

SmartOS

0.3.3

Generated by Doxygen 1.8.14



# Contents

<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>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	BlockedQueueWidget Class Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Constructor & Destructor Documentation . . . . .	7
4.1.2.1	BlockedQueueWidget() . . . . .	7
4.1.3	Member Function Documentation . . . . .	8
4.1.3.1	update() . . . . .	8
4.2	CentralProcessingUnit Class Reference . . . . .	8
4.2.1	Detailed Description . . . . .	8
4.2.2	Constructor & Destructor Documentation . . . . .	8
4.2.2.1	CentralProcessingUnit() . . . . .	8
4.2.3	Member Function Documentation . . . . .	9
4.2.3.1	currentProcess() . . . . .	9
4.2.3.2	setActiveProcess() . . . . .	9
4.3	CpuWidget Class Reference . . . . .	9
4.3.1	Detailed Description . . . . .	10

4.3.2	Constructor & Destructor Documentation . . . . .	10
4.3.2.1	CpuWidget() . . . . .	10
4.3.3	Member Function Documentation . . . . .	10
4.3.3.1	update() . . . . .	10
4.4	HelpDialog Class Reference . . . . .	10
4.4.1	Detailed Description . . . . .	11
4.4.2	Constructor & Destructor Documentation . . . . .	11
4.4.2.1	HelpDialog() . . . . .	11
4.5	HistoryDialog Class Reference . . . . .	11
4.5.1	Detailed Description . . . . .	11
4.5.2	Constructor & Destructor Documentation . . . . .	12
4.5.2.1	HistoryDialog() . . . . .	12
4.6	IOEvent Class Reference . . . . .	12
4.6.1	Detailed Description . . . . .	12
4.6.2	Member Enumeration Documentation . . . . .	12
4.6.2.1	Type . . . . .	12
4.6.3	Constructor & Destructor Documentation . . . . .	13
4.6.3.1	IOEvent() . . . . .	13
4.6.4	Member Function Documentation . . . . .	13
4.6.4.1	cycleStamp() . . . . .	13
4.6.4.2	type() . . . . .	14
4.7	MainWindow Class Reference . . . . .	14
4.7.1	Detailed Description . . . . .	14
4.7.2	Constructor & Destructor Documentation . . . . .	15
4.7.2.1	MainWindow() . . . . .	15
4.7.2.2	~MainWindow() . . . . .	15
4.7.3	Member Function Documentation . . . . .	15
4.7.3.1	addHistory() . . . . .	15
4.7.3.2	showProcessScheduler . . . . .	15
4.7.3.3	showUserDocumentation . . . . .	16

4.8	ProcessControlBlock Class Reference	16
4.8.1	Detailed Description	17
4.8.2	Constructor & Destructor Documentation	17
4.8.2.1	ProcessControlBlock()	17
4.8.3	Member Function Documentation	17
4.8.3.1	clearWaitEvent()	17
4.8.3.2	cpuUsageTerm()	17
4.8.3.3	ioEvent()	18
4.8.3.4	ioReqTerm()	18
4.8.3.5	memory()	18
4.8.3.6	pid()	18
4.8.3.7	priority()	19
4.8.3.8	processType()	19
4.8.3.9	setPriority()	19
4.8.3.10	setProcessType()	19
4.8.3.11	setWaitEvent()	20
4.8.3.12	updateCpuUsageTerm()	20
4.8.3.13	updateIoReqTerm()	20
4.8.3.14	updateWaitTerm()	21
4.8.3.15	waitTerm()	21
4.9	ProcessCreationDialog Class Reference	21
4.9.1	Detailed Description	22
4.9.2	Constructor & Destructor Documentation	22
4.9.2.1	ProcessCreationDialog()	22
4.9.3	Member Function Documentation	22
4.9.3.1	memoryRequired()	22
4.9.3.2	pid()	22
4.9.3.3	processType()	23
4.10	ProcessSchedulerWidget Class Reference	23
4.10.1	Detailed Description	23

4.10.2	Constructor & Destructor Documentation . . . . .	23
4.10.2.1	ProcessSchedulerWidget() . . . . .	24
4.10.3	Member Function Documentation . . . . .	24
4.10.3.1	addProcessControlBlock . . . . .	24
4.10.3.2	addRandomProcessControlBlocks . . . . .	24
4.10.3.3	execute . . . . .	24
4.10.3.4	executeStep . . . . .	24
4.10.3.5	paintEvent() . . . . .	25
4.10.3.6	pause . . . . .	25
4.10.3.7	stop . . . . .	25
4.11	ProcessWidget Class Reference . . . . .	25
4.11.1	Detailed Description . . . . .	26
4.11.2	Constructor & Destructor Documentation . . . . .	26
4.11.2.1	ProcessWidget() . . . . .	26
4.11.3	Member Function Documentation . . . . .	26
4.11.3.1	paintEvent() . . . . .	26
4.11.3.2	update() . . . . .	26
4.12	ReadyQueueWidget Class Reference . . . . .	26
4.12.1	Detailed Description . . . . .	27
4.12.2	Constructor & Destructor Documentation . . . . .	27
4.12.2.1	ReadyQueueWidget() . . . . .	27
4.12.3	Member Function Documentation . . . . .	27
4.12.3.1	update() . . . . .	27
4.13	SchedulingDialog Class Reference . . . . .	27
4.13.1	Detailed Description . . . . .	28
4.13.2	Constructor & Destructor Documentation . . . . .	28
4.13.2.1	SchedulingDialog() . . . . .	28
4.13.3	Member Function Documentation . . . . .	28
4.13.3.1	priorityQueues() . . . . .	28
4.13.3.2	quantum() . . . . .	28

4.13.3.3 schedulerType()	29
4.14 SmartOS Class Reference	29
4.14.1 Detailed Description	30
4.14.2 Constructor & Destructor Documentation	30
4.14.2.1 SmartOS()	30
4.14.3 Member Function Documentation	31
4.14.3.1 addOperatingSystemProcess()	31
4.14.3.2 blockedQueue()	31
4.14.3.3 blockProcessControlBlock()	31
4.14.3.4 cpu()	32
4.14.3.5 createProcessControlBlock()	32
4.14.3.6 cycleCount()	32
4.14.3.7 deleteProcessControlBlock()	32
4.14.3.8 determineNextProcess()	33
4.14.3.9 execute()	33
4.14.3.10 findProcessControlBlock()	33
4.14.3.11 getVersionNumber()	34
4.14.3.12 ioEventQueue()	34
4.14.3.13 maxMemory()	34
4.14.3.14 nextSequentialPID()	35
4.14.3.15 readyQueue()	35
4.14.3.16 reset()	35
4.14.3.17 setActiveProcess()	35
4.14.3.18 setMaximumPriority()	36
4.14.3.19 setScheduler()	36
4.14.3.20 setTimeQuantum()	36
4.14.3.21 unblockProcessControlBlock()	37
4.14.3.22 updateCurrentProcessControlBlock()	37
4.14.3.23 usedMemory()	37
4.15 WelcomeWidget Class Reference	38
4.15.1 Detailed Description	38
4.15.2 Constructor & Destructor Documentation	38
4.15.2.1 WelcomeWidget()	38

<b>5</b>	<b>File Documentation</b>	<b>39</b>
5.1	/home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.cpp File Reference . . .	39
5.2	/home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.h File Reference . . . . .	39
5.3	/home/nicholas/qtcreator/smartos/SmartOS_Core/Global.h File Reference . . . . .	39
5.3.1	Macro Definition Documentation . . . . .	39
5.3.1.1	SMARTOS_CORESHARED_EXPORT . . . . .	40
5.4	/home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.cpp File Reference . . . . .	40
5.5	/home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.h File Reference . . . . .	40
5.6	/home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.cpp File Reference . . . . .	40
5.7	/home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.h File Reference . . . . .	40
5.7.1	Typedef Documentation . . . . .	41
5.7.1.1	ProcessControlBlockPtr . . . . .	41
5.7.2	Enumeration Type Documentation . . . . .	41
5.7.2.1	ProcessType . . . . .	41
5.8	/home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.cpp File Reference . . . . .	41
5.9	/home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.h File Reference . . . . .	42
5.9.1	Typedef Documentation . . . . .	42
5.9.1.1	IOEventQueue . . . . .	42
5.9.1.2	PCBQueue . . . . .	42
5.9.2	Enumeration Type Documentation . . . . .	43
5.9.2.1	SchedulerType . . . . .	43
5.10	/home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.cpp File Reference . . . . .	43
5.11	/home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.h File Reference . . . . .	43
5.12	/home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.cpp File Reference . . . . .	44
5.13	/home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.h File Reference . . . . .	44
5.14	/home/nicholas/qtcreator/smartos/SmartOS_GUI/Globals.h File Reference . . . . .	44
5.14.1	Variable Documentation . . . . .	44
5.14.1.1	g_SmartOS . . . . .	44
5.15	/home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.cpp File Reference . . . . .	45
5.16	/home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.h File Reference . . . . .	45



5.17	/home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.cpp File Reference . . . . .	45
5.18	/home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.h File Reference . . . . .	45
5.19	/home/nicholas/qtcreator/smartos/SmartOS_GUI/main.cpp File Reference . . . . .	45
5.19.1	Function Documentation . . . . .	46
5.19.1.1	main() . . . . .	46
5.19.2	Variable Documentation . . . . .	46
5.19.2.1	g_SmartOS . . . . .	46
5.19.2.2	MEMORY . . . . .	46
5.20	/home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.cpp File Reference . . . . .	46
5.21	/home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.h File Reference . . . . .	47
5.22	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.cpp File Reference . . . . .	47
5.23	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.h File Reference . . . . .	47
5.24	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.cpp File Reference . . . . .	48
5.24.1	Variable Documentation . . . . .	48
5.24.1.1	g_SmartOS . . . . .	48
5.25	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.h File Reference . . . . .	48
5.26	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.cpp File Reference . . . . .	49
5.27	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.h File Refer- ence . . . . .	49
5.28	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.cpp File Reference . . . . .	49
5.29	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.h File Reference . . . . .	49
5.30	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.cpp File Reference . . . . .	50
5.31	/home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.h File Reference . . . . .	50
5.32	/home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.cpp File Reference . . . . .	50
5.33	/home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.h File Reference . . . . .	50



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

CentralProcessingUnit . . . . .	8
IOEvent . . . . .	12
ProcessControlBlock . . . . .	16
QDialog	
HelpDialog . . . . .	10
HistoryDialog . . . . .	11
ProcessCreationDialog . . . . .	21
SchedulingDialog . . . . .	27
QMainWindow	
MainWindow . . . . .	14
QWidget	
BlockedQueueWidget . . . . .	7
CpuWidget . . . . .	9
ProcessSchedulerWidget . . . . .	23
ProcessWidget . . . . .	25
ReadyQueueWidget . . . . .	26
WelcomeWidget . . . . .	38
SmartOS . . . . .	29



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">BlockedQueueWidget</a> . . . . .	7
<a href="#">CentralProcessingUnit</a>	
Holds the current process . . . . .	8
<a href="#">CpuWidget</a> . . . . .	9
<a href="#">HelpDialog</a> . . . . .	10
<a href="#">HistoryDialog</a> . . . . .	11
<a href="#">IOEvent</a> . . . . .	12
<a href="#">MainWindow</a> . . . . .	14
<a href="#">ProcessControlBlock</a>	
Process Control BLock in the simulator . . . . .	16
<a href="#">ProcessCreationDialog</a> . . . . .	21
<a href="#">ProcessSchedulerWidget</a> . . . . .	23
<a href="#">ProcessWidget</a> . . . . .	25
<a href="#">ReadyQueueWidget</a> . . . . .	26
<a href="#">SchedulingDialog</a> . . . . .	27
<a href="#">SmartOS</a>	
Heart of the <a href="#">SmartOS</a> operating system simulator. All functions to manipulate processes are found here . . . . .	29
<a href="#">WelcomeWidget</a> . . . . .	38



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

/home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.cpp	39
/home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.h	39
/home/nicholas/qtcreator/smartos/SmartOS_Core/Global.h	39
/home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.cpp	40
/home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.h	40
/home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.cpp	40
/home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.h	40
/home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.cpp	41
/home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.h	42
/home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.cpp	43
/home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.h	43
/home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.cpp	44
/home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.h	44
/home/nicholas/qtcreator/smartos/SmartOS_GUI/Globals.h	44
/home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.cpp	45
/home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.h	45
/home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.cpp	45
/home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.h	45
/home/nicholas/qtcreator/smartos/SmartOS_GUI/main.cpp	45
/home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.cpp	46
/home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.h	47
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.cpp	49
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.h	49
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.cpp	50
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.h	50
/home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.cpp	50
/home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.h	50
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.cpp	47
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.h	47
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.cpp	48
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.h	48
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.cpp	49
/home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.h	49





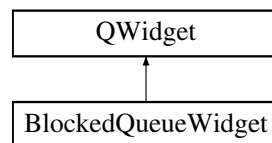
## Chapter 4

# Class Documentation

### 4.1 BlockedQueueWidget Class Reference

```
#include <BlockedQueueWidget.h>
```

Inheritance diagram for BlockedQueueWidget:



#### Public Member Functions

- [BlockedQueueWidget](#) (`QWidget *parent=nullptr`)
- void [update](#) ()

#### 4.1.1 Detailed Description

Definition at line 7 of file `BlockedQueueWidget.h`.

#### 4.1.2 Constructor & Destructor Documentation

##### 4.1.2.1 BlockedQueueWidget()

```
BlockedQueueWidget::BlockedQueueWidget (
    QWidget * parent = nullptr )
```

Definition at line 13 of file `BlockedQueueWidget.cpp`.

### 4.1.3 Member Function Documentation

#### 4.1.3.1 update()

```
void BlockedQueueWidget::update ( )
```

Definition at line 29 of file BlockedQueueWidget.cpp.

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

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/BlockedQueueWidget.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/BlockedQueueWidget.cpp](#)

## 4.2 CentralProcessingUnit Class Reference

The [CentralProcessingUnit](#) class holds the current process.

```
#include <CentralProcessingUnit.h>
```

### Public Member Functions

- [CentralProcessingUnit \(\)](#)  
*CentralProcessingUnit construct a new Central Processing Unit.*
- [ProcessControlBlockPtr setActiveProcess \(ProcessControlBlockPtr pcb\)](#)  
*setActiveProcess sets the active process*
- [ProcessControlBlockPtr & currentProcess \(\)](#)  
*currentProcess returns a reference to the current process.*

#### 4.2.1 Detailed Description

The [CentralProcessingUnit](#) class holds the current process.

Definition at line 9 of file CentralProcessingUnit.h.

#### 4.2.2 Constructor & Destructor Documentation

##### 4.2.2.1 CentralProcessingUnit()

```
CentralProcessingUnit::CentralProcessingUnit ( )
```

[CentralProcessingUnit](#) construct a new Central Processing Unit.

Definition at line 3 of file CentralProcessingUnit.cpp.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 currentProcess()

```
ProcessControlBlockPtr & CentralProcessingUnit::currentProcess ( )
```

currentProcess returns a reference to the current process.

##### Returns

a reference to the current process

Definition at line 13 of file CentralProcessingUnit.cpp.

#### 4.2.3.2 setActiveProcess()

```
ProcessControlBlockPtr CentralProcessingUnit::setActiveProcess (
    ProcessControlBlockPtr pcb )
```

setActiveProcess sets the active process

##### Parameters

<i>pcb</i>	the process to make active
------------	----------------------------

##### Returns

the old process that was in the CPU, or nullptr

Definition at line 6 of file CentralProcessingUnit.cpp.

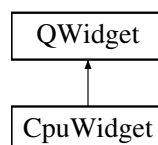
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS\_Core/[CentralProcessingUnit.h](#)
- /home/nicholas/qtcreator/smartos/SmartOS\_Core/[CentralProcessingUnit.cpp](#)

## 4.3 CpuWidget Class Reference

```
#include <CpuWidget.h>
```

Inheritance diagram for CpuWidget:



## Public Member Functions

- [CpuWidget](#) (QWidget \*parent=nullptr)
- void [update](#) ()

### 4.3.1 Detailed Description

Definition at line 13 of file CpuWidget.h.

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 CpuWidget()

```
CpuWidget::CpuWidget (
    QWidget * parent = nullptr ) [explicit]
```

Definition at line 11 of file CpuWidget.cpp.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 update()

```
void CpuWidget::update ( )
```

Definition at line 24 of file CpuWidget.cpp.

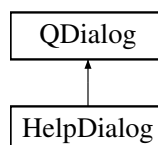
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/CpuWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/CpuWidget.cpp

## 4.4 HelpDialog Class Reference

```
#include <HelpDialog.h>
```

Inheritance diagram for HelpDialog:



## Public Member Functions

- [HelpDialog](#) ()

### 4.4.1 Detailed Description

Definition at line 6 of file HelpDialog.h.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 HelpDialog()

```
HelpDialog::HelpDialog ( )
```

Definition at line 3 of file HelpDialog.cpp.

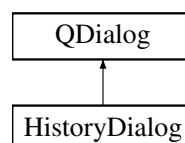
The documentation for this class was generated from the following files:

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/HelpDialog.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/HelpDialog.cpp](#)

## 4.5 HistoryDialog Class Reference

```
#include <HistoryDialog.h>
```

Inheritance diagram for HistoryDialog:



## Public Member Functions

- [HistoryDialog](#) (QStringList &historyList)

### 4.5.1 Detailed Description

Definition at line 7 of file HistoryDialog.h.

## 4.5.2 Constructor & Destructor Documentation

### 4.5.2.1 HistoryDialog()

```
HistoryDialog::HistoryDialog (
    QStringList & historyList )
```

Definition at line 7 of file HistoryDialog.cpp.

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

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/HistoryDialog.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/HistoryDialog.cpp](#)

## 4.6 IOEvent Class Reference

```
#include <IOEvent.h>
```

### Public Types

- enum [Type](#) { [Type::NONE](#), [Type::HARD\\_DRIVE](#), [Type::USER\\_IO](#) }
- The Type enum represents the different types of IOEvents.*

### Public Member Functions

- [IOEvent](#) ([Type](#) type, size\_t cycleStamp)
- *[IOEvent](#) constructs a new [IOEvent](#) with the specified type and cycleStamp.*
- [Type](#) type () const
- *type returns the type of event*
- size\_t [cycleStamp](#) () const
- *cycleStamp returns the time cycle this event was added.*

### 4.6.1 Detailed Description

Definition at line 6 of file IOEvent.h.

### 4.6.2 Member Enumeration Documentation

#### 4.6.2.1 Type

```
enum IOEvent::Type [strong]
```

The Type enum represents the different types of IOEvents.

## Enumerator

NONE	
HARD_DRIVE	
USER_IO	

Definition at line 12 of file IOEvent.h.

### 4.6.3 Constructor & Destructor Documentation

#### 4.6.3.1 IOEvent()

```
IOEvent::IOEvent (
    Type type,
    size_t cycleStamp )
```

[IOEvent](#) constructs a new [IOEvent](#) with the specified type and cycleStamp.

## Parameters

<i>type</i>	the type of <a href="#">IOEvent</a>
<i>cycleStamp</i>	the time cycle this event was added

Definition at line 3 of file IOEvent.cpp.

### 4.6.4 Member Function Documentation

#### 4.6.4.1 cycleStamp()

```
size_t IOEvent::cycleStamp ( ) const
```

`cycleStamp` returns the time cycle this event was added.

## Returns

the time cycle this event was added

Definition at line 13 of file IOEvent.cpp.

#### 4.6.4.2 type()

```
IOEvent::Type IOEvent::type ( ) const
```

type returns the type of event

##### Returns

the type of event

Definition at line 8 of file IOEvent.cpp.

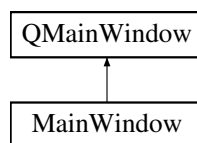
The documentation for this class was generated from the following files:

- [/home/nicholas/qtcreator/smartos/SmartOS\\_Core/IOEvent.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_Core/IOEvent.cpp](#)

## 4.7 MainWindow Class Reference

```
#include <MainWindow.h>
```

Inheritance diagram for MainWindow:



### Public Slots

- void [showProcessScheduler](#) ()  
*showProcessScheduler shows the view to deal with process scheduling.*
- void [showUserDocumentation](#) ()  
*showUserDocumentation shows the user documentation PDF.*

### Public Member Functions

- [MainWindow](#) (QWidget \*parent=0)
- [~MainWindow](#) ()
- void [addHistory](#) (const QString &info)  
*addHistory add the following information to the history*

#### 4.7.1 Detailed Description

Definition at line 10 of file MainWindow.h.



## 4.7.2 Constructor & Destructor Documentation

### 4.7.2.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = 0 ) [explicit]
```

Definition at line 24 of file MainWindow.cpp.

### 4.7.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

Definition at line 37 of file MainWindow.cpp.

## 4.7.3 Member Function Documentation

### 4.7.3.1 addHistory()

```
void MainWindow::addHistory (
    const QString & info )
```

addHistory add the following information to the history

#### Parameters

<i>info</i>	the info to add to the history
-------------	--------------------------------

Definition at line 40 of file MainWindow.cpp.

### 4.7.3.2 showProcessScheduler

```
void MainWindow::showProcessScheduler ( ) [slot]
```

showProcessScheduler shows the view to deal with process scheduling.

Definition at line 113 of file MainWindow.cpp.

#### 4.7.3.3 showUserDocumentation

```
void MainWindow::showUserDocumentation ( ) [slot]
```

showUserDocumentation shows the user documentation PDF.

Definition at line 132 of file MainWindow.cpp.

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

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/MainWindow.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/MainWindow.cpp](#)

## 4.8 ProcessControlBlock Class Reference

The [ProcessControlBlock](#) class represents a Process Control BLock in the simulator.

```
#include <ProcessControlBlock.h>
```

### Public Member Functions

- [ProcessControlBlock](#) (size\_t [pid](#), size\_t [memory](#))  
*ProcessControlBlock constructs a new Process Control Block with the specified pid and memory required.*
- size\_t [pid](#) () const  
*pid returns the process's id*
- size\_t [cpuUsageTerm](#) () const  
*cpuUsageTerm returns the time spent using the cpu*
- size\_t [ioReqTerm](#) () const  
*ioReqTerm returns the time waiting for IO*
- size\_t [waitTerm](#) () const  
*waitTerm returns the time waiting in the ready queue*
- size\_t [memory](#) () const  
*memory returns the memory required by the process*
- size\_t [priority](#) () const  
*priority returns the priority for this process*
- [IOEvent](#) [ioEvent](#) () const  
*ioEvent returns the type of IOEvent that the process is waiting on.*
- [ProcessType](#) [processType](#) () const  
*processType returns the type of process this is*
- void [updateCpuUsageTerm](#) (size\_t elapsed)  
*updateCpuUsageTerm adds the elapsed time to the cpu usage.*
- void [updateIoReqTerm](#) (size\_t elapsed)  
*updateIoReqTerm adds the elapsed time to the io waiting time*
- void [updateWaitTerm](#) (size\_t elapsed)  
*updateWaitTerm adds the elapsed time to the time waiting in the ready queue.*
- void [setWaitEvent](#) ([IOEvent](#) event)  
*setWaitEvent sets the type of event this process is waiting for*
- void [clearWaitEvent](#) ()  
*clearWaitEvent clears the event this process is waiting for*
- void [setProcessType](#) ([ProcessType](#) type)  
*setProcessType sets the type of process this is*
- void [setPriority](#) (size\_t [priority](#))  
*setPriority sets this process's priority*

### 4.8.1 Detailed Description

The [ProcessControlBlock](#) class represents a Process Control BLock in the simulator.

Definition at line 16 of file ProcessControlBlock.h.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 ProcessControlBlock()

```
ProcessControlBlock::ProcessControlBlock (
    size_t pid,
    size_t memory ) [explicit]
```

[ProcessControlBlock](#) constructs a new Process Control Block with the specified pid and memory required.

##### Parameters

<i>pid</i>	the id of the process
<i>memory</i>	the amount of memory needed by the process

Definition at line 3 of file ProcessControlBlock.cpp.

### 4.8.3 Member Function Documentation

#### 4.8.3.1 clearWaitEvent()

```
void ProcessControlBlock::clearWaitEvent ( )
```

clearWaitEvent clears the event this process is waiting for

Definition at line 74 of file ProcessControlBlock.cpp.

#### 4.8.3.2 cpuUsageTerm()

```
size_t ProcessControlBlock::cpuUsageTerm ( ) const
```

cpuUsageTerm returns the time spent using the cpu

##### Returns

the time spent using the cpu

Definition at line 19 of file ProcessControlBlock.cpp.

#### 4.8.3.3 ioEvent()

```
IOEvent ProcessControlBlock::ioEvent ( ) const
```

ioEvent returns the type of [IOEvent](#) that the process is waiting on.

##### Returns

the type of [IOEvent](#) the process is waiting on

Definition at line 44 of file ProcessControlBlock.cpp.

#### 4.8.3.4 ioReqTerm()

```
size_t ProcessControlBlock::ioReqTerm ( ) const
```

ioReqTerm returns the time waiting for IO

##### Returns

the time waiting for IO

Definition at line 24 of file ProcessControlBlock.cpp.

#### 4.8.3.5 memory()

```
size_t ProcessControlBlock::memory ( ) const
```

memory returns the memory required by the process

##### Returns

the memory required by the process

Definition at line 34 of file ProcessControlBlock.cpp.

#### 4.8.3.6 pid()

```
size_t ProcessControlBlock::pid ( ) const
```

pid returns the process's id

##### Returns

the process's id

Definition at line 14 of file ProcessControlBlock.cpp.

#### 4.8.3.7 priority()

```
size_t ProcessControlBlock::priority ( ) const
```

priority returns the priority for this process

##### Returns

the priority of the process

Definition at line 39 of file ProcessControlBlock.cpp.

#### 4.8.3.8 processType()

```
ProcessType ProcessControlBlock::processType ( ) const
```

processType returns the type of process this is

##### Returns

the type of process this is

Definition at line 49 of file ProcessControlBlock.cpp.

#### 4.8.3.9 setPriority()

```
void ProcessControlBlock::setPriority (
    size_t priority )
```

setPriority sets this process's priority

##### Parameters

<i>priority</i>	the new priority
-----------------	------------------

Definition at line 84 of file ProcessControlBlock.cpp.

#### 4.8.3.10 setProcessType()

```
void ProcessControlBlock::setProcessType (
    ProcessType type )
```

setProcessType sets the type of process this is

**Parameters**

<i>type</i>	the type of process this is
-------------	-----------------------------

Definition at line 79 of file ProcessControlBlock.cpp.

**4.8.3.11 setWaitEvent()**

```
void ProcessControlBlock::setWaitEvent (
    IOEvent event )
```

setWaitEvent sets the type of event this process is waiting for

**Parameters**

<i>event</i>	the type of event this process is waiting for
--------------	---

Definition at line 69 of file ProcessControlBlock.cpp.

**4.8.3.12 updateCpuUsageTerm()**

```
void ProcessControlBlock::updateCpuUsageTerm (
    size_t elapsed )
```

updateCpuUsageTerm adds the elapsed time to the cpu usage.

**Parameters**

<i>elapsed</i>	the time elapsed
----------------	------------------

Definition at line 54 of file ProcessControlBlock.cpp.

**4.8.3.13 updateIoReqTerm()**

```
void ProcessControlBlock::updateIoReqTerm (
    size_t elapsed )
```

updateIoReqTerm adds the elapsed time to the io waiting time

**Parameters**

<i>elapsed</i>	the time elapsed
----------------	------------------

Definition at line 59 of file ProcessControlBlock.cpp.

#### 4.8.3.14 updateWaitTerm()

```
void ProcessControlBlock::updateWaitTerm (
    size_t elapsed )
```

updateWaitTerm adds the elapsed time to the time waiting in the ready queue.

##### Parameters

<i>elapsed</i>	the time elapsed
----------------	------------------

Definition at line 64 of file ProcessControlBlock.cpp.

#### 4.8.3.15 waitTerm()

```
size_t ProcessControlBlock::waitTerm ( ) const
```

waitTerm returns the time waiting in the ready queue

##### Returns

the time waiting in the ready queue

Definition at line 29 of file ProcessControlBlock.cpp.

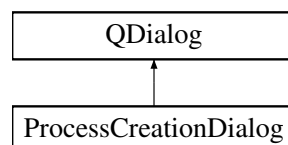
The documentation for this class was generated from the following files:

- [/home/nicholas/qtcreator/smartos/SmartOS\\_Core/ProcessControlBlock.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_Core/ProcessControlBlock.cpp](#)

## 4.9 ProcessCreationDialog Class Reference

```
#include <ProcessCreationDialog.h>
```

Inheritance diagram for ProcessCreationDialog:



## Public Member Functions

- [ProcessCreationDialog](#) ()
- `size_t` [pid](#) () const
- `size_t` [memoryRequired](#) () const
- `ProcessType` [processType](#) () const

### 4.9.1 Detailed Description

Definition at line 10 of file `ProcessCreationDialog.h`.

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 `ProcessCreationDialog()`

```
ProcessCreationDialog::ProcessCreationDialog ( )
```

Definition at line 10 of file `ProcessCreationDialog.cpp`.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 `memoryRequired()`

```
size_t ProcessCreationDialog::memoryRequired ( ) const
```

Definition at line 21 of file `ProcessCreationDialog.cpp`.

#### 4.9.3.2 `pid()`

```
size_t ProcessCreationDialog::pid ( ) const
```

Definition at line 16 of file `ProcessCreationDialog.cpp`.



#### 4.9.3.3 processType()

```
ProcessType ProcessCreationDialog::processType ( ) const
```

Definition at line 26 of file ProcessCreationDialog.cpp.

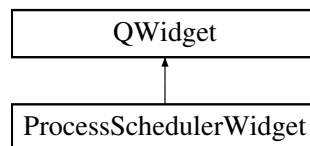
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessCreationDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessCreationDialog.cpp

## 4.10 ProcessSchedulerWidget Class Reference

```
#include <ProcessSchedulerWidget.h>
```

Inheritance diagram for ProcessSchedulerWidget:



### Public Slots

- void [addProcessControlBlock](#) ()
- void [addRandomProcessControlBlocks](#) ()
- void [execute](#) ()
- void [pause](#) ()
- void [stop](#) ()
- void [executeStep](#) ()

### Public Member Functions

- [ProcessSchedulerWidget](#) ([MainWindow](#) \*parent=nullptr)
- void [paintEvent](#) (QPaintEvent \*paintEvent) override

#### 4.10.1 Detailed Description

Definition at line 17 of file ProcessSchedulerWidget.h.

#### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 ProcessSchedulerWidget()

```
ProcessSchedulerWidget::ProcessSchedulerWidget (  
    MainWindow * parent = nullptr ) [explicit]
```

Definition at line 36 of file ProcessSchedulerWidget.cpp.

### 4.10.3 Member Function Documentation

#### 4.10.3.1 addProcessControlBlock

```
void ProcessSchedulerWidget::addProcessControlBlock ( ) [slot]
```

Definition at line 115 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.2 addRandomProcessControlBlocks

```
void ProcessSchedulerWidget::addRandomProcessControlBlocks ( ) [slot]
```

Definition at line 130 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.3 execute

```
void ProcessSchedulerWidget::execute ( ) [slot]
```

Definition at line 163 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.4 executeStep

```
void ProcessSchedulerWidget::executeStep ( ) [slot]
```

Definition at line 206 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.5 paintEvent()

```
void ProcessSchedulerWidget::paintEvent (
    QPaintEvent * paintEvent ) [override]
```

Definition at line 110 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.6 pause

```
void ProcessSchedulerWidget::pause ( ) [slot]
```

Definition at line 177 of file ProcessSchedulerWidget.cpp.

#### 4.10.3.7 stop

```
void ProcessSchedulerWidget::stop ( ) [slot]
```

Definition at line 190 of file ProcessSchedulerWidget.cpp.

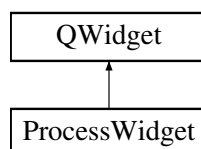
The documentation for this class was generated from the following files:

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/ProcessScheduler/ProcessSchedulerWidget.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/ProcessScheduler/ProcessSchedulerWidget.cpp](#)

## 4.11 ProcessWidget Class Reference

```
#include <ProcessWidget.h>
```

Inheritance diagram for ProcessWidget:



### Public Member Functions

- [ProcessWidget](#) ([ProcessControlBlockPtr](#) &pcb, bool detailed=false, `QWidget` \*parent=nullptr)
- void [paintEvent](#) (`QPaintEvent` \*event) override
- void [update](#) ()

### 4.11.1 Detailed Description

Definition at line 9 of file ProcessWidget.h.

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 ProcessWidget()

```
ProcessWidget::ProcessWidget (
    ProcessControlBlockPtr & pcb,
    bool detailed = false,
    QWidget * parent = nullptr )
```

Definition at line 9 of file ProcessWidget.cpp.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 paintEvent()

```
void ProcessWidget::paintEvent (
    QPaintEvent * event ) [override]
```

Definition at line 69 of file ProcessWidget.cpp.

#### 4.11.3.2 update()

```
void ProcessWidget::update ( )
```

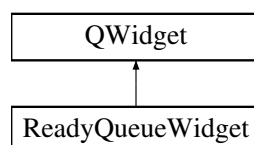
The documentation for this class was generated from the following files:

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/ProcessWidget.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/ProcessWidget.cpp](#)

## 4.12 ReadyQueueWidget Class Reference

```
#include <ReadyQueueWidget.h>
```

Inheritance diagram for ReadyQueueWidget:



## Public Member Functions

- [ReadyQueueWidget](#) (QWidget \*parent=nullptr)
- void [update](#) ()

### 4.12.1 Detailed Description

Definition at line 7 of file ReadyQueueWidget.h.

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 ReadyQueueWidget()

```
ReadyQueueWidget::ReadyQueueWidget (
    QWidget * parent = nullptr )
```

Definition at line 13 of file ReadyQueueWidget.cpp.

### 4.12.3 Member Function Documentation

#### 4.12.3.1 update()

```
void ReadyQueueWidget::update ( )
```

Definition at line 29 of file ReadyQueueWidget.cpp.

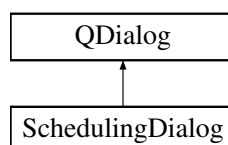
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ReadyQueueWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ReadyQueueWidget.cpp

## 4.13 SchedulingDialog Class Reference

```
#include <SchedulingDialog.h>
```

Inheritance diagram for SchedulingDialog:



## Public Member Functions

- [SchedulingDialog](#) ()
- [SchedulerType schedulerType](#) () const
- [size\\_t quantum](#) () const
- [size\\_t priorityQueues](#) () const

### 4.13.1 Detailed Description

Definition at line 10 of file SchedulingDialog.h.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 SchedulingDialog()

```
SchedulingDialog::SchedulingDialog ( )
```

Definition at line 8 of file SchedulingDialog.cpp.

### 4.13.3 Member Function Documentation

#### 4.13.3.1 priorityQueues()

```
size_t SchedulingDialog::priorityQueues ( ) const
```

Definition at line 62 of file SchedulingDialog.cpp.

#### 4.13.3.2 quantum()

```
size_t SchedulingDialog::quantum ( ) const
```

Definition at line 57 of file SchedulingDialog.cpp.

## 4.13.3.3 schedulerType()

```
SchedulerType SchedulingDialog::schedulerType ( ) const
```

Definition at line 46 of file SchedulingDialog.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/SchedulingDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/SchedulingDialog.cpp

## 4.14 SmartOS Class Reference

The [SmartOS](#) class is the heart of the [SmartOS](#) operating system simulator. All functions to manipulate processes are found here.

```
#include <SmartOS.h>
```

### Public Member Functions

- [SmartOS](#) (size\_t memory)  
*SmartOS* construct a new simulator with the specified maximum memory.
- size\_t [nextSequentialPID](#) ()  
*nextSequentialPID* the next process id that is available starting from 0.
- void [createProcessControlBlock](#) (size\_t pid, size\_t memory, [ProcessType](#) type=[ProcessType::RANDOM](#))  
*createProcessControlBlock* creates a new Process Control Block and adds it to the ready queue.
- bool [deleteProcessControlBlock](#) (size\_t pid)  
*deleteProcessControlBlock* deletes the process control block with the specified pid.
- bool [blockProcessControlBlock](#) (size\_t pid, [IOEvent](#) ioEvent)  
*blockProcessControlBlock* moves the specified process to the block queue
- bool [unblockProcessControlBlock](#) (size\_t pid)  
*unblockProcessControlBlock* removes the process from the block queue and inserts it into the ready queue
- bool [setActiveProcess](#) (size\_t pid)  
*setActiveProcess* sets the active process to the process with the specified pid.
- [CentralProcessingUnit](#) & [cpu](#) ()  
*cpu* returns a reference to the Central Processing Unit.
- [PCBQueue](#) & [readyQueue](#) ()  
*readyQueue* returns a reference to the Ready Queue
- [PCBQueue](#) & [blockedQueue](#) ()  
*blockedQueue* returns a reference to the Blocked Queue
- const [ProcessControlBlockPtr](#) & [findProcessControlBlock](#) (size\_t pid)  
*findProcessControlBlock* finds a reference to the Process Control Block with the specified id.
- const [IOEventQueue](#) & [ioEventQueue](#) ()  
*ioEventQueue* returns a reference to the io event queue
- size\_t [maxMemory](#) () const  
*maxMemory* returns the maximum amount of memory in the operating system.
- size\_t [usedMemory](#) ()  
*memory* returns the amount of memory used in the operating system.
- void [execute](#) ()

- execute execute a single step in the operating system.*
- void [updateCurrentProcessControlBlock](#) ()  
*updateCurrent sets the next active process in the CPU.*
- void [addOperatingSystemProcess](#) ()  
*addOperatingSystemProcess adds the operating system process*
- [ProcessControlBlockPtr](#) [determineNextProcess](#) ()  
*determineNextProcess determines the next process in the ready queue to run.*
- size\_t [cycleCount](#) () const  
*cycleCount returns the current time cycle count*
- void [reset](#) ()  
*reset resets the operating system.*
- void [setScheduler](#) ([SchedulerType](#) type)  
*setScheduler sets the type of scheduler to use.*
- void [setTimeQuantum](#) (size\_t quantum)  
*setTimeQuantum sets the time quantum*
- void [setMaximumPriority](#) (size\_t priority)  
*setMaximumPriority sets the maximum priority level.*

## Static Public Member Functions

- static std::string [getVersionNumber](#) ()  
*getVersionNumber returns the version number*

### 4.14.1 Detailed Description

The [SmartOS](#) class is the heart of the [SmartOS](#) operating system simulator. All functions to manipulate processes are found here.

Definition at line 36 of file SmartOS.h.

### 4.14.2 Constructor & Destructor Documentation

#### 4.14.2.1 SmartOS()

```
SmartOS::SmartOS (
    size_t memory )
```

[SmartOS](#) construct a new simulator with the specified maximum memory.

#### Parameters

<i>memory</i>	the maximum amount of memory
---------------	------------------------------

Definition at line 13 of file SmartOS.cpp.



### 4.14.3 Member Function Documentation

#### 4.14.3.1 addOperatingSystemProcess()

```
void SmartOS::addOperatingSystemProcess ( )
```

addOperatingSystemProcess adds the operating system process

Definition at line 323 of file SmartOS.cpp.

#### 4.14.3.2 blockedQueue()

```
PCBQueue & SmartOS::blockedQueue ( )
```

blockedQueue returns a reference to the Blocked Queue

##### Returns

a reference to the Blocked Queue

Definition at line 185 of file SmartOS.cpp.

#### 4.14.3.3 blockProcessControlBlock()

```
bool SmartOS::blockProcessControlBlock (
    size_t pid,
    IOEvent ioEvent )
```

blockProcessControlBlock moves the specified process to the block queue

##### Parameters

<i>pid</i>	the process's id
<i>ioEvent</i>	the event that being waited on

##### Returns

true if successful

Definition at line 76 of file SmartOS.cpp.

#### 4.14.3.4 cpu()

```
CentralProcessingUnit & SmartOS::cpu ( )
```

cpu returns a reference to the Central Processing Unit.

##### Returns

a reference to the Central Processing Unit

Definition at line 175 of file SmartOS.cpp.

#### 4.14.3.5 createProcessControlBlock()

```
void SmartOS::createProcessControlBlock (
    size_t pid,
    size_t memory,
    ProcessType type = ProcessType::RANDOM )
```

createProcessControlBlock creates a new Process Control Block and adds it to the ready queue.

##### Parameters

<i>pid</i>	the pid
<i>memory</i>	the amount of memory needed for the processes
<i>type</i>	the type of process.

Definition at line 34 of file SmartOS.cpp.

#### 4.14.3.6 cycleCount()

```
size_t SmartOS::cycleCount ( ) const
```

cycleCount returns the current time cycle count

##### Returns

the current time cycle count

Definition at line 450 of file SmartOS.cpp.

#### 4.14.3.7 deleteProcessControlBlock()

```
bool SmartOS::deleteProcessControlBlock (
    size_t pid )
```

deleteProcessControlBlock deletes the process control block with the specified pid.

**Parameters**

<i>pid</i>	the process's id
------------	------------------

**Returns**

true if successful

Definition at line 50 of file SmartOS.cpp.

**4.14.3.8 determineNextProcess()**

```
ProcessControlBlockPtr SmartOS::determineNextProcess ( )
```

determineNextProcess determines the next process in the ready queue to run.

**Returns**

the next process in the ready queue to run.

**4.14.3.9 execute()**

```
void SmartOS::execute ( )
```

execute execute a single step in the operating system.

Definition at line 343 of file SmartOS.cpp.

**4.14.3.10 findProcessControlBlock()**

```
const ProcessControlBlockPtr & SmartOS::findProcessControlBlock (
    size_t pid )
```

findProcessControlBlock finds a reference to the Process Control Block with the specified id.

**Parameters**

<i>pid</i>	the process's id
------------	------------------

**Returns**

a reference to the Process Control Block if found, otherwise a null pcb.

Definition at line 190 of file SmartOS.cpp.

**4.14.3.11 getVersionNumber()**

```
static std::string SmartOS::getVersionNumber ( ) [inline], [static]
```

getVersionNumber returns the version number

**Returns**

the version number

Definition at line 43 of file SmartOS.h.

**4.14.3.12 ioEventQueue()**

```
const IOEventQueue & SmartOS::ioEventQueue ( )
```

ioEventQueue returns a reference to the io event queue

**Returns**

a reference to the io event queue

Definition at line 217 of file SmartOS.cpp.

**4.14.3.13 maxMemory()**

```
size_t SmartOS::maxMemory ( ) const
```

maxMemory returns the maximum amount of memory in the operating system.

**Returns**

the maximum amount of memory in the operating system

Definition at line 480 of file SmartOS.cpp.

#### 4.14.3.14 nextSequentialPID()

```
size_t SmartOS::nextSequentialPID ( )
```

nextSequentialPID the next process id that is available starting from 0.

##### Returns

the next process id

Definition at line 23 of file SmartOS.cpp.

#### 4.14.3.15 readyQueue()

```
PCBQueue & SmartOS::readyQueue ( )
```

readyQueue returns a reference to the Ready Queue

##### Returns

a reference to the Ready Queue

Definition at line 180 of file SmartOS.cpp.

#### 4.14.3.16 reset()

```
void SmartOS::reset ( )
```

reset resets the operating system.

Definition at line 455 of file SmartOS.cpp.

#### 4.14.3.17 setActiveProcess()

```
bool SmartOS::setActiveProcess (
    size_t pid )
```

setActiveProcess sets the active process to the process with the specified pid.

##### Parameters

<i>pid</i>	the process's pid
------------	-------------------

**Returns**

true if successful

Definition at line 135 of file SmartOS.cpp.

**4.14.3.18 setMaximumPriority()**

```
void SmartOS::setMaximumPriority (
    size_t priority )
```

setMaximumPriority sets the maximum priority level.

**Parameters**

<i>priority</i>	the priority level
-----------------	--------------------

Definition at line 475 of file SmartOS.cpp.

**4.14.3.19 setScheduler()**

```
void SmartOS::setScheduler (
    SchedulerType type )
```

setScheduler sets the type of scheduler to use.

**Parameters**

<i>type</i>	the scheduler type
-------------	--------------------

Definition at line 465 of file SmartOS.cpp.

**4.14.3.20 setTimeQuantum()**

```
void SmartOS::setTimeQuantum (
    size_t quantum )
```

setTimeQuantum sets the time quantum

**Parameters**

<i>quantum</i>	the time quantum
----------------	------------------

Definition at line 470 of file SmartOS.cpp.

#### 4.14.3.21 unblockProcessControlBlock()

```
bool SmartOS::unblockProcessControlBlock (
    size_t pid )
```

unblockProcessControlBlock removes the process from the block queue and inserts it into the ready queue

##### Parameters

<i>pid</i>	the process's id
------------	------------------

##### Returns

true if successful

Definition at line 109 of file SmartOS.cpp.

#### 4.14.3.22 updateCurrentProcessControlBlock()

```
void SmartOS::updateCurrentProcessControlBlock ( )
```

updateCurrent sets the next active process in the CPU.

Definition at line 245 of file SmartOS.cpp.

#### 4.14.3.23 usedMemory()

```
size_t SmartOS::usedMemory ( )
```

memory returns the amount of memory used in the operating system.

##### Returns

the amount of memory used in the operating system

Definition at line 222 of file SmartOS.cpp.

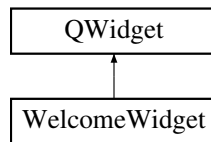
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS\_Core/SmartOS.h
- /home/nicholas/qtcreator/smartos/SmartOS\_Core/SmartOS.cpp

## 4.15 WelcomeWidget Class Reference

```
#include <WelcomeWidget.h>
```

Inheritance diagram for WelcomeWidget:



### Public Member Functions

- [WelcomeWidget](#) ([MainWindow](#) \*mainWindow, QWidget \*parent=nullptr)

#### 4.15.1 Detailed Description

Definition at line 8 of file WelcomeWidget.h.

#### 4.15.2 Constructor & Destructor Documentation

##### 4.15.2.1 WelcomeWidget()

```
WelcomeWidget::WelcomeWidget (
    MainWindow * mainWindow,
    QWidget * parent = nullptr )
```

Definition at line 8 of file WelcomeWidget.cpp.

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

- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/WelcomeWidget.h](#)
- [/home/nicholas/qtcreator/smartos/SmartOS\\_GUI/WelcomeWidget.cpp](#)



## Chapter 5

# File Documentation

### 5.1 /home/nicholas/qtcreator/smartos/SmartOS\_Core/CentralProcessingUnit.cpp File Reference

```
#include "CentralProcessingUnit.h"
```

### 5.2 /home/nicholas/qtcreator/smartos/SmartOS\_Core/CentralProcessingUnit.h File Reference

```
#include "ProcessControlBlock.h"
```

#### Classes

- class [CentralProcessingUnit](#)  
*The [CentralProcessingUnit](#) class holds the current process.*

### 5.3 /home/nicholas/qtcreator/smartos/SmartOS\_Core/Global.h File Reference

```
#include <QtCore/qglobal.h>
```

#### Macros

- #define [SMARTOS\\_CORESHARED\\_EXPORT](#) Q\_DECL\_IMPORT

#### 5.3.1 Macro Definition Documentation

#### 5.3.1.1 SMARTOS\_CORESHARED\_EXPORT

```
#define SMARTOS_CORESHARED_EXPORT Q_DECL_IMPORT
```

Definition at line 9 of file Global.h.

### 5.4 /home/nicholas/qtcreator/smartos/SmartOS\_Core/IOEvent.cpp File Reference

```
#include "IOEvent.h"
```

### 5.5 /home/nicholas/qtcreator/smartos/SmartOS\_Core/IOEvent.h File Reference

```
#include <memory>
```

#### Classes

- class [IOEvent](#)

### 5.6 /home/nicholas/qtcreator/smartos/SmartOS\_Core/ProcessControlBlock.cpp File Reference

```
#include "ProcessControlBlock.h"
```

### 5.7 /home/nicholas/qtcreator/smartos/SmartOS\_Core/ProcessControlBlock.h File Reference

```
#include "IOEvent.h"  
#include <memory>
```

#### Classes

- class [ProcessControlBlock](#)  
*The [ProcessControlBlock](#) class represents a Process Control BLock in the simulator.*

#### Typedefs

- typedef std::unique\_ptr< [ProcessControlBlock](#) > [ProcessControlBlockPtr](#)

## Enumerations

- enum `ProcessType` {  
`ProcessType::RANDOM`, `ProcessType::INTERACTIVE`, `ProcessType::CPU_BOUND`, `ProcessType::MIXED`,  
`ProcessType::LAST` }

*The ProcessType enum represents the type of process.*

### 5.7.1 Typedef Documentation

#### 5.7.1.1 ProcessControlBlockPtr

```
typedef std::unique_ptr<ProcessControlBlock> ProcessControlBlockPtr
```

Definition at line 127 of file ProcessControlBlock.h.

### 5.7.2 Enumeration Type Documentation

#### 5.7.2.1 ProcessType

```
enum ProcessType [strong]
```

The ProcessType enum represents the type of process.

##### Enumerator

RANDOM	
INTERACTIVE	
CPU_BOUND	
MIXED	
LAST	

Definition at line 11 of file ProcessControlBlock.h.

## 5.8 /home/nicholas/qtcreator/smartos/SmartOS\_Core/SmartOS.cpp File Reference

```
#include "SmartOS.h"
#include <random>
#include <iostream>
```

## 5.9 /home/nicholas/qtcreator/smartos/SmartOS\_Core/SmartOS.h File Reference

```
#include "Global.h"
#include "CentralProcessingUnit.h"
#include "IOEvent.h"
#include "ProcessControlBlock.h"
#include <list>
#include <fstream>
```

### Classes

- class [SmartOS](#)

*The [SmartOS](#) class is the heart of the [SmartOS](#) operating system simulator. All functions to manipulate processes are found here.*

### Typedefs

- typedef std::list< [ProcessControlBlockPtr](#) > [PCBQueue](#)  
*PCBQueue a data structure used to iterate proceses.*
- typedef std::list< [IOEvent](#) > [IOEventQueue](#)  
*IOEventQueue a data structure used to iterate IO events.*

### Enumerations

- enum [SchedulerType](#) { [SchedulerType::DEFAULT](#), [SchedulerType::ROUND\\_ROBIN](#), [SchedulerType::MLFQ](#) }
- The SchedulerType enum represents the different types of process schedulers.*

#### 5.9.1 Typedef Documentation

##### 5.9.1.1 IOEventQueue

```
typedef std::list<IOEvent> IOEventQueue
```

[IOEventQueue](#) a data structure used to iterate IO events.

Definition at line 22 of file SmartOS.h.

##### 5.9.1.2 PCBQueue

```
typedef std::list<ProcessControlBlockPtr> PCBQueue
```

[PCBQueue](#) a data structure used to iterate proceses.

Definition at line 17 of file SmartOS.h.

## 5.9.2 Enumeration Type Documentation

### 5.9.2.1 SchedulerType

enum [SchedulerType](#) [strong]

The SchedulerType enum represents the different types of process schedulers.

#### Enumerator

DEFAULT	
ROUND_ROBIN	
MLFQ	

Definition at line 27 of file SmartOS.h.

## 5.10 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/BlockedQueueWidget.cpp File Reference

```
#include "BlockedQueueWidget.h"
#include "Globals.h"
#include "ProcessWidget.h"
#include <QLabel>
#include <QListWidgetItem>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

## 5.11 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/BlockedQueueWidget.h File Reference

```
#include <QListWidget>
#include <QWidget>
```

### Classes

- class [BlockedQueueWidget](#)

## 5.12 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/CpuWidget.cpp File Reference

```
#include "CpuWidget.h"
#include "Globals.h"
#include <QLabel>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

## 5.13 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/CpuWidget.h File Reference

```
#include "ProcessWidget.h"
#include <CentralProcessingUnit.h>
#include <ProcessControlBlock.h>
#include <QPaintEvent>
#include <QStackedWidget>
#include <QWidget>
```

### Classes

- class [CpuWidget](#)

## 5.14 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/Globals.h File Reference

```
#include <SmartOS.h>
#include <memory>
```

### Variables

- `std::unique_ptr< SmartOS > g\_SmartOS`

### 5.14.1 Variable Documentation

#### 5.14.1.1 [g\\_SmartOS](#)

```
std::unique_ptr<SmartOS> g\_SmartOS
```

Definition at line 7 of file main.cpp.

## 5.15 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/HelpDialog.cpp File Reference

```
#include "HelpDialog.h"
```

## 5.16 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/HelpDialog.h File Reference

```
#include <QDialog>
```

### Classes

- class [HelpDialog](#)

## 5.17 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/HistoryDialog.cpp File Reference

```
#include "HistoryDialog.h"  
#include <QListWidget>  
#include <QPushButton>  
#include <QVBoxLayout>
```

## 5.18 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/HistoryDialog.h File Reference

```
#include <QDialog>  
#include <QStringList>
```

### Classes

- class [HistoryDialog](#)

## 5.19 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/main.cpp File Reference

```
#include "Globals.h"  
#include "MainWindow.h"  
#include <QApplication>
```

### Functions

- int [main](#) (int argc, char \*argv[])

## Variables

- `const size_t MEMORY = 1024`
- `std::unique_ptr< SmartOS > g_SmartOS = std::make_unique<SmartOS>(MEMORY)`

## 5.19.1 Function Documentation

### 5.19.1.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

Definition at line 9 of file main.cpp.

## 5.19.2 Variable Documentation

### 5.19.2.1 g\_SmartOS

```
std::unique_ptr<SmartOS> g_SmartOS = std::make_unique<SmartOS>(MEMORY)
```

Definition at line 7 of file main.cpp.

### 5.19.2.2 MEMORY

```
const size_t MEMORY = 1024
```

Definition at line 6 of file main.cpp.

## 5.20 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/MainWindow.cpp File Reference

```
#include "MainWindow.h"  
#include "Globals.h"  
#include "HistoryDialog.h"  
#include "ProcessScheduler/ProcessSchedulerWidget.h"  
#include "ProcessScheduler/SchedulingDialog.h"  
#include "WelcomeWidget.h"  
#include <QDesktopServices>  
#include <QFileInfo>  
#include <QHBoxLayout>  
#include <QListWidget>  
#include <QListWidgetItem>  
#include <QMenu>  
#include <QMenuBar>  
#include <QMessageBox>  
#include <QPushButton>  
#include <QVBoxLayout>  
#include <SmartOS.h>  
#include <QDebug>
```



## 5.21 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/MainWindow.h File Reference

```
#include <QMainWindow>
#include <QStackedWidget>
#include <QStringList>
#include <QWidget>
```

### Classes

- class [MainWindow](#)

## 5.22 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessCreationDialog.cpp File Reference

```
#include "ProcessCreationDialog.h"
#include "Globals.h"
#include <QDebug>
#include <QGridLayout>
#include <QLabel>
#include <QMessageBox>
#include <QPushButton>
```

## 5.23 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessCreationDialog.h File Reference

```
#include <ProcessControlBlock.h>
#include <QComboBox>
#include <QDialog>
#include <QLineEdit>
```

### Classes

- class [ProcessCreationDialog](#)

## 5.24 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessSchedulerWidget.cpp File Reference

```
#include "ProcessSchedulerWidget.h"
#include "ProcessCreationDialog.h"
#include "MainWindow.h"
#include <QAction>
#include <QBoxLayout>
#include <QHBoxLayout>
#include <QIcon>
#include <QInputDialog>
#include <QLabel>
#include <QListView>
#include <QPushButton>
#include <QSlider>
#include <QStringListModel>
#include <QToolBar>
#include <QVBoxLayout>
#include <memory>
#include <random>
#include <SmartOS.h>
#include <QDebug>
```

### Variables

- `std::unique_ptr< SmartOS > g_SmartOS`

### 5.24.1 Variable Documentation

#### 5.24.1.1 g\_SmartOS

```
std::unique_ptr<SmartOS> g_SmartOS
```

Definition at line 7 of file main.cpp.

## 5.25 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/ProcessSchedulerWidget.h File Reference

```
#include "BlockedQueueWidget.h"
#include "CpuWidget.h"
#include "ReadyQueueWidget.h"
#include <QAction>
#include <QLabel>
#include <QPaintEvent>
#include <QSlider>
#include <QTimer>
#include <QWidget>
```

## Classes

- class [ProcessSchedulerWidget](#)

## 5.26 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/SchedulingDialog.cpp File Reference

```
#include "SchedulingDialog.h"
#include <QGridLayout>
#include <QMessageBox>
#include <QPushButton>
#include <QRadioButton>
```

## 5.27 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessScheduler/SchedulingDialog.h File Reference

```
#include <SmartOS.h>
#include <QDialog>
#include <QRadioButton>
#include <QSpinBox>
```

## Classes

- class [SchedulingDialog](#)

## 5.28 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessWidget.cpp File Reference

```
#include "ProcessWidget.h"
#include <QGridLayout>
#include <QLabel>
#include <QPainter>
#include <QDebug>
```

## 5.29 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ProcessWidget.h File Reference

```
#include <ProcessControlBlock.h>
#include <QLabel>
#include <QWidget>
```

## Classes

- class [ProcessWidget](#)

### 5.30 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ReadyQueueWidget.cpp File Reference

```
#include "ReadyQueueWidget.h"
#include "Globals.h"
#include "ProcessWidget.h"
#include <QLabel>
#include <QListWidgetItem>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

### 5.31 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/ReadyQueueWidget.h File Reference

```
#include <QListWidget>
#include <QWidget>
```

## Classes

- class [ReadyQueueWidget](#)

### 5.32 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/WelcomeWidget.cpp File Reference

```
#include "WelcomeWidget.h"
#include <QHBoxLayout>
#include <QIcon>
#include <QPushButton>
#include <QVBoxLayout>
```

### 5.33 /home/nicholas/qtcreator/smartos/SmartOS\_GUI/WelcomeWidget.h File Reference

```
#include "MainWindow.h"
#include <QWidget>
```

## Classes

- class [WelcomeWidget](#)

# Index

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
CentralProcessingUnit.cpp, 39

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
CentralProcessingUnit.h, 39

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
Global.h, 39

/home/nicholas/qtcreator/smartos/SmartOS\_Core/IO↔  
Event.cpp, 40

/home/nicholas/qtcreator/smartos/SmartOS\_Core/IO↔  
Event.h, 40

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
ProcessControlBlock.cpp, 40

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
ProcessControlBlock.h, 40

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
SmartOS.cpp, 41

/home/nicholas/qtcreator/smartos/SmartOS\_Core/↔  
SmartOS.h, 42

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
BlockedQueueWidget.cpp, 43

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
BlockedQueueWidget.h, 43

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/Cpu↔  
Widget.cpp, 44

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/Cpu↔  
Widget.h, 44

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
Globals.h, 44

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
HelpDialog.cpp, 45

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
HelpDialog.h, 45

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
HistoryDialog.cpp, 45

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
HistoryDialog.h, 45

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
MainWindow.cpp, 46

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
MainWindow.h, 47

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessScheduler/ProcessCreationDialog.↔  
cpp, 47

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessScheduler/ProcessCreationDialog.h, 47

/home/nicholas/qtcreator/smartos/SmartOS\_GU↔  
I/ProcessScheduler/ProcessScheduler↔  
Widget.cpp, 48

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessScheduler/SchedulingDialog.cpp, 49

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessScheduler/SchedulingDialog.h, 49

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessWidget.cpp, 49

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ProcessWidget.h, 49

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ReadyQueueWidget.cpp, 50

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
ReadyQueueWidget.h, 50

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
WelcomeWidget.cpp, 50

/home/nicholas/qtcreator/smartos/SmartOS\_GUI/↔  
WelcomeWidget.h, 50

/home/nicholas/qtcreator/smartos/SmartOS\_GU↔  
I/main.cpp, 45

~MainWindow  
MainWindow, 15

addHistory  
MainWindow, 15

addOperatingSystemProcess  
SmartOS, 31

addProcessControlBlock  
ProcessSchedulerWidget, 24

addRandomProcessControlBlocks  
ProcessSchedulerWidget, 24

blockProcessControlBlock  
SmartOS, 31

blockedQueue  
SmartOS, 31

BlockedQueueWidget, 7  
BlockedQueueWidget, 7  
update, 8

CentralProcessingUnit, 8  
CentralProcessingUnit, 8  
currentProcess, 9  
setActiveProcess, 9

clearWaitEvent  
ProcessControlBlock, 17

cpu  
SmartOS, 31

cpuUsageTerm

- ProcessControlBlock, 17
- CpuWidget, 9
  - CpuWidget, 10
  - update, 10
- createProcessControlBlock
  - SmartOS, 32
- currentProcess
  - CentralProcessingUnit, 9
- cycleCount
  - SmartOS, 32
- cycleStamp
  - IOEvent, 13
- deleteProcessControlBlock
  - SmartOS, 32
- determineNextProcess
  - SmartOS, 33
- execute
  - ProcessSchedulerWidget, 24
  - SmartOS, 33
- executeStep
  - ProcessSchedulerWidget, 24
- findProcessControlBlock
  - SmartOS, 33
- g\_SmartOS
  - Globals.h, 44
  - main.cpp, 46
  - ProcessSchedulerWidget.cpp, 48
- getVersionNumber
  - SmartOS, 34
- Global.h
  - SMARTOS\_CORESHARED\_EXPORT, 39
- Globals.h
  - g\_SmartOS, 44
- HelpDialog, 10
  - HelpDialog, 11
- HistoryDialog, 11
  - HistoryDialog, 12
- IOEvent, 12
  - cycleStamp, 13
  - IOEvent, 13
  - Type, 12
  - type, 13
- IOEventQueue
  - SmartOS.h, 42
- ioEvent
  - ProcessControlBlock, 17
- ioEventQueue
  - SmartOS, 34
- ioReqTerm
  - ProcessControlBlock, 18
- main.cpp, 46
- main
  - main.cpp, 46
- main.cpp
  - g\_SmartOS, 46
  - MEMORY, 46
  - main, 46
- MainWindow, 14
  - ~MainWindow, 15
  - addHistory, 15
  - MainWindow, 15
  - showProcessScheduler, 15
  - showUserDocumentation, 15
- maxMemory
  - SmartOS, 34
- memory
  - ProcessControlBlock, 18
- memoryRequired
  - ProcessCreationDialog, 22
- nextSequentialPID
  - SmartOS, 34
- PCBQueue
  - SmartOS.h, 42
- paintEvent
  - ProcessSchedulerWidget, 24
  - ProcessWidget, 26
- pause
  - ProcessSchedulerWidget, 25
- pid
  - ProcessControlBlock, 18
  - ProcessCreationDialog, 22
- priority
  - ProcessControlBlock, 18
- priorityQueues
  - SchedulingDialog, 28
- ProcessControlBlock, 16
  - clearWaitEvent, 17
  - cpuUsageTerm, 17
  - ioEvent, 17
  - ioReqTerm, 18
  - memory, 18
  - pid, 18
  - priority, 18
  - ProcessControlBlock, 17
  - processType, 19
  - setPriority, 19
  - setProcessType, 19
  - setWaitEvent, 20
  - updateCpuUsageTerm, 20
  - updateIoReqTerm, 20
  - updateWaitTerm, 21
  - waitTerm, 21
- ProcessControlBlock.h
  - ProcessControlBlockPtr, 41
  - ProcessType, 41
- ProcessControlBlockPtr
  - ProcessControlBlock.h, 41
- ProcessCreationDialog, 21
  - memoryRequired, 22

- pid, [22](#)
- ProcessCreationDialog, [22](#)
- processType, [22](#)
- ProcessSchedulerWidget, [23](#)
  - addProcessControlBlock, [24](#)
  - addRandomProcessControlBlocks, [24](#)
  - execute, [24](#)
  - executeStep, [24](#)
  - paintEvent, [24](#)
  - pause, [25](#)
  - ProcessSchedulerWidget, [23](#)
  - stop, [25](#)
- ProcessSchedulerWidget.cpp
  - g\_SmartOS, [48](#)
- ProcessType
  - ProcessControlBlock.h, [41](#)
- processType
  - ProcessControlBlock, [19](#)
  - ProcessCreationDialog, [22](#)
- ProcessWidget, [25](#)
  - paintEvent, [26](#)
  - ProcessWidget, [26](#)
  - update, [26](#)
- quantum
  - SchedulingDialog, [28](#)
- readyQueue
  - SmartOS, [35](#)
- ReadyQueueWidget, [26](#)
  - ReadyQueueWidget, [27](#)
  - update, [27](#)
- reset
  - SmartOS, [35](#)
- SMARTOS\_CORESHARED\_EXPORT
  - Global.h, [39](#)
- SchedulerType
  - SmartOS.h, [43](#)
- schedulerType
  - SchedulingDialog, [28](#)
- SchedulingDialog, [27](#)
  - priorityQueues, [28](#)
  - quantum, [28](#)
  - schedulerType, [28](#)
  - SchedulingDialog, [28](#)
- setActiveProcess
  - CentralProcessingUnit, [9](#)
  - SmartOS, [35](#)
- setMaximumPriority
  - SmartOS, [36](#)
- setPriority
  - ProcessControlBlock, [19](#)
- setProcessType
  - ProcessControlBlock, [19](#)
- setScheduler
  - SmartOS, [36](#)
- setTimeQuantum
  - SmartOS, [36](#)
- setWaitEvent
  - ProcessControlBlock, [20](#)
- showProcessScheduler
  - MainWindow, [15](#)
- showUserDocumentation
  - MainWindow, [15](#)
- SmartOS.h
  - IOEventQueue, [42](#)
  - PCBQueue, [42](#)
  - SchedulerType, [43](#)
- SmartOS, [29](#)
  - addOperatingSystemProcess, [31](#)
  - blockProcessControlBlock, [31](#)
  - blockedQueue, [31](#)
  - cpu, [31](#)
  - createProcessControlBlock, [32](#)
  - cycleCount, [32](#)
  - deleteProcessControlBlock, [32](#)
  - determineNextProcess, [33](#)
  - execute, [33](#)
  - findProcessControlBlock, [33](#)
  - getVersionNumber, [34](#)
  - ioEventQueue, [34](#)
  - maxMemory, [34](#)
  - nextSequentialPID, [34](#)
  - readyQueue, [35](#)
  - reset, [35](#)
  - setActiveProcess, [35](#)
  - setMaximumPriority, [36](#)
  - setScheduler, [36](#)
  - setTimeQuantum, [36](#)
  - SmartOS, [30](#)
  - unblockProcessControlBlock, [37](#)
  - updateCurrentProcessControlBlock, [37](#)
  - usedMemory, [37](#)
- stop
  - ProcessSchedulerWidget, [25](#)
- Type
  - IOEvent, [12](#)
- type
  - IOEvent, [13](#)
- unblockProcessControlBlock
  - SmartOS, [37](#)
- update
  - BlockedQueueWidget, [8](#)
  - CpuWidget, [10](#)
  - ProcessWidget, [26](#)
  - ReadyQueueWidget, [27](#)
- updateCpuUsageTerm
  - ProcessControlBlock, [20](#)
- updateCurrentProcessControlBlock
  - SmartOS, [37](#)
- updateIoReqTerm
  - ProcessControlBlock, [20](#)
- updateWaitTerm
  - ProcessControlBlock, [21](#)
- usedMemory

SmartOS, [37](#)

waitTerm

ProcessControlBlock, [21](#)

WelcomeWidget, [38](#)

WelcomeWidget, [38](#)