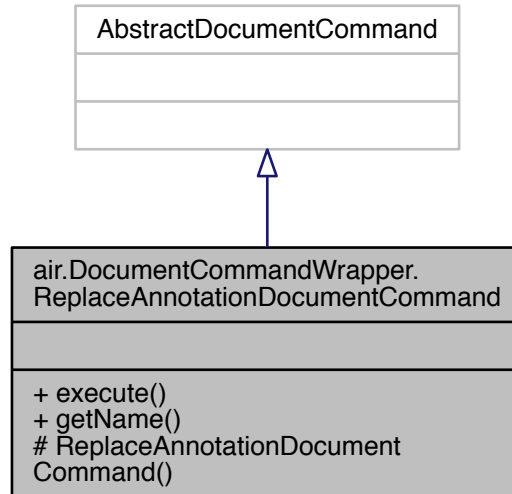
- src/air/AbstractDocumentCommandWrapper.java

## 3.12 air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand:



Collaboration diagram for `air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand`:



## Public Member Functions

- `AbstractDocumentCommand` `execute ()`
- `String` `getName ()`

## Protected Member Functions

- **`ReplaceAnnotationDocumentCommand`** (`AbstractDocumentCommandWrapper` anOwner, `List< PDAnnotation >` someAnnotations, `ArrayList< String >` args)

### 3.12.1 Member Function Documentation

#### 3.12.1.1 `AbstractDocumentCommand` `air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand.execute ( )`

##### Returns

If `execute()` succeeds, a `Command` that is the reciprocal of the receiver is returned. Otherwise, `null` is returned.

## 3.12.1.2 String air.DocumentCommandWrapper.ReplaceAnnotationDocumentCommand.getName ( )

## Returns

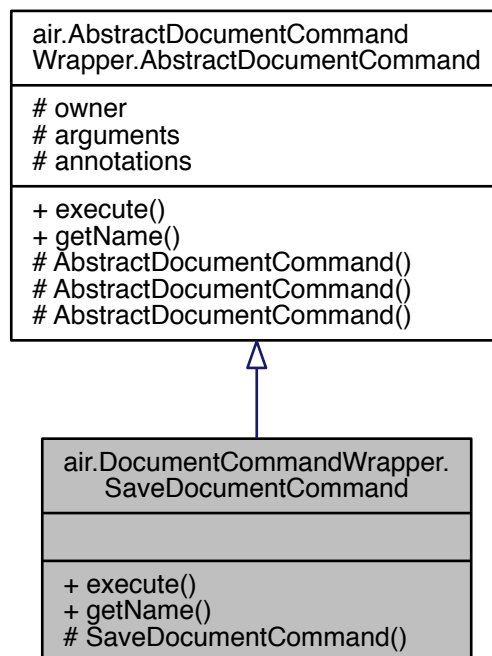
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

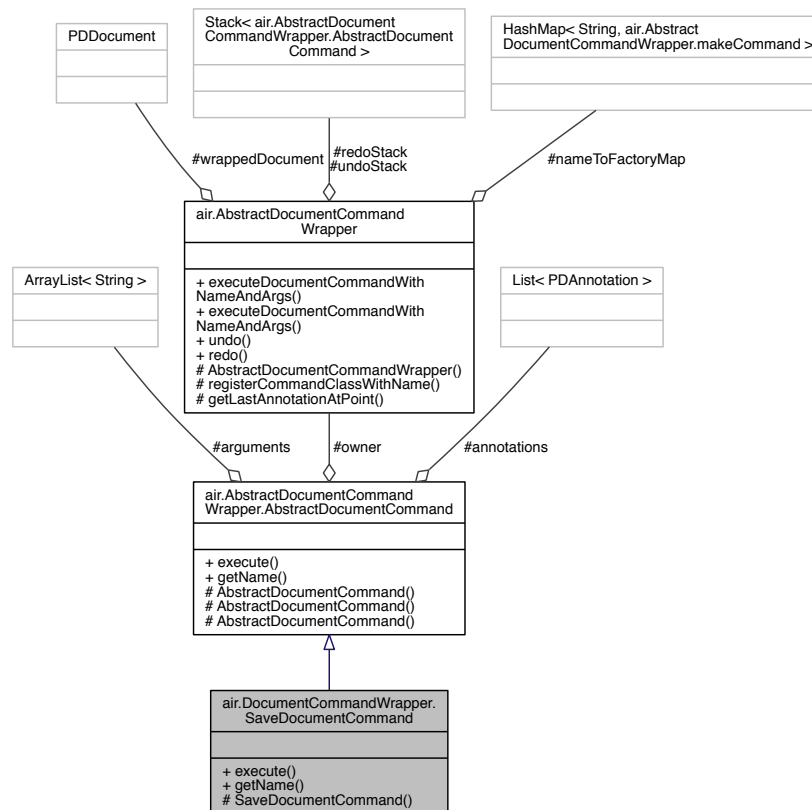
- src/air/DocumentCommandWrapper.java

## 3.13 air.DocumentCommandWrapper.SaveDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.SaveDocumentCommand:



Collaboration diagram for `air.DocumentCommandWrapper.SaveDocumentCommand`:



## Public Member Functions

- [AbstractDocumentCommand execute \(\)](#)
- String [getName \(\)](#)

## Protected Member Functions

- **SaveDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, ArrayList< String > args)

## Additional Inherited Members

### 3.13.1 Member Function Documentation

#### 3.13.1.1 AbstractDocumentCommand `air.DocumentCommandWrapper.SaveDocumentCommand.execute ( )`

##### Returns

If [execute\(\)](#) succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.

## 3.13.1.2 String air.DocumentCommandWrapper.SaveDocumentCommand.getName ( )

## Returns

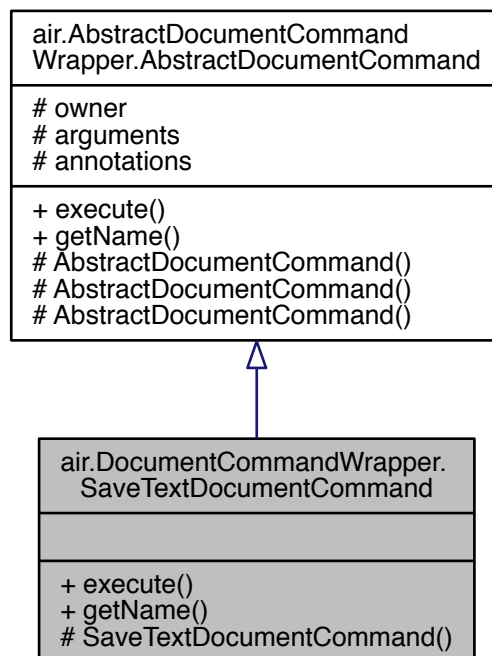
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

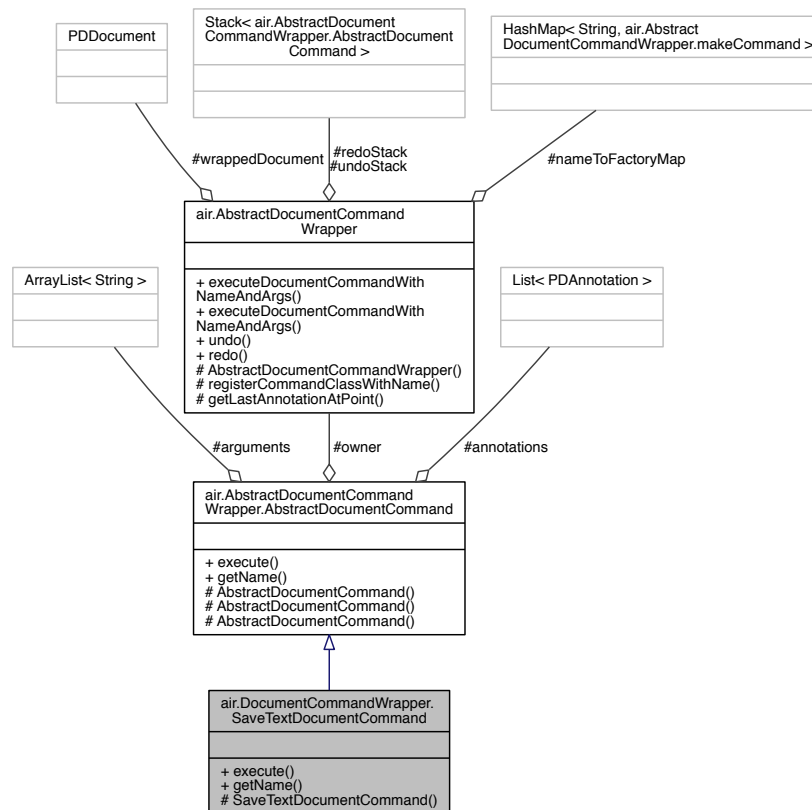
- src/air/DocumentCommandWrapper.java

## 3.14 air.DocumentCommandWrapper.SaveTextDocumentCommand Class Reference

Inheritance diagram for air.DocumentCommandWrapper.SaveTextDocumentCommand:



Collaboration diagram for `air.DocumentCommandWrapper.SaveTextDocumentCommand`:



## Public Member Functions

- [AbstractDocumentCommand execute \(\)](#)
- String [getName \(\)](#)

## Protected Member Functions

- **SaveTextDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, ArrayList< String > args)

## Additional Inherited Members

### 3.14.1 Member Function Documentation

#### 3.14.1.1 AbstractDocumentCommand `air.DocumentCommandWrapper.SaveTextDocumentCommand.execute ( )`

##### Returns

If [execute\(\)](#) succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.

## 3.14.1.2 String air.DocumentCommandWrapper.SaveTextDocumentCommand.getName ( )

## Returns

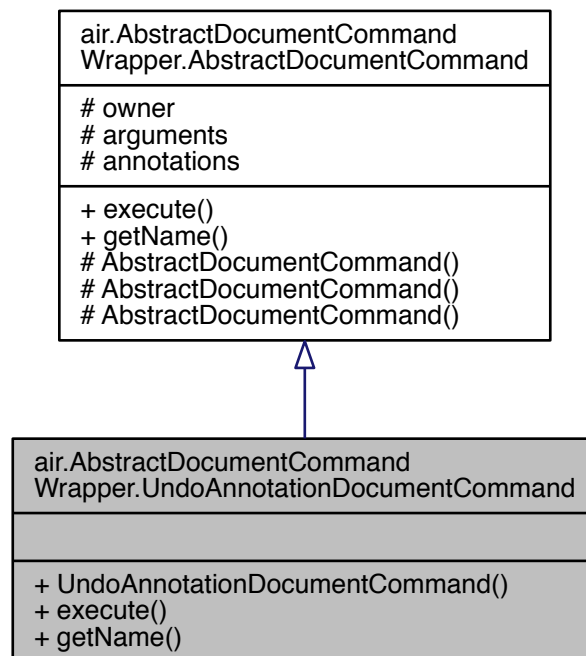
The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from [getName\(\)](#).

The documentation for this class was generated from the following file:

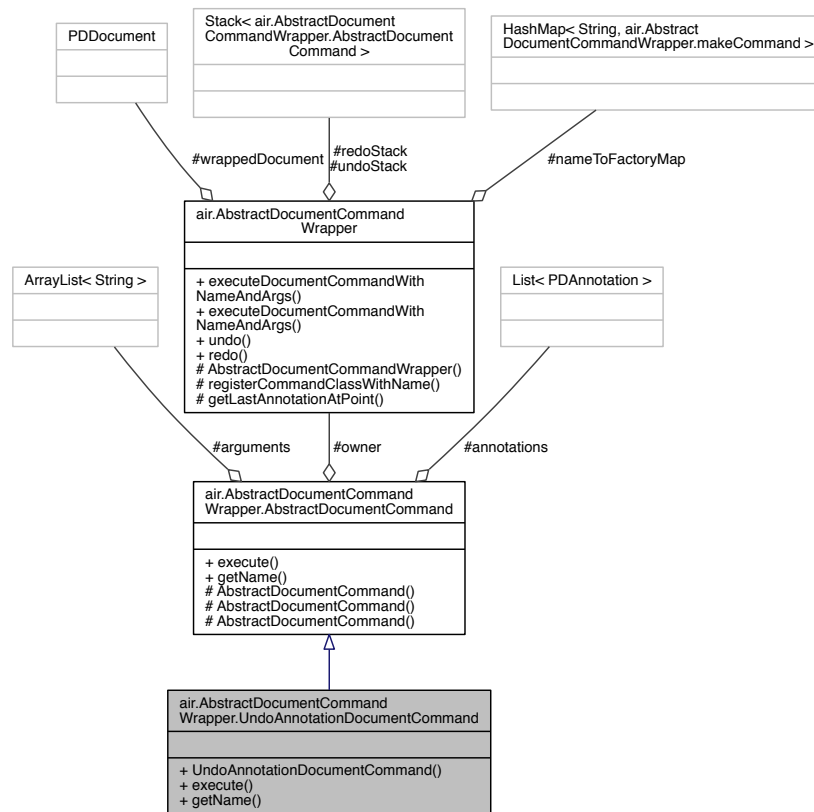
- src/air/DocumentCommandWrapper.java

## 3.15 air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand Class Reference

Inheritance diagram for air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand:



Collaboration diagram for `air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand`:



## Public Member Functions

- **UndoAnnotationDocumentCommand** ([AbstractDocumentCommandWrapper](#) anOwner, `ArrayList< String >` args)
- [AbstractDocumentCommand](#) `execute` ()
- `String` `getName` ()

## Additional Inherited Members

### 3.15.1 Member Function Documentation

#### 3.15.1.1 `AbstractDocumentCommand` `air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand.execute` ( )

#### Returns

If `execute()` succeeds, a Command that is the reciprocal of the receiver is returned. Otherwise, null is returned.



3.15.1.2 `String air.AbstractDocumentCommandWrapper.UndoAnnotationDocumentCommand.getName ( )`**Returns**

The name of the command as it will appear in a user interface for undo and redo operations e.g. "Undo Delete Annotation" where the string after "Undo " is returned from `getName()`.

The documentation for this class was generated from the following file:

- `src/air/AbstractDocumentCommandWrapper.java`



# Index

air.AIRMain, 16  
air.AbstractDocumentCommandWrapper, 8  
air.AbstractDocumentCommandWrapper.Abstract↔  
    DocumentCommand, 5  
air.AbstractDocumentCommandWrapper.makeCommand,  
    22  
air.AbstractDocumentCommandWrapper.RedoAnnotation↔  
    DocumentCommand, 25  
air.AbstractDocumentCommandWrapper.UndoAnnotation↔  
    DocumentCommand, 33  
air.DocumentCommandWrapper, 19  
air.DocumentCommandWrapper.AddBoxAnnotation↔  
    DocumentCommand, 10  
air.DocumentCommandWrapper.AddCircleAnnotation↔  
    DocumentCommand, 12  
air.DocumentCommandWrapper.AddTextAnnotation↔  
    DocumentCommand, 14  
air.DocumentCommandWrapper.DeleteAnnotation↔  
    DocumentCommand, 17  
air.DocumentCommandWrapper.MoveAnnotation↔  
    DocumentCommand, 23  
air.DocumentCommandWrapper.ReplaceAnnotation↔  
    DocumentCommand, 27  
air.DocumentCommandWrapper.SaveDocumentCommand,  
    29  
air.DocumentCommandWrapper.SaveTextDocument↔  
    Command, 31  
air::AIRMain  
    test, 16  
air::AbstractDocumentCommandWrapper::Abstract↔  
    DocumentCommand  
        execute, 7  
        getName, 7  
air::AbstractDocumentCommandWrapper::RedoAnnotation↔  
    DocumentCommand  
        execute, 26  
        getName, 26  
air::AbstractDocumentCommandWrapper::UndoAnnotation↔  
    DocumentCommand  
        execute, 34  
        getName, 34  
air::DocumentCommandWrapper  
    loadDocumentAtPath, 21  
    save, 21  
air::DocumentCommandWrapper::AddBoxAnnotation↔  
    DocumentCommand  
        execute, 11  
        getName, 11  
air::DocumentCommandWrapper::AddCircleAnnotation↔  
    DocumentCommand  
        execute, 13  
        getName, 13  
air::DocumentCommandWrapper::AddTextAnnotation↔  
    DocumentCommand  
        execute, 15  
        getName, 15  
air::DocumentCommandWrapper::DeleteAnnotation↔  
    DocumentCommand  
        execute, 18  
        getName, 18  
air::DocumentCommandWrapper::MoveAnnotation↔  
    DocumentCommand  
        execute, 24  
        getName, 24  
air::DocumentCommandWrapper::ReplaceAnnotation↔  
    DocumentCommand  
        execute, 28  
        getName, 28  
air::DocumentCommandWrapper::SaveDocument↔  
    Command  
        execute, 30  
        getName, 30  
air::DocumentCommandWrapper::SaveTextDocument↔  
    Command  
        execute, 32  
        getName, 32  
execute  
air::AbstractDocumentCommandWrapper::Abstract↔  
    DocumentCommand, 7  
air::AbstractDocumentCommandWrapper::Redo↔  
    AnnotationDocumentCommand, 26  
air::AbstractDocumentCommandWrapper::Undo↔  
    AnnotationDocumentCommand, 34  
air::DocumentCommandWrapper::AddBoxAnnotation↔  
    DocumentCommand, 11  
air::DocumentCommandWrapper::AddCircle↔  
    AnnotationDocumentCommand, 13  
air::DocumentCommandWrapper::AddTextAnnotation↔  
    DocumentCommand, 15

air::DocumentCommandWrapper::DeleteAnnotation↔  
DocumentCommand, 18  
air::DocumentCommandWrapper::MoveAnnotation↔  
DocumentCommand, 24  
air::DocumentCommandWrapper::ReplaceAnnotation↔  
DocumentCommand, 28  
air::DocumentCommandWrapper::SaveDocument↔  
Command, 30  
air::DocumentCommandWrapper::SaveText↔  
DocumentCommand, 32

#### getName

air::AbstractDocumentCommandWrapper::Abstract↔  
DocumentCommand, 7  
air::AbstractDocumentCommandWrapper::Redo↔  
AnnotationDocumentCommand, 26  
air::AbstractDocumentCommandWrapper::Undo↔  
AnnotationDocumentCommand, 34  
air::DocumentCommandWrapper::AddBoxAnnotation↔  
DocumentCommand, 11  
air::DocumentCommandWrapper::AddCircle↔  
AnnotationDocumentCommand, 13  
air::DocumentCommandWrapper::AddTextAnnotation↔  
DocumentCommand, 15  
air::DocumentCommandWrapper::DeleteAnnotation↔  
DocumentCommand, 18  
air::DocumentCommandWrapper::MoveAnnotation↔  
DocumentCommand, 24  
air::DocumentCommandWrapper::ReplaceAnnotation↔  
DocumentCommand, 28  
air::DocumentCommandWrapper::SaveDocument↔  
Command, 30  
air::DocumentCommandWrapper::SaveText↔  
DocumentCommand, 32

#### loadDocumentAtPath

air::DocumentCommandWrapper, 21

#### save

air::DocumentCommandWrapper, 21

#### test

air::AIRMain, 16