

Project BacpFseC

Generated by Doxygen 1.8.6

Mon Apr 6 2015 13:00:13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	BfcPrototype Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	BfcPrototype	8
4.1.2.2	~BfcPrototype	8
4.1.3	Member Function Documentation	8
4.1.3.1	getEndDate	8
4.1.3.2	getMaxTaskInProgress	8
4.1.3.3	getNumOfTaskCancelled	8
4.1.3.4	getNumOfTaskUncomplete	8
4.1.3.5	getRecordLoaded	8
4.1.3.6	getStartDate	9
4.1.3.7	getTasks	9
4.1.3.8	init	9
4.1.3.9	readStates	9
4.1.3.10	readTasks	9
4.1.3.11	run	9
4.1.3.12	setEndDate	9
4.1.3.13	setNumOfTaskCancelled	9
4.1.3.14	setNumOfTaskUncomplete	9
4.1.3.15	setRecordLoaded	9
4.1.3.16	setStartDate	9

4.1.4	Member Data Documentation	10
4.1.4.1	endDate	10
4.1.4.2	maxTaskInProgress	10
4.1.4.3	numOfTaskCancelled	10
4.1.4.4	numOfTaskUncomplete	10
4.1.4.5	recordLoaded	10
4.1.4.6	startDate	10
4.1.4.7	tasks	10
4.2	BfcTerminal Class Reference	10
4.2.1	Member Function Documentation	11
4.2.1.1	instruct	11
4.2.1.2	run	11
4.3	BfcTUI Class Reference	11
4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	12
4.3.2.1	BfcTUI	12
4.3.2.2	~BfcTUI	12
4.3.3	Member Function Documentation	12
4.3.3.1	briefReport	12
4.3.3.2	printDay	12
4.3.3.3	run	12
4.3.3.4	selectAction	13
4.3.3.5	selectTask	14
4.3.3.6	showRecord	14
4.3.3.7	timeline	14
4.3.3.8	timelineHor	14
4.3.3.9	timelineVer	14
4.3.3.10	working	15
4.3.3.11	writeRecord	15
4.4	Date Class Reference	15
4.4.1	Detailed Description	16
4.4.2	Constructor & Destructor Documentation	16
4.4.2.1	Date	16
4.4.2.2	Date	16
4.4.2.3	Date	16
4.4.2.4	~Date	16
4.4.3	Member Function Documentation	16
4.4.3.1	getValue	16
4.4.3.2	isValid	16
4.4.3.3	nextDate	17

4.4.3.4	operator!=	17
4.4.3.5	operator==	17
4.4.3.6	setValue	17
4.4.4	Member Data Documentation	17
4.4.4.1	date	17
4.5	FixtureBfcPrototype Class Reference	17
4.5.1	Member Function Documentation	18
4.5.1.1	testInitAndGetters	18
4.5.1.2	testReadStates	18
4.5.1.3	testReadTasks	18
4.5.1.4	testSetters	18
4.5.1.5	testTasksGetter	18
4.6	FixtureBfcTUI Class Reference	18
4.6.1	Member Function Documentation	19
4.6.1.1	testBriefReport	19
4.6.1.2	testPrintDay	19
4.6.1.3	testSelectTask	19
4.6.1.4	testShowRecord	19
4.6.1.5	testTimelineHor	20
4.6.1.6	testTimelineVer	20
4.6.1.7	testWriteRecord	20
4.7	FixtureDate Class Reference	20
4.7.1	Member Function Documentation	21
4.7.1.1	testCopyConstructor	21
4.7.1.2	testEquivalent	21
4.7.1.3	testGetter	21
4.7.1.4	testIsValid	21
4.7.1.5	testNextDate	21
4.7.1.6	testParaConstructor	21
4.7.1.7	testSetter	21
4.8	FixtureState Class Reference	21
4.8.1	Member Function Documentation	22
4.8.1.1	testContentGetter	22
4.8.1.2	testContentMerge	22
4.8.1.3	testContentSetter	22
4.8.1.4	testCopyConstructor	22
4.8.1.5	testDateGetter	22
4.8.1.6	testDateSetter	22
4.8.1.7	testParaConstructor	23
4.8.1.8	testStateMerge	23

4.9	FixtureTask Class Reference	23
4.9.1	Member Function Documentation	24
4.9.1.1	testCopyConstructor	24
4.9.1.2	testParaConstructor	24
4.9.1.3	testStateGetter	24
4.9.1.4	testStateSetter	24
4.9.1.5	testStatusGetter	24
4.9.1.6	testStatusSetter	24
4.9.1.7	testTaskNameGetter	24
4.9.1.8	testTaskNameSetter	24
4.10	State Class Reference	24
4.10.1	Detailed Description	25
4.10.2	Constructor & Destructor Documentation	25
4.10.2.1	State	25
4.10.2.2	State	25
4.10.2.3	State	25
4.10.2.4	~State	25
4.10.3	Member Function Documentation	25
4.10.3.1	getContent	25
4.10.3.2	getDate	26
4.10.3.3	merge	26
4.10.3.4	merge	26
4.10.3.5	setContent	26
4.10.3.6	setDate	26
4.10.4	Member Data Documentation	26
4.10.4.1	content	26
4.10.4.2	date	26
4.11	Task Class Reference	27
4.11.1	Detailed Description	27
4.11.2	Constructor & Destructor Documentation	27
4.11.2.1	Task	27
4.11.2.2	Task	27
4.11.2.3	Task	28
4.11.2.4	~Task	28
4.11.3	Member Function Documentation	28
4.11.3.1	getStates	28
4.11.3.2	getStatus	28
4.11.3.3	getTaskName	28
4.11.3.4	setStates	28
4.11.3.5	setStatus	28

4.11.3.6	setTaskName	29
4.11.4	Member Data Documentation	29
4.11.4.1	states	29
4.11.4.2	taskName	29
5	File Documentation	31
5.1	BfcPrototype.cc File Reference	31
5.1.1	Detailed Description	31
5.2	BfcPrototype.h File Reference	31
5.2.1	Detailed Description	31
5.3	BfcTerminal.cc File Reference	32
5.3.1	Detailed Description	32
5.4	BfcTerminal.h File Reference	32
5.4.1	Detailed Description	32
5.5	BfcTR.cc File Reference	32
5.5.1	Detailed Description	33
5.6	BfcTUI.cc File Reference	33
5.6.1	Detailed Description	33
5.7	BfcTUI.h File Reference	33
5.7.1	Detailed Description	33
5.8	Date.cc File Reference	34
5.8.1	Detailed Description	34
5.9	Date.h File Reference	34
5.9.1	Detailed Description	34
5.10	FixtureBfcPrototype.h File Reference	34
5.10.1	Detailed Description	35
5.11	FixtureBfcTUI.h File Reference	35
5.11.1	Detailed Description	35
5.12	FixtureDate.h File Reference	35
5.12.1	Detailed Description	36
5.13	FixtureState.h File Reference	36
5.13.1	Detailed Description	36
5.14	FixtureTask.h File Reference	36
5.14.1	Detailed Description	36
5.15	State.cc File Reference	37
5.15.1	Detailed Description	37
5.16	State.h File Reference	37
5.16.1	Detailed Description	37
5.17	Task.cc File Reference	37
5.17.1	Detailed Description	37

5.18 Task.h File Reference	38
5.18.1 Detailed Description	38
5.19 TestDate.cc File Reference	38
5.19.1 Detailed Description	38
5.20 TestState.cc File Reference	38
5.20.1 Detailed Description	39
5.21 TestTask.cc File Reference	39
5.21.1 Detailed Description	39
Index	40

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

BfcPrototype	7
BfcTUI	11
BfcTerminal	10
Date	15
State	24
Task	27
TestFixture	
FixtureBfcPrototype	17
FixtureBfcTUI	18
FixtureDate	20
FixtureState	21
FixtureTask	23

Chapter 2

Class Index

2.1 Class List

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

BfcPrototype		
BfcPrototype	Class is abstract base class	7
BfcTerminal		10
BfcTUI		
BfcTUI	Class is inherited from BfcPrototype	11
Date		
Date	Class is used for time recording	15
FixtureBfcPrototype		17
FixtureBfcTUI		18
FixtureDate		20
FixtureState		21
FixtureTask		23
State		
State	Class is used to record a progress of a task	24
Task		
Task	Class for actual recording of tasks	27

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

BfcPrototype.cc	Implementation for BfcPrototype Class	31
BfcPrototype.h	Header file of BfcPrototype Class	31
BfcTerminal.cc	Implementation for BfcTerminal Class	32
BfcTerminal.h	Header file of BfcTerminal inherited from BfcTUI	32
BfcTR.cc	Executable for BacpFesC in terminal	32
BfcTUI.cc	Implementation for BfcTUI Class	33
BfcTUI.h	Header file of BfcTUI inherited from BfcPrototype	33
Date.cc	Implementation of Date Class	34
Date.h	Header file of Date Class	34
FixtureBfcPrototype.h	CppUnit Fixture setup for BfcPrototype class	34
FixtureBfcTUI.h	CppUnit Fixture setup for BfcTUI class	35
FixtureDate.h	CppUnit Fixture setup for Date class	35
FixtureState.h	CppUnit Fixture setup for State class	36
FixtureTask.h	CppUnit Fixture setup for Task class	36
State.cc	Implementation of State Class	37
State.h	Header file of State Class	37
Task.cc	Implementation for Task Class	37
Task.h	Header file of Task Class	38
TestDate.cc	Test program for Date Class	38

TestState.cc	
Test program for State Class	38
TestTask.cc	
Test program for Task Class	39

Chapter 4

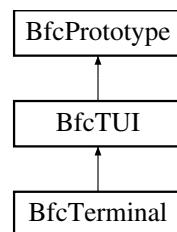
Class Documentation

4.1 BfcPrototype Class Reference

[BfcPrototype](#) Class is abstract base class.

```
#include <BfcPrototype.h>
```

Inheritance diagram for BfcPrototype:



Public Member Functions

- [BfcPrototype](#) ()
- [~BfcPrototype](#) ()
- const int [getMaxTaskInProgress](#) ()
- void [setRecordLoaded](#) (bool rl)
- bool [getRecordLoaded](#) ()
- void [setStartDate](#) (const [Date](#) &sd)
- [Date](#) [getStartDate](#) ()
- void [setEndDate](#) (const [Date](#) &ed)
- [Date](#) [getEndDate](#) ()
- void [setNumOfTaskUncomplete](#) (int ntu)
- int [getNumOfTaskUncomplete](#) ()
- void [setNumOfTaskCancelled](#) (int ntc)
- int [getNumOfTaskCancelled](#) ()
- std::vector< [Task](#) > & [getTasks](#) ()
- virtual void [readStates](#) (std::istream &is, [Task](#) &bk)
- virtual void [readTasks](#) (std::istream &is, std::vector< [Task](#) > &ts)
- virtual void [run](#) ()

Protected Member Functions

- void [init](#) ()

Private Attributes

- const int `maxTaskInProgress` = 15
- bool `recordLoaded`
- Date `startDate`
- Date `endDate`
- int `numOfTaskUncomplete`
- int `numOfTaskCancelled`
- std::vector< Task > `tasks`

4.1.1 Detailed Description

`BfcPrototype` Class is abstract base class.

`BfcPrototype` Class serves as the abstract base class for variety `BacpfsecTUI` and `BacpfsecGUI` would inherited from `BfcPrototype`

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `BfcPrototype::BfcPrototype ()`

Default constructor

4.1.2.2 `BfcPrototype::~~BfcPrototype ()`

Destructor

4.1.3 Member Function Documentation

4.1.3.1 `Date BfcPrototype::getEndDate ()`

Getter for `endDate`

4.1.3.2 `const int BfcPrototype::getMaxTaskInProgress ()`

Getter for `maxTaskInProgress`

4.1.3.3 `int BfcPrototype::getNumOfTaskCancelled ()`

Getter for `numOfTaskCancelled`

4.1.3.4 `int BfcPrototype::getNumOfTaskUncomplete ()`

Getter for `numOfTaskUncomplete`

4.1.3.5 `bool BfcPrototype::getRecordLoaded ()`

Getter for `recordLoaded`

4.1.3.6 Date BfcPrototype::getStartDate ()

Getter for startDate

4.1.3.7 std::vector< Task > & BfcPrototype::getTasks ()

Reference getter for vector tasks

4.1.3.8 void BfcPrototype::init () [protected]

Protected initializer

Helper to initialize all the private member

4.1.3.9 void BfcPrototype::readStates (std::istream & is, Task & bk) [virtual]

Virtual [readStates\(\)](#)

Get the record from stream and update the task and private members

4.1.3.10 void BfcPrototype::readTasks (std::istream & is, std::vector< Task > & ts) [virtual]

Virtual [readTasks\(\)](#)

Get the record from stream and update the tasks and private members

4.1.3.11 void BfcPrototype::run () [virtual]

Virtual [run\(\)](#)

Implemented through inheritance

Reimplemented in [BfcTUI](#), and [BfcTerminal](#).

4.1.3.12 void BfcPrototype::setEndDate (const Date & ed)

Setter for endDate

4.1.3.13 void BfcPrototype::setNumOfTaskCancelled (int ntc)

Setter for numOfTaskCancelled

4.1.3.14 void BfcPrototype::setNumOfTaskUncomplete (int ntu)

Setter for numOfTaskUncomplete

4.1.3.15 void BfcPrototype::setRecordLoaded (bool rl)

Setter for recordLoaded

4.1.3.16 void BfcPrototype::setStartDate (const Date & sd)

Setter for startDate

4.1.4 Member Data Documentation

4.1.4.1 `Date BfcPrototype::endDate` [private]

The last date of recorded readed

4.1.4.2 `const int BfcPrototype::maxTaskInProgress = 15` [private]

Const limit of tasks in progress

4.1.4.3 `int BfcPrototype::numOfTaskCancelled` [private]

The number of tasks cancelled

4.1.4.4 `int BfcPrototype::numOfTaskUncomplete` [private]

The number of tasks in progress

4.1.4.5 `bool BfcPrototype::recordLoaded` [private]

Boolean indicate if record is loaded

4.1.4.6 `Date BfcPrototype::startDate` [private]

The first date of recorded readed

4.1.4.7 `std::vector<Task> BfcPrototype::tasks` [private]

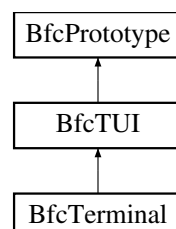
Storage of the tasks as a vector

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

- [BfcPrototype.h](#)
- [BfcPrototype.cc](#)

4.2 BfcTerminal Class Reference

Inheritance diagram for BfcTerminal:



Public Member Functions

- `bool instruct ()`
- `void run ()`

Additional Inherited Members

4.2.1 Member Function Documentation

4.2.1.1 `bool BfcTerminal::instruct ()`

instruct to dictate instructions

Returns

boolean as valid instruction

This function based on

4.2.1.2 `void BfcTerminal::run () [virtual]`

run as the runner of [BfcTUI](#)

Reimplemented from [BfcTUI](#).

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

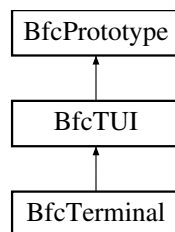
- [BfcTerminal.h](#)
- [BfcTerminal.cc](#)

4.3 BfcTUI Class Reference

[BfcTUI](#) Class is inherited from [BfcPrototype](#).

```
#include <BfcTUI.h>
```

Inheritance diagram for [BfcTUI](#):



Public Member Functions

- [BfcTUI \(\)](#)
- [~BfcTUI \(\)](#)
- void [printDay](#) (std::ostream &os, [Date](#) d, int width)
- void [showRecord](#) (std::ostream &os, std::vector< [Task](#) > &ts, int s)
- void [writeRecord](#) (std::ostream &os, std::vector< [Task](#) > &ts)
- void [briefReport](#) (std::ostream &os, std::vector< [Task](#) > &ts)
- void [timeline](#) (std::ostream &os, std::vector< [Task](#) > &ts, [Date](#) s, [Date](#) e, bool m)
- int [selectTask](#) (std::ostream &os, std::istream &is, std::vector< [Task](#) > &ts)
- void [selectAction](#) (std::ostream &os, std::istream &is, std::vector< [Task](#) > &ts)
- void [working](#) (std::ostream &os, std::istream &is, [Task](#) &t)
- virtual void [run](#) ()

Private Member Functions

- void [timelineVer](#) (std::ostream &os, std::vector< [Task](#) > &ts, [Date](#) s, [Date](#) e)
- void [timelineHor](#) (std::ostream &os, std::vector< [Task](#) > &ts, [Date](#) s, [Date](#) e)

Additional Inherited Members

4.3.1 Detailed Description

[BfcTUI](#) Class is inherited from [BfcPrototype](#).

[BfcTUI](#) provides some interesting functions to build the text-based UI This is one main variant of the Bacpfsec Project main

4.3.2 Constructor & Destructor Documentation

4.3.2.1 [BfcTUI::BfcTUI](#) ()

Default constructor

4.3.2.2 [BfcTUI::~~BfcTUI](#) ()

Destructor

4.3.3 Member Function Documentation

4.3.3.1 void [BfcTUI::briefReport](#) (std::ostream & *os*, std::vector< [Task](#) > & *ts*)

briefReport to ostream

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector

4.3.3.2 void [BfcTUI::printDay](#) (std::ostream & *os*, [Date](#) *d*, int *width*)

Print given date with given width

Parameters

<i>os</i>	ostream
<i>d</i>	a Date object
<i>width</i>	desired output width

4.3.3.3 void [BfcTUI::run](#) () [virtual]

Virtual [run\(\)](#)

Implemented through inheritance

Reimplemented from [BfcPrototype](#).

Reimplemented in [BfcTerminal](#).

4.3.3.4 void BfcTUI::selectAction (std::ostream & *os*, std::istream & *is*, std::vector< Task > & *ts*)

selectAction to select proper action basd on a task status

Parameters

<i>os</i>	ostream
<i>is</i>	istream
<i>ts</i>	reference to a task vector

4.3.3.5 `int BfcTUI::selectTask (std::ostream & os, std::istream & is, std::vector< Task > & ts)`

selectTask function select uncompleted task

Parameters

<i>os</i>	ostream
<i>is</i>	istream
<i>ts</i>	reference to a task vector

Returns

integer index for a task

4.3.3.6 `void BfcTUI::showRecord (std::ostream & os, std::vector< Task > & ts, int s)`

showRecord to ostream

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector
<i>s</i>	status integer

4.3.3.7 `void BfcTUI::timeline (std::ostream & os, std::vector< Task > & ts, Date s, Date e, bool m)`

timeline function with variant

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector
<i>s</i>	start Date input from user
<i>e</i>	end Date input from user
<i>m</i>	boolean selector, 0:horizontal, 1:vertical

4.3.3.8 `void BfcTUI::timelineHor (std::ostream & os, std::vector< Task > & ts, Date s, Date e)` `[private]`

Private helper to build horizontal timeline

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector
<i>s</i>	start Date input from user
<i>e</i>	end Date input from user

4.3.3.9 `void BfcTUI::timelineVer (std::ostream & os, std::vector< Task > & ts, Date s, Date e)` `[private]`

Private helper to build vertical timeline

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector
<i>s</i>	start Date input from user
<i>e</i>	end Date input from user

4.3.3.10 void BfcTUI::working (std::ostream & *os*, std::istream & *is*, Task & *t*)

working to perform proper action on a task

Parameters

<i>os</i>	ostream
<i>is</i>	istream
<i>t</i>	reference to a task

4.3.3.11 void BfcTUI::writeRecord (std::ostream & *os*, std::vector< Task > & *ts*)

writeRecord to ostream

Parameters

<i>os</i>	ostream
<i>ts</i>	reference to a task vector

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

- [BfcTUI.h](#)
- [BfcTUI.cc](#)

4.4 Date Class Reference

[Date](#) Class is used for time recording.

```
#include <Date.h>
```

Public Member Functions

- [Date](#) ()
- [Date](#) (int d)
- [Date](#) (const [Date](#) &dt)
- [~Date](#) ()
- void [setValue](#) (int d)
- int [getValue](#) ()
- void [nextDate](#) ()
- bool [operator==](#) ([Date](#) d)
- bool [operator!=](#) ([Date](#) d)

Private Member Functions

- bool [isValid](#) ()

Private Attributes

- int [date](#)

4.4.1 Detailed Description

[Date](#) Class is used for time recording.

[Date](#) Class provides supplementary operations based on the <time> library

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `Date::Date ()`

Default constructor

Note

Default value is 150301 (i.e. 2015/03/01)

4.4.2.2 `Date::Date (int d)`

Parameterized constructor

Parameters

<i>d</i>	Value stands for a date
----------	-------------------------

set back to default for invalid date

4.4.2.3 `Date::Date (const Date & dt)`

Copy constructor

4.4.2.4 `Date::~~Date ()`

Destructor

4.4.3 Member Function Documentation

4.4.3.1 `int Date::getValue ()`

Getter

Returns

Value of the private date

4.4.3.2 `bool Date::isValid ()` `[private]`

[Date](#) check

Private helper to test if a date is valid

4.4.3.3 void Date::nextDate ()

Proceed date to next valid date

4.4.3.4 bool Date::operator!= (Date *d*)

Overload != operator param *d* another date

4.4.3.5 bool Date::operator== (Date *d*)

Overload == operator param *d* another date

4.4.3.6 void Date::setValue (int *d*)

Setter

Parameters

<i>d</i>	Value stands for a date
----------	-------------------------

4.4.4 Member Data Documentation

4.4.4.1 int Date::date [private]

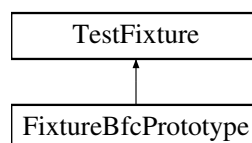
Storage of date value

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

- [Date.h](#)
- [Date.cc](#)

4.5 FixtureBfcPrototype Class Reference

Inheritance diagram for FixtureBfcPrototype:



Public Member Functions

- **CPPUNIT_TEST_SUITE** ([FixtureBfcPrototype](#))
- **CPPUNIT_TEST** ([testInitAndGetters](#))
- **CPPUNIT_TEST** ([testSetters](#))
- **CPPUNIT_TEST** ([testTasksGetter](#))
- **CPPUNIT_TEST** ([testReadStates](#))
- **CPPUNIT_TEST** ([testReadTasks](#))
- **CPPUNIT_TEST_SUITE_END** ()
- void **setUp** ()
- void **tearDown** ()

- void [testInitAndGetters](#) ()
- void [testSetters](#) ()
- void [testTasksGetter](#) ()
- void [testReadStates](#) ()
- void [testReadTasks](#) ()

Private Attributes

- [BfcPrototype](#) * **bfc**

4.5.1 Member Function Documentation

4.5.1.1 void FixtureBfcPrototype::testInitAndGetters () [inline]

Tests for Getters and init

Test1: getRecordLoaded() Test2: getStartDate() Test3: getEndDate() Test4: getNumOfTaskUncomplete() Test5: getNumOfTaskCancelled()

4.5.1.2 void FixtureBfcPrototype::testReadStates () [inline]

Test for BfcPrototype::readStates(istream, Task&)

Test input: [~P2] 20150302 [~P3 + ~P5] 20150304 1

4.5.1.3 void FixtureBfcPrototype::testReadTasks () [inline]

Test for BfcPrototype::readStates(istream, vector<Task>&)

Test input: Book1 [~P2] 20150302 [~P3 + ~P5] 20150304 1 Book2 [~C1] 20150101 [~C3] 20150502 0

4.5.1.4 void FixtureBfcPrototype::testSetters () [inline]

Tests for Setters

Test1: setRecordLoaded() Test2: setStartDate() Test3: setEndDate() Test4: setNumOfTaskUncomplete() Test5: setNumOfTaskCancelled()

4.5.1.5 void FixtureBfcPrototype::testTasksGetter () [inline]

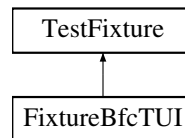
Test for [BfcPrototype::getTasks\(\)](#)

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

- [FixtureBfcPrototype.h](#)

4.6 FixtureBfcTUI Class Reference

Inheritance diagram for FixtureBfcTUI:



Public Member Functions

- **CPPUNIT_TEST_SUITE** (*FixtureBfcTUI*)
- **CPPUNIT_TEST** (*testPrintDay*)
- **CPPUNIT_TEST** (*testShowRecord*)
- **CPPUNIT_TEST** (*testWriteRecord*)
- **CPPUNIT_TEST** (*testBriefReport*)
- **CPPUNIT_TEST** (*testTimelineVer*)
- **CPPUNIT_TEST** (*testTimelineHor*)
- **CPPUNIT_TEST** (*testSelectTask*)
- **CPPUNIT_TEST_SUITE_END** ()
- void **setUp** ()
- void **tearDown** ()
- void *testPrintDay* ()
- void *testShowRecord* ()
- void *testWriteRecord* ()
- void *testBriefReport* ()
- void *testTimelineVer* ()
- void *testTimelineHor* ()
- void *testSelectTask* ()

Private Attributes

- *BfcTUI* * **tui**

4.6.1 Member Function Documentation

4.6.1.1 void FixtureBfcTUI::testBriefReport () [inline]

Tests for *BfcTUI::briefReport*(std::ostream&,std::vector<Task>&);

4.6.1.2 void FixtureBfcTUI::testPrintDay () [inline]

Tests for *BfcTUI::printDay*(ostream&, Date, int);

4.6.1.3 void FixtureBfcTUI::testSelectTask () [inline]

Tests for *selectTask*()

Test 1: select a uncompleted task would return correct number Test 2: the prompt information should exclude finished tasks

4.6.1.4 void FixtureBfcTUI::testShowRecord () [inline]

Tests for *BfcTUI::showRecord*(std::ostream&,std::vector<Task>&,int);

4.6.1.5 void FixtureBfcTUI::testTimelineHor () [inline]

Tests for horizontal timeline

4.6.1.6 void FixtureBfcTUI::testTimelineVer () [inline]

Tests for vertical timeline

4.6.1.7 void FixtureBfcTUI::testWriteRecord () [inline]

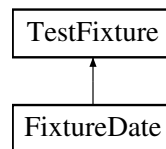
Tests for [BfcTUI::writeRecord\(std::ostream&,std::vector<Task>&\);](#)

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

- [FixtureBfcTUI.h](#)

4.7 FixtureDate Class Reference

Inheritance diagram for FixtureDate:



Public Member Functions

- **CPPUNIT_TEST_SUITE** ([FixtureDate](#))
- **CPPUNIT_TEST** ([testGetter](#))
- **CPPUNIT_TEST** ([testSetter](#))
- **CPPUNIT_TEST** ([testParaConstructor](#))
- **CPPUNIT_TEST** ([testCopyConstructor](#))
- **CPPUNIT_TEST** ([testNextDate](#))
- **CPPUNIT_TEST** ([testIsValid](#))
- **CPPUNIT_TEST** ([testEquivalent](#))
- **CPPUNIT_TEST_SUITE_END** ()
- void **setUp** ()
- void **tearDown** ()
- void [testGetter](#) ()
- void [testSetter](#) ()
- void [testParaConstructor](#) ()
- void [testCopyConstructor](#) ()
- void [testNextDate](#) ()
- void [testIsValid](#) ()
- void [testEquivalent](#) ()

Private Attributes

- [Date](#) * **dt**

4.7.1 Member Function Documentation

4.7.1.1 void FixtureDate::testCopyConstructor () [inline]

Test for [Date::Date\(const Date& dt\)](#)

4.7.1.2 void FixtureDate::testEquivalent () [inline]

Test for equivalent check

Test case 1: 20150201==20150201 Test case 2: 20150202!=20150201

4.7.1.3 void FixtureDate::testGetter () [inline]

Test for [Date::getDate\(\)](#)

Default date is 20150301

4.7.1.4 void FixtureDate::testIsValid () [inline]

Test for [Date::isValid\(\)](#)

Test case 1: 20150229(invalid)->20150301 Test case 2: 20160229(valid)->20160229 Test case 3: 20151930(invalid)->20150301

4.7.1.5 void FixtureDate::testNextDate () [inline]

Test for [Date::nextDate\(\)](#)

Test case 1: 20150228->20150301 Test case 2: 20160228->20160229 Test case 3: 20150930->20151001 Test case 4: 20151215->20151216

4.7.1.6 void FixtureDate::testParaConstructor () [inline]

Test for [Date::Date\(int d\)](#)

4.7.1.7 void FixtureDate::testSetter () [inline]

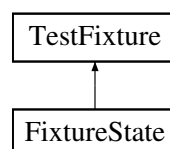
Test for [Date::setDate\(int d\)](#)

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

- [FixtureDate.h](#)

4.8 FixtureState Class Reference

Inheritance diagram for FixtureState:



Public Member Functions

- **CPPUNIT_TEST_SUITE** ([FixtureState](#))
- **CPPUNIT_TEST** ([testDateGetter](#))
- **CPPUNIT_TEST** ([testDateSetter](#))
- **CPPUNIT_TEST** ([testContentGetter](#))
- **CPPUNIT_TEST** ([testContentSetter](#))
- **CPPUNIT_TEST** ([testParaConstructor](#))
- **CPPUNIT_TEST** ([testCopyConstructor](#))
- **CPPUNIT_TEST** ([testContentMerge](#))
- **CPPUNIT_TEST** ([testStateMerge](#))
- **CPPUNIT_TEST_SUITE_END** ()
- void **setUp** ()
- void **tearDown** ()
- void [testDateSetter](#) ()
- void [testDateGetter](#) ()
- void [testContentSetter](#) ()
- void [testContentGetter](#) ()
- void [testParaConstructor](#) ()
- void [testCopyConstructor](#) ()
- void [testContentMerge](#) ()
- void [testStateMerge](#) ()

Private Attributes

- [State](#) * **st**

4.8.1 Member Function Documentation

4.8.1.1 void [FixtureState::testContentGetter](#) () [[inline](#)]

Test for [State::getContent\(\)](#)

4.8.1.2 void [FixtureState::testContentMerge](#) () [[inline](#)]

Test for [State::merge\(std::string s\)](#)

4.8.1.3 void [FixtureState::testContentSetter](#) () [[inline](#)]

Test for [State::setContent\(std::string s\)](#)

4.8.1.4 void [FixtureState::testCopyConstructor](#) () [[inline](#)]

Test for [State::State\(const State& st\)](#)

4.8.1.5 void [FixtureState::testDateGetter](#) () [[inline](#)]

Test for [State::getDate\(\)](#)

4.8.1.6 void [FixtureState::testDateSetter](#) () [[inline](#)]

Test for [State::setDate\(const Date& dt\)](#)

4.8.1.7 void FixtureState::testParaConstructor () [inline]

Test for [State::State\(std::string s, const Date& dt\)](#)

4.8.1.8 void FixtureState::testStateMerge () [inline]

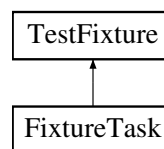
Test for [State::merge\(State st\)](#)

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

- [FixtureState.h](#)

4.9 FixtureTask Class Reference

Inheritance diagram for FixtureTask:



Public Member Functions

- **CPPUNIT_TEST_SUITE** ([FixtureTask](#))
- **CPPUNIT_TEST** ([testTaskNameGetter](#))
- **CPPUNIT_TEST** ([testTaskNameSetter](#))
- **CPPUNIT_TEST** ([testStatusGetter](#))
- **CPPUNIT_TEST** ([testStatusSetter](#))
- **CPPUNIT_TEST** ([testStateGetter](#))
- **CPPUNIT_TEST** ([testStateSetter](#))
- **CPPUNIT_TEST** ([testParaConstructor](#))
- **CPPUNIT_TEST** ([testCopyConstructor](#))
- **CPPUNIT_TEST_SUITE_END** ()
- void **setUp** ()
- void **tearDown** ()
- void [testTaskNameGetter](#) ()
- void [testTaskNameSetter](#) ()
- void [testStatusGetter](#) ()
- void [testStatusSetter](#) ()
- void [testStateGetter](#) ()
- void [testStateSetter](#) ()
- void [testParaConstructor](#) ()
- void [testCopyConstructor](#) ()

Private Attributes

- [Task](#) * **t**

4.9.1 Member Function Documentation

4.9.1.1 void FixtureTask::testCopyConstructor () [inline]

Test for Task::testCopyConstructor(const Task&)

4.9.1.2 void FixtureTask::testParaConstructor () [inline]

Test for Task::testParaConstructor

4.9.1.3 void FixtureTask::testStateGetter () [inline]

Test for Task::testStatusGetter()

4.9.1.4 void FixtureTask::testStateSetter () [inline]

Test for Task::testStatusSetter(const& vector<state>)

4.9.1.5 void FixtureTask::testStatusGetter () [inline]

Test for Task::testStatusGetter()

4.9.1.6 void FixtureTask::testStatusSetter () [inline]

Test for Task::testStatusSetter(int)

4.9.1.7 void FixtureTask::testTaskNameGetter () [inline]

Test for [Task::getTaskName\(\)](#)

4.9.1.8 void FixtureTask::testTaskNameSetter () [inline]

Test for Task::setTaskName(string)

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

- [FixtureTask.h](#)

4.10 State Class Reference

[State](#) Class is used to record a progress of a task.

```
#include <State.h>
```

Public Member Functions

- [State](#) ()
- [State](#) (std::string s, const [Date](#) &dt)
- [State](#) (const [State](#) &st)
- [~State](#) ()
- void [setDate](#) (const [Date](#) &dt)

- [Date](#) `getDate ()`
- void [setContent](#) (std::string s)
- std::string [getContent \(\)](#)
- void [merge](#) (std::string s)
- void [merge](#) ([State](#) st)

Private Attributes

- std::string [content](#)
- [Date](#) [date](#)

4.10.1 Detailed Description

[State](#) Class is used to record a progress of a task.

[State](#) Class records both progress detail and performing date. [State](#) Class also provides some basic operations related with a progress.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 `State::State ()`

Default constructor

4.10.2.2 `State::State (std::string s, const Date & dt)`

Parameterized constructor

Parameters

<i>s</i>	string content for a progress
<i>dt</i>	a Date object

4.10.2.3 `State::State (const State & st)`

Copy constructor

4.10.2.4 `State::~~State ()`

Destructor

4.10.3 Member Function Documentation

4.10.3.1 `std::string State::getContent ()`

Getter for content

Returns

A string with details from private content

4.10.3.2 Date State::getDate ()

Getter for date

Returns

A [Date](#) object with details from private date

4.10.3.3 void State::merge (std::string s)

Merge another content to this state

Parameters

<i>s</i>	A string for another progress detail
----------	--------------------------------------

4.10.3.4 void State::merge (State st)

Merge another state at the same day

Parameters

<i>st</i>	A state which is the same as current
-----------	--------------------------------------

Note

Designed for flexibility

4.10.3.5 void State::setContent (std::string s)

Setter for content

Parameters

<i>s</i>	A string for progress detail
----------	------------------------------

4.10.3.6 void State::setDate (const Date & dt)

Setter for date

Parameters

<i>dt</i>	A Date object
-----------	-------------------------------

4.10.4 Member Data Documentation

4.10.4.1 std::string State::content [private]

Storage of progress detail

4.10.4.2 Date State::date [private]

Storage of date value

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

- [State.h](#)
- [State.cc](#)

4.11 Task Class Reference

[Task](#) Class for actual recording of tasks.

```
#include <Task.h>
```

Public Member Functions

- [Task](#) ()
- [Task](#) (std::string tn, const std::vector< [State](#) > &st, int s)
- [Task](#) (const [Task](#) &t)
- [~Task](#) ()
- void [setStates](#) (const std::vector< [State](#) > &sts)
- std::vector< [State](#) > & [getStates](#) ()
- void [setTaskName](#) (std::string s)
- std::string [getTaskName](#) ()
- void [setStatus](#) (int f)
- int [getStatus](#) ()

Private Attributes

- std::string [taskName](#)
- std::vector< [State](#) > [states](#)
- int [status](#)

4.11.1 Detailed Description

[Task](#) Class for actual recording of tasks.

[Task](#) Class records several states on different dates for a single task. [Task](#) Class also provides some basic operations related with a task.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 [Task::Task](#) ()

Default constructor

4.11.2.2 [Task::Task](#) (std::string tn, const std::vector< [State](#) > &st, int s)

Parameterized constructor

Parameters

<i>tn</i>	string content for a task name
<i>st</i>	a vector of states

<i>s</i>	a boolean to indicate the task status
----------	---------------------------------------

4.11.2.3 Task::Task (const Task & t)

Copy constructor

4.11.2.4 Task::~~Task ()

Destructor

4.11.3 Member Function Documentation

4.11.3.1 std::vector< State > & Task::getStates ()

Getter for states

Returns

A reference to vector of recorded states

4.11.3.2 int Task::getStatus ()

Getter for task status

Returns

An integer stands for status

4.11.3.3 std::string Task::getTaskName ()

Getter for task name

Returns

A string with task name

4.11.3.4 void Task::setStates (const std::vector< State > & sts)

Setter for states

Parameters

<i>dt</i>	A vector of states
-----------	--------------------

4.11.3.5 void Task::setStatus (int f)

Setter for task status

Parameters

<i>s</i>	An integer stands for status
----------	------------------------------

4.11.3.6 void Task::setTaskName (std::string *s*)

Setter for task name

Parameters

<i>s</i>	A string for task name
----------	------------------------

4.11.4 Member Data Documentation

4.11.4.1 std::vector<State> Task::states [private]

Storage of states in vector

4.11.4.2 std::string Task::taskName [private]

Storage of task name

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

- [Task.h](#)
- [Task.cc](#)

Chapter 5

File Documentation

5.1 BfcPrototype.cc File Reference

Implementation for [BfcPrototype](#) Class.

```
#include "BfcPrototype.h"
```

5.1.1 Detailed Description

Implementation for [BfcPrototype](#) Class. Bacpfsec Rick

Date

20150405 [BfcPrototype](#) is the base class for TUI and GUI

5.2 BfcPrototype.h File Reference

Header file of [BfcPrototype](#) Class.

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include "Date.h"
#include "Task.h"
#include <vector>
```

Classes

- class [BfcPrototype](#)
[BfcPrototype](#) Class is abstract base class.

5.2.1 Detailed Description

Header file of [BfcPrototype](#) Class. Bacpfsec Rick

Date

20150405 [BfcPrototype](#) Class is used as abstract base class

5.3 BfcTerminal.cc File Reference

Implementation for [BfcTerminal](#) Class.

```
#include "BfcTerminal.h"  
#include <sstream>  
#include <time.h>  
#include <fstream>
```

5.3.1 Detailed Description

Implementation for [BfcTerminal](#) Class. Bacpfsec Rick

Date

20150406 [BfcTerminal](#) Class used the terminal for user-interface

5.4 BfcTerminal.h File Reference

Header file of [BfcTerminal](#) inherited from [BfcTUI](#).

```
#include <stdio.h>  
#include <stdlib.h>  
#include <iostream>  
#include "Date.h"  
#include "Task.h"  
#include "BfcPrototype.h"  
#include "BfcTUI.h"  
#include <vector>
```

Classes

- class [BfcTerminal](#)

5.4.1 Detailed Description

Header file of [BfcTerminal](#) inherited from [BfcTUI](#). Bacpfsec Rick

Date

20150406 [BfcTerminal](#) Class used the terminal for user-interface

5.5 BfcTR.cc File Reference

Executable for BacpFesC in terminal.

```
#include "BfcPrototype.h"  
#include "BfcTUI.h"  
#include "BfcTerminal.h"
```


Functions

- int **main** ()

5.5.1 Detailed Description

Executable for BacpFesC in terminal. Bacpfsec Rick

Date

20150406 This is the program to run the terminal version of BacpFesC project

5.6 BfcTUI.cc File Reference

Implementation for [BfcTUI](#) Class.

```
#include "BfcTUI.h"
#include <sstream>
#include <iomanip>
#include <time.h>
#include <fstream>
```

5.6.1 Detailed Description

Implementation for [BfcTUI](#) Class. Bacpfsec Rick

Date

20150405 [BfcTUI](#) is the TUI version of program

5.7 BfcTUI.h File Reference

Header file of [BfcTUI](#) inherited from [BfcPrototype](#).

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include "Date.h"
#include "Task.h"
#include "BfcPrototype.h"
#include <vector>
```

Classes

- class [BfcTUI](#)
[BfcTUI](#) Class is inherited from [BfcPrototype](#).

5.7.1 Detailed Description

Header file of [BfcTUI](#) inherited from [BfcPrototype](#). Bacpfsec Rick

Date

20150405 [BfcTUI](#) Class is used to build the text-based user interface

5.8 Date.cc File Reference

Implementation of [Date](#) Class.

```
#include "Date.h"
```

5.8.1 Detailed Description

Implementation of [Date](#) Class. Bacpfsec Rick

Date

20150324 [Date](#) Class is used for [State](#) Class

5.9 Date.h File Reference

Header file of [Date](#) Class.

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
```

Classes

- class [Date](#)
[Date](#) Class is used for time recording.

5.9.1 Detailed Description

Header file of [Date](#) Class. Bacpfsec Rick

Date

20150324 [Date](#) Class is used for [State](#) Class

5.10 FixtureBfcPrototype.h File Reference

CppUnit Fixture setup for [BfcPrototype](#) class.

```
#include <cppunit/TestFixture.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/ui/text/TextTestRunner.h>
#include <cppunit/ui/text/TestRunner.h>
#include <cppunit/extensions/HelperMacros.h>
#include "BfcPrototype.h"
#include <vector>
#include <iostream>
```

Classes

- class [FixtureBfcPrototype](#)

5.10.1 Detailed Description

CppUnit Fixture setup for [BfcPrototype](#) class. Bacpfsec Rick

Date

20150405 [FixtureBfcPrototype](#) is used in TestBfcPrototype for [BfcPrototype](#) Class

5.11 FixtureBfcTUI.h File Reference

CppUnit Fixture setup for [BfcTUI](#) class.

```
#include <cppunit/TestFixture.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/ui/text/TextTestRunner.h>
#include <cppunit/ui/text/TestRunner.h>
#include <cppunit/extensions/HelperMacros.h>
#include "BfcTUI.h"
#include "Task.h"
#include <vector>
#include <iostream>
```

Classes

- class [FixtureBfcTUI](#)

5.11.1 Detailed Description

CppUnit Fixture setup for [BfcTUI](#) class. Bacpfsec Rick

Date

20150405 [FixtureBfcTUI](#) is used in TestBfcTUI for [BfcTUI](#) Class

5.12 FixtureDate.h File Reference

CppUnit Fixture setup for [Date](#) class.

```
#include <cppunit/TestFixture.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/ui/text/TextTestRunner.h>
#include <cppunit/ui/text/TestRunner.h>
#include <cppunit/extensions/HelperMacros.h>
#include "Date.h"
```

Classes

- class [FixtureDate](#)

5.12.1 Detailed Description

CppUnit Fixture setup for [Date](#) class. Bacpfsec Rick

Date

20150324 [FixtureDate](#) is used in TestDate for [Date](#) Class

5.13 FixtureState.h File Reference

CppUnit Fixture setup for [State](#) class.

```
#include <cppunit/TestFixture.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/ui/text/TextTestRunner.h>
#include <cppunit/ui/text/TestRunner.h>
#include <cppunit/extensions/HelperMacros.h>
#include "State.h"
```

Classes

- class [FixtureState](#)

5.13.1 Detailed Description

CppUnit Fixture setup for [State](#) class. Bacpfsec Rick

Date

20150324 [FixtureSate](#) is used in TestSate for [State](#) Class

5.14 FixtureTask.h File Reference

CppUnit Fixture setup for [Task](#) class.

```
#include <cppunit/TestFixture.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/ui/text/TextTestRunner.h>
#include <cppunit/ui/text/TestRunner.h>
#include <cppunit/extensions/HelperMacros.h>
#include "Task.h"
#include <vector>
```

Classes

- class [FixtureTask](#)

5.14.1 Detailed Description

CppUnit Fixture setup for [Task](#) class. Bacpfsec Rick

Date

20150405 [FixtureTask](#) is used in TestDate for [Task](#) Class

5.15 State.cc File Reference

Implementation of [State](#) Class.

```
#include "State.h"
```

5.15.1 Detailed Description

Implementation of [State](#) Class. Bacpfsec Rick

Date

20150324 [State](#) Class is used for [Task](#) Class

5.16 State.h File Reference

Header file of [State](#) Class.

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include "Date.h"
```

Classes

- class [State](#)
[State](#) Class is used to record a progress of a task.

5.16.1 Detailed Description

Header file of [State](#) Class. Bacpfsec Rick

Date

20150324 [State](#) Class is used for [Task](#) Class

5.17 Task.cc File Reference

Implementation for [Task](#) Class.

```
#include "Task.h"
```

5.17.1 Detailed Description

Implementation for [Task](#) Class. Bacpfsec Rick

Date

20150405 [Task](#) Class for actual recording of one task

5.18 Task.h File Reference

Header file of [Task](#) Class.

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include "Date.h"
#include "State.h"
```

Classes

- class [Task](#)
[Task](#) Class for actual recording of tasks.

5.18.1 Detailed Description

Header file of [Task](#) Class. Bacpfsec Rick

Date

20150405 [Task](#) Class for actual recording of one task

5.19 TestDate.cc File Reference

Test program for [Date](#) Class.

```
#include "FixtureDate.h"
```

Functions

- int **main** ()

5.19.1 Detailed Description

Test program for [Date](#) Class. Bacpfsec Rick

Date

20150324 TestDate applies CppUnit test towards [Date](#) Class

5.20 TestState.cc File Reference

Test program for [State](#) Class.

```
#include "FixtureState.h"
```

Functions

- int **main** ()

5.20.1 Detailed Description

Test program for [State](#) Class. Bacpfsec Rick

Date

20150324 TestDate applies CppUnit test towards [Date](#) Class

5.21 TestTask.cc File Reference

Test program for [Task](#) Class.

```
#include "FixtureTask.h"
```

Functions

- int **main** ()

5.21.1 Detailed Description

Test program for [Task](#) Class. Bacpfsec Rick

Date

20150405 TestTask applies CppUnit test towards [Task](#) Class

Index

- ~BfcPrototype
 - BfcPrototype, 8
- ~BfcTUI
 - BfcTUI, 12
- ~Date
 - Date, 16
- ~State
 - State, 25
- ~Task
 - Task, 28
- BfcPrototype, 7
 - ~BfcPrototype, 8
 - BfcPrototype, 8
 - BfcPrototype, 8
 - endDate, 10
 - getEndDate, 8
 - getMaxTaskInProgress, 8
 - getNumOfTaskCancelled, 8
 - getNumOfTaskUncomplete, 8
 - getRecordLoaded, 8
 - getStartDate, 8
 - getTasks, 9
 - init, 9
 - maxTaskInProgress, 10
 - numOfTaskCancelled, 10
 - numOfTaskUncomplete, 10
 - readStates, 9
 - readTasks, 9
 - recordLoaded, 10
 - run, 9
 - setEndDate, 9
 - setNumOfTaskCancelled, 9
 - setNumOfTaskUncomplete, 9
 - setRecordLoaded, 9
 - setStartDate, 9
 - startDate, 10
 - tasks, 10
- BfcPrototype.cc, 31
- BfcPrototype.h, 31
- BfcTR.cc, 32
- BfcTUI, 11
 - ~BfcTUI, 12
 - BfcTUI, 12
 - BfcTUI, 12
 - briefReport, 12
 - printDay, 12
 - run, 12
 - selectAction, 12
 - selectTask, 14
 - showRecord, 14
 - timeline, 14
 - timelineHor, 14
 - timelineVer, 14
 - working, 15
 - writeRecord, 15
- BfcTUI.cc, 33
- BfcTUI.h, 33
- BfcTerminal, 10
 - instruct, 11
 - run, 11
- BfcTerminal.cc, 32
- BfcTerminal.h, 32
- briefReport
 - BfcTUI, 12
- content
 - State, 26
- Date, 15
 - ~Date, 16
 - Date, 16
 - date, 17
 - getValue, 16
 - isValid, 16
 - nextDate, 16
 - operator==, 17
 - setValue, 17
- date
 - Date, 17
 - State, 26
- Date.cc, 34
- Date.h, 34
- endDate
 - BfcPrototype, 10
- FixtureBfcPrototype, 17
 - testInitAndGetters, 18
 - testReadStates, 18
 - testReadTasks, 18
 - testSetters, 18
 - testTasksGetter, 18
- FixtureBfcPrototype.h, 34
- FixtureBfcTUI, 18
 - testBriefReport, 19
 - testPrintDay, 19
 - testSelectTask, 19
 - testShowRecord, 19
 - testTimelineHor, 19

- testTimelineVer, [20](#)
- testWriteRecord, [20](#)
- FixtureBfcTUI.h, [35](#)
- FixtureDate, [20](#)
 - testCopyConstructor, [21](#)
 - testEquivalent, [21](#)
 - testGetter, [21](#)
 - testIsValid, [21](#)
 - testNextDate, [21](#)
 - testParaConstructor, [21](#)
 - testSetter, [21](#)
- FixtureDate.h, [35](#)
- FixtureState, [21](#)
 - testContentGetter, [22](#)
 - testContentMerge, [22](#)
 - testContentSetter, [22](#)
 - testCopyConstructor, [22](#)
 - testDateGetter, [22](#)
 - testDateSetter, [22](#)
 - testParaConstructor, [22](#)
 - testStateMerge, [23](#)
- FixtureState.h, [36](#)
- FixtureTask, [23](#)
 - testCopyConstructor, [24](#)
 - testParaConstructor, [24](#)
 - testStateGetter, [24](#)
 - testStateSetter, [24](#)
 - testStatusGetter, [24](#)
 - testStatusSetter, [24](#)
 - testTaskNameGetter, [24](#)
 - testTaskNameSetter, [24](#)
- FixtureTask.h, [36](#)
- getContent
 - State, [25](#)
- getDate
 - State, [25](#)
- getEndDate
 - BfcPrototype, [8](#)
- getMaxTaskInProgress
 - BfcPrototype, [8](#)
- getNumOfTaskCancelled
 - BfcPrototype, [8](#)
- getNumOfTaskUncomplete
 - BfcPrototype, [8](#)
- getRecordLoaded
 - BfcPrototype, [8](#)
- getStartDate
 - BfcPrototype, [8](#)
- getStates
 - Task, [28](#)
- getStatus
 - Task, [28](#)
- getTaskName
 - Task, [28](#)
- getTasks
 - BfcPrototype, [9](#)
- getValue
 - Date, [16](#)
- init
 - BfcPrototype, [9](#)
- instruct
 - BfcTerminal, [11](#)
- isValid
 - Date, [16](#)
- maxTaskInProgress
 - BfcPrototype, [10](#)
- merge
 - State, [26](#)
- nextDate
 - Date, [16](#)
- numOfTaskCancelled
 - BfcPrototype, [10](#)
- numOfTaskUncomplete
 - BfcPrototype, [10](#)
- operator==
 - Date, [17](#)
- printDay
 - BfcTUI, [12](#)
- readStates
 - BfcPrototype, [9](#)
- readTasks
 - BfcPrototype, [9](#)
- recordLoaded
 - BfcPrototype, [10](#)
- run
 - BfcPrototype, [9](#)
 - BfcTerminal, [11](#)
 - BfcTUI, [12](#)
- selectAction
 - BfcTUI, [12](#)
- selectTask
 - BfcTUI, [14](#)
- setContent
 - State, [26](#)
- setDate
 - State, [26](#)
- setEndDate
 - BfcPrototype, [9](#)
- setNumOfTaskCancelled
 - BfcPrototype, [9](#)
- setNumOfTaskUncomplete
 - BfcPrototype, [9](#)
- setRecordLoaded
 - BfcPrototype, [9](#)
- setStartDate
 - BfcPrototype, [9](#)
- setStates
 - Task, [28](#)
- setStatus
 - Task, [28](#)
- setTaskName
 - Task, [29](#)

- setValue
 - Date, [17](#)
- showRecord
 - BfcTUI, [14](#)
- startDate
 - BfcPrototype, [10](#)
- State, [24](#)
 - ~State, [25](#)
 - content, [26](#)
 - date, [26](#)
 - getContent, [25](#)
 - getDate, [25](#)
 - merge, [26](#)
 - setContent, [26](#)
 - setDate, [26](#)
 - State, [25](#)
- State.cc, [37](#)
- State.h, [37](#)
- states
 - Task, [29](#)
- Task, [27](#)
 - ~Task, [28](#)
 - getStates, [28](#)
 - getStatus, [28](#)
 - getTaskName, [28](#)
 - setStates, [28](#)
 - setStatus, [28](#)
 - setTaskName, [29](#)
 - states, [29](#)
 - Task, [27](#), [28](#)
 - taskName, [29](#)
- Task.cc, [37](#)
- Task.h, [38](#)
- taskName
 - Task, [29](#)
- tasks
 - BfcPrototype, [10](#)
- testBriefReport
 - FixtureBfcTUI, [19](#)
- testContentGetter
 - FixtureState, [22](#)
- testContentMerge
 - FixtureState, [22](#)
- testContentSetter
 - FixtureState, [22](#)
- testCopyConstructor
 - FixtureDate, [21](#)
 - FixtureState, [22](#)
 - FixtureTask, [24](#)
- TestDate.cc, [38](#)
- testDateGetter
 - FixtureState, [22](#)
- testDateSetter
 - FixtureState, [22](#)
- testEquivalent
 - FixtureDate, [21](#)
- testGetter
 - FixtureDate, [21](#)
- testInitAndGetters
 - FixtureBfcPrototype, [18](#)
- testIsValid
 - FixtureDate, [21](#)
- testNextDate
 - FixtureDate, [21](#)
- testParaConstructor
 - FixtureDate, [21](#)
 - FixtureState, [22](#)
 - FixtureTask, [24](#)
- testPrintDay
 - FixtureBfcTUI, [19](#)
- testReadStates
 - FixtureBfcPrototype, [18](#)
- testReadTasks
 - FixtureBfcPrototype, [18](#)
- testSelectTask
 - FixtureBfcTUI, [19](#)
- testSetter
 - FixtureDate, [21](#)
- testSetters
 - FixtureBfcPrototype, [18](#)
- testShowRecord
 - FixtureBfcTUI, [19](#)
- TestState.cc, [38](#)
- testStateGetter
 - FixtureTask, [24](#)
- testStateMerge
 - FixtureState, [23](#)
- testStateSetter
 - FixtureTask, [24](#)
- testStatusGetter
 - FixtureTask, [24](#)
- testStatusSetter
 - FixtureTask, [24](#)
- TestTask.cc, [39](#)
- testTaskNameGetter
 - FixtureTask, [24](#)
- testTaskNameSetter
 - FixtureTask, [24](#)
- testTasksGetter
 - FixtureBfcPrototype, [18](#)
- testTimelineHor
 - FixtureBfcTUI, [19](#)
- testTimelineVer
 - FixtureBfcTUI, [20](#)
- testWriteRecord
 - FixtureBfcTUI, [20](#)
- timeline
 - BfcTUI, [14](#)
- timelineHor
 - BfcTUI, [14](#)
- timelineVer
 - BfcTUI, [14](#)
- working
 - BfcTUI, [15](#)
- writeRecord
 - BfcTUI, [15](#)