PA05

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

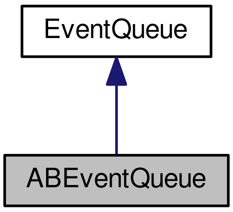
Class Documentation

4.1 ABEventQueue Class Reference

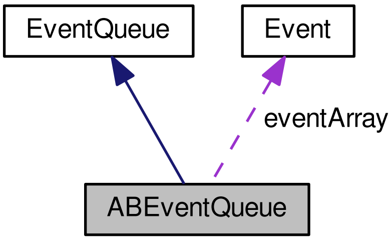
An Array-Based Event Queue.

```
#include <ABEventQueue.h>
```

Inheritance diagram for ABEventQueue:



Collaboration diagram for ABEventQueue:



Public Member Functions

- ABEventQueue (int capacity)

  Constructs an empty queue with given capacity.
- ABEventQueue (int capacity, int lobArrival, int hibArrival, int lobDuration, int hibDuration)

  Constructs a queue with given capacity and fills it randomly.
- ∼ABEventQueue ()
- int length ()
- void addBack (Event newEv)
- Event popFront ()
- Event peekFront ()

Private Attributes

- Event ∗ eventArray
- int cap
- int len

4.1.1 Detailed Description

An Array-Based Event Queue.

Capacity cannot change.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ABEventQueue::ABEventQueue (int *capacity*)

Constructs an empty queue with given capacity.

Parameters

capacity	The capacity of the queue.

4.1.2.2 ABEventQueue::ABEventQueue (int *capacity,* int *lobArrival,* int *hibArrival,* int *lobDuration,* int *hibDuration*)

Constructs a queue with given capacity and fills it randomly.

Parameters

capacity	The capacity of the queue.
lobArrival	Passed to fillRandomly.
hibArrival	Passed to fillRandomly.
lobDuration	Passed to fillRandomly.
hibDuration	Passed to fillRandomly.

4.1.2.3 ABEventQueue::∼ABEventQueue ( )

Cleans up the queue.

4.1.3 Member Function Documentation

4.1.3.1 void ABEventQueue::addBack (Event *newEv*) `[virtual]`

Adds to the back of the queue.

Implements EventQueue.

4.1.3.2 int ABEventQueue::length ( ) `[virtual]`

Get the length of the queue.

Implements EventQueue.

4.1.3.3 Event ABEventQueue::peekFront ( ) `[virtual]`

Peek at the front of the queue.

Implements EventQueue.

4.1.3.4 Event ABEventQueue::popFront ( ) `[virtual]`

Get and remove the event at the front of the queue.

Implements EventQueue.

4.1.4 Member Data Documentation

4.1.4.1 int ABEventQueue::cap `[private]`

4.1.4.2 Event∗ ABEventQueue::eventArray `[private]`

4.1.4.3 int ABEventQueue::len `[private]`

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

- ABEventQueue.h
- ABEventQueue.cpp

4.2 Event Struct Reference

```
#include <Event.h>
```

Public Attributes

- int arrival
- int duration

4.2.1 Member Data Documentation

4.2.1.1 int Event::arrival

4.2.1.2 int Event::duration

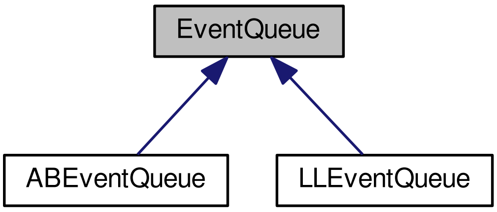
The documentation for this struct was generated from the following file:

- Event.h

4.3 EventQueue Class Reference

```
#include <EventQueue.h>
```

Inheritance diagram for EventQueue:



Public Member Functions

- virtual ∼EventQueue ()

  Cleans up the queue.
- virtual int length ()=0

  Get the length of the queue.
- bool isEmpty ()

  Get whether the queue is empty.
- void fillRandomly (int num, int lobArrival, int hibArrival, int lobDuration, int hibDuration)

  Fills the queue with randomly generated events.
- virtual void addBack (Event newEv)=0

  Adds to the back of the queue.
- virtual Event popFront ()=0

  Get and remove the event at the front of the queue.
- virtual Event peekFront ()=0

  Peek at the front of the queue.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 EventQueue::∼EventQueue ( ) [virtual]

Cleans up the queue.

4.3.2 Member Function Documentation

4.3.2.1 virtual void EventQueue::addBack (Event *newEv*) [pure virtual]

Adds to the back of the queue.

Parameters

newEv	The event to add.

Implemented in LLEventQueue, and ABEventQueue.

4.3.2.2 void EventQueue::fillRandomly (int *num,* int *lobArrival,* int *hibArrival,* int *lobDuration,* int *hibDuration*)

Fills the queue with randomly generated events.

num The number of events to generate.

Parameters

lobArrival	The low bound for arrival times.
hibArrival	The high bound for arrival times.
lobDuration	The low bound for duration times.
hibDuration	The high bound for duration times.

4.3.2.3 bool EventQueue::isEmpty ( )

Get whether the queue is empty.

Returns

True if empty; false otherwise.

4.3.2.4 virtual int EventQueue::length ( ) `[pure virtual]`

Get the length of the queue.

Returns

The length of the queue.

Implemented in LLEventQueue, and ABEventQueue.

4.3.2.5 virtual Event EventQueue::peekFront ( ) `[pure virtual]`

Peek at the front of the queue.

Returns

The event at the front of the queue.

Implemented in LLEventQueue, and ABEventQueue.

4.3.2.6 virtual Event EventQueue::popFront ( ) `[pure virtual]`

Get and remove the event at the front of the queue.

Returns

The event at the front of the queue.

Implemented in LLEventQueue, and ABEventQueue.

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

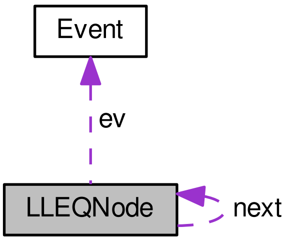
- EventQueue.h
- EventQueue.cpp

4.4 LLEQNode Struct Reference

A node in an LLEventQueue.

```
#include <LLEventQueue.h>
```

Collaboration diagram for LLEQNode:



Public Attributes

- Event ev
- LLEQNode ∗ next

4.4.1 Detailed Description

A node in an LLEventQueue.

4.4.2 Member Data Documentation

4.4.2.1 Event LLEQNode::ev

4.4.2.2 LLEQNode∗ LLEQNode::next

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

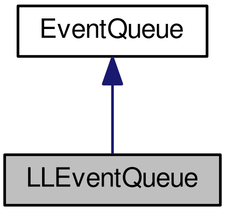
- LLEventQueue.h

4.5 LLEventQueue Class Reference

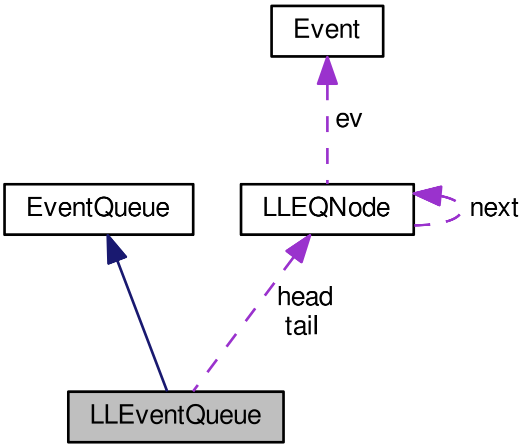
A Linked-List-Based Event Queue.

```
#include <LLEventQueue.h>
```

Inheritance diagram for LLEventQueue:



Collaboration diagram for LLEventQueue:



Public Member Functions

- LLEventQueue ()

  Constructs an empty queue.
- LLEventQueue (int num, int lobArrival, int hibArrival, int lobDuration, int hibDuration)

  Constructs a queue and fills it randomly.
- ∼LLEventQueue ()
- int length ()
- void addBack (Event newEv)
- Event popFront ()
- Event peekFront ()

Public Attributes

- LLEQNode ∗ head
- LLEQNode ∗ tail
- int len

4.5.1 Detailed Description

A Linked-List-Based Event Queue.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 LLEventQueue::LLEventQueue ( )

Constructs an empty queue.

4.5.2.2 LLEventQueue::LLEventQueue (int *num*, int *lobArrival*, int *hibArrival*, int *lobDuration*, int *hibDuration*)

Constructs a queue and fills it randomly.

Parameters

num	Passed to fillRandomly.
lobArrival	Passed to fillRandomly.
hibArrival	Passed to fillRandomly.
lobDuration	Passed to fillRandomly.
hibDuration	Passed to fillRandomly.

4.5.2.3 LLEventQueue::∼LLEventQueue ( )

Cleans up the queue.

4.5.3 Member Function Documentation

4.5.3.1 void LLEventQueue::addBack (Event *newEv*) [virtual]

Adds to the back of the queue.

Implements EventQueue.

4.5.3.2 int LLEventQueue::length ( ) [virtual]

Get the length of the queue.

Implements EventQueue.

4.5.3.3 Event LLEventQueue::peekFront ( ) [virtual]

Peek at the front of the queue.

Implements EventQueue.

4.5.3.4 Event LLEventQueue::popFront ( ) [virtual]

Get and remove the event at the front of the queue.

Implements EventQueue.

4.5.4 Member Data Documentation

4.5.4.1 LLEQNode* LLEventQueue::head

4.5.4.2 int LLEventQueue::len

4.5.4.3 LLEQNode* LLEventQueue::tail

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

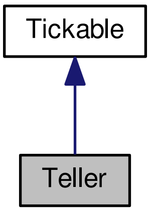
- LLEventQueue.h
- LLEventQueue.cpp

4.6 Teller Class Reference

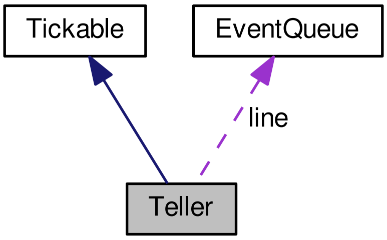
Models a bank teller.

`#include <TellerSetup.h>`

Inheritance diagram for Teller:



Collaboration diagram for Teller:



Public Member Functions

- Teller (EventQueue ∗ln, int ∗nc, double ∗twt, double ∗mwt)

  Construct a Teller with given stat vars and source line.
- void tick (int now)
- int whenNext ()
- void finish (int now)

  Finishes the Teller model. Should be called at sim end.

Public Attributes

- int idleTime

Protected Attributes

- EventQueue ∗ line
- int ∗ numCustomers
- double ∗ totWaitTime
- double ∗ maxWaitTime
- int idleStart
- bool busy
- int busyUntil

4.6.1 Detailed Description

Models a bank teller.

For use by TellerSetup.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Teller::Teller (**EventQueue** ∗ *ln,* int ∗ *nc,* double ∗ *twt,* double ∗ *mwt*)

Construct a Teller with given stat vars and source line.

Parameters

ln	The line to draw customers from.
nc	Pointer to the customer counter.
twt	Pointer to the wait time counter.
mwt	Pointer to the max wait time.

4.6.3 Member Function Documentation

4.6.3.1 void Teller::finish (int *now*)

Finishes the Teller model. Should be called at sim end.

Parameters

now	The last tick.

4.6.3.2 void Teller::tick (int *now*) [virtual]

Do one tick at the given time.

Implements Tickable.

4.6.3.3 int Teller::whenNext ( ) [virtual]

Get when this tickable needs to be ticked next.

Implements Tickable.

4.6.4 Member Data Documentation

4.6.4.1 bool Teller::busy [protected]

4.6.4.2 int Teller::busyUntil [protected]

4.6.4.3 int Teller::idleStart [protected]

4.6.4.4 int Teller::idleTime

4.6.4.5 EventQueue* Teller::line [protected]

4.6.4.6 double* Teller::maxWaitTime [protected]

4.6.4.7 int* Teller::numCustomers [protected]

4.6.4.8 double* Teller::totWaitTime [protected]

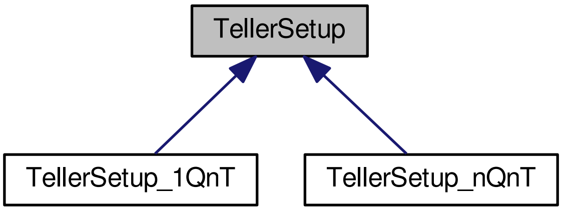
The documentation for this class was generated from the following files:

- TellerSetup.h
- TellerSetup.cpp

4.7 TellerSetup Class Reference

```
#include <TellerSetup.h>
```

Inheritance diagram for TellerSetup:



Public Member Functions

- virtual void simulate (EventQueue ∗pEq)=0

  Run the simulation (1 trial).
- void printStats (std::ostream &out)

  Output setup stats.

Protected Attributes

- double numTrials
- double totCPUTime
- double totVirtTime
- double totAvgWaitTime
- double totMaxWaitTime
- double totAvgLineLen
- double totMaxLineLen
- std::vector< double > totIdleTimePerTeller
- int numTellers
- int numLines

4.7.1 Member Function Documentation

4.7.1.1 void TellerSetup::printStats (std::ostream & *out*)

Output setup stats.

Parameters

out	The stream to write to.

4.7.1.2 virtual void TellerSetup::simulate (EventQueue ∗ *pEq*) [pure virtual]

Run the simulation (1 trial).

Parameters

pEq	Pointer to the EventQueue of events to feed the simulation.

Implemented in TellerSetup_1QnT, and TellerSetup_nQnT.

4.7.2 Member Data Documentation

4.7.2.1 int TellerSetup::numLines [protected]

4.7.2.2 int TellerSetup::numTellers [protected]

4.7.2.3 double TellerSetup::numTrials [protected]

4.7.2.4 double TellerSetup::totAvgLineLen [protected]

4.7.2.5 double TellerSetup::totAvgWaitTime [protected]

4.7.2.6 double TellerSetup::totCPUTime [protected]

4.7.2.7 std::vector<double> TellerSetup::totIdleTimePerTeller [protected]

4.7.2.8 double TellerSetup::totMaxLineLen [protected]

4.7.2.9 double TellerSetup::totMaxWaitTime [protected]

4.7.2.10 double TellerSetup::totVirtTime [protected]

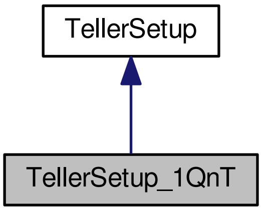
The documentation for this class was generated from the following files:

- TellerSetup.h
- TellerSetup.cpp

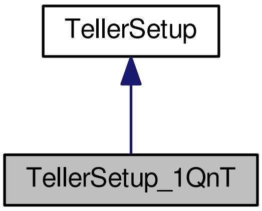
4.8 TellerSetup_1QnT Class Reference

```
#include <TellerSetup_1QnT.h>
```

Inheritance diagram for TellerSetup_1QnT:



Collaboration diagram for TellerSetup_1QnT:



Public Member Functions

- TellerSetup_1QnT (int n)

    Constructs a 1QnT Teller Setup with given n.
- void simulate (EventQueue ∗pEq)

Additional Inherited Members

4.8.1 Constructor & Destructor Documentation

4.8.1.1 TellerSetup_1QnT::TellerSetup_1QnT (int *n*)

Constructs a 1QnT Teller Setup with given n.

Parameters

n	The number of tellers.

4.8.2 Member Function Documentation

4.8.2.1 void TellerSetup_1QnT::simulate (EventQueue * *pEq*) [virtual]

Run the simulation (1 trial).

Implements TellerSetup.

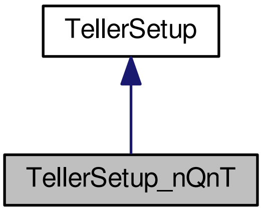
The documentation for this class was generated from the following files:

- TellerSetup_1QnT.h
- TellerSetup_1QnT.cpp

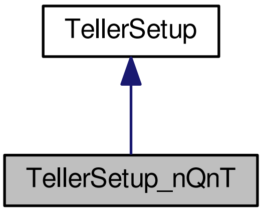
4.9 TellerSetup_nQnT Class Reference

```
#include <TellerSetup_nQnT.h>
```

Inheritance diagram for TellerSetup_nQnT:



Collaboration diagram for TellerSetup_nQnT:



Public Member Functions

- TellerSetup_nQnT (int n)

  Constructs a nQnT Teller Setup with given n.
- void simulate (EventQueue ∗pEq)

Additional Inherited Members

4.9.1 Constructor & Destructor Documentation

4.9.1.1 TellerSetup_nQnT::TellerSetup_nQnT (int *n*)

Constructs a nQnT Teller Setup with given n.

Parameters

n	The number of tellers and queues.

4.9.2 Member Function Documentation

4.9.2.1 void TellerSetup_nQnT::simulate (EventQueue ∗ *pEq*) [virtual]

Run the simulation (1 trial).

Implements TellerSetup.

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

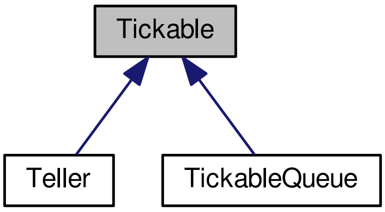
- TellerSetup_nQnT.h
- TellerSetup_nQnT.cpp

4.10 Tickable Class Reference

A Tickable object.

`#include <TellerSetup.h>`

Inheritance diagram for Tickable:



Public Member Functions

- virtual void tick (int now)=0

  Do one tick at the given time.
- virtual int whenNext ()=0

  Get when this tickable needs to be ticked next.

4.10.1 Detailed Description

A Tickable object.

For use by TellerSetup.

4.10.2 Member Function Documentation

4.10.2.1 virtual void Tickable::tick (int *now*) [pure virtual]

Do one tick at the given time.

Parameters

now	The current time.

Implemented in TickableQueue, and Teller.

4.10.2.2 virtual int Tickable::whenNext ( ) [pure virtual]

Get when this tickable needs to be ticked next.

Returns -1 when there is no future event.

Returns

When this tickable needs to be ticked next.

Implemented in TickableQueue, and Teller.

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

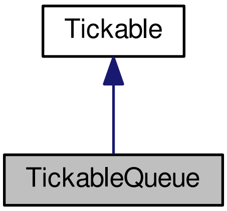
- TellerSetup.h

4.11 TickableQueue Class Reference

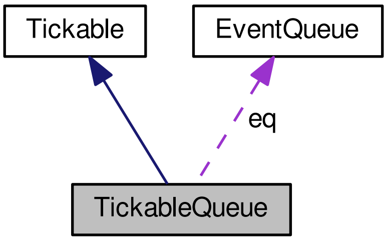
An EventQueue wrapper that enables tick functionality.

```
#include <TellerSetup.h>
```

Inheritance diagram for TickableQueue:



Collaboration diagram for TickableQueue:



Public Member Functions

- TickableQueue (EventQueue ∗evQ, double ∗tll, double ∗mll)

  Constructs a TickableQueue around the given EventQueue.
- void regDestQueue (EventQueue ∗dest)

  Register a destination queue for this queue to feed into.
- void tick (int now)
- int whenNext ()

Private Attributes

- EventQueue ∗ eq
- std::vector< EventQueue ∗ > destLines
- double ∗ totLineLen
- double ∗ maxLineLen
- int lastTicked

4.11.1 Detailed Description

An EventQueue wrapper that enables tick functionality.

An EventQueue wrapper that can be "ticked".

4.11.2 Constructor & Destructor Documentation

4.11.2.1 TickableQueue::TickableQueue (EventQueue ∗ *evQ*, double ∗ *tll*, double ∗ *mll*)

Constructs a TickableQueue around the given EventQueue.

Parameters

evQ	The EventQueue to wrap.
tll	Pointer to the total line length over time counter.
mll	Pointer to the max line length counter.

4.11.3 Member Function Documentation

4.11.3.1 void TickableQueue::regDestQueue (EventQueue ∗ *dest*)

Register a destination queue for this queue to feed into.

Parameters

dest	The queue to register.

4.11.3.2 void TickableQueue::tick (int *now*) `[virtual]`

Do one tick at the given time.

Implements Tickable.

4.11.3.3 int TickableQueue::whenNext ( ) `[virtual]`

Get when this tickable needs to be ticked next.

Implements Tickable.

4.11.4 Member Data Documentation

4.11.4.1 std::vector<EventQueue∗> TickableQueue::destLines [private]

4.11.4.2 EventQueue∗ TickableQueue::eq [private]

4.11.4.3 int TickableQueue::lastTicked [private]

4.11.4.4 double∗ TickableQueue::maxLineLen [private]

4.11.4.5 double∗ TickableQueue::totLineLen [private]

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

- TellerSetup.h
- TellerSetup.cpp

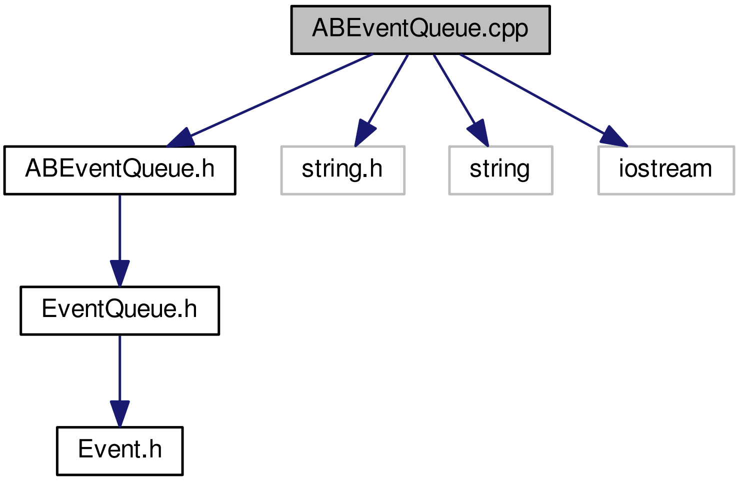
Chapter 5

File Documentation

5.1 ABEventQueue.cpp File Reference

```
#include "ABEventQueue.h"
#include <string.h>
#include <string>
#include <iostream>
```

Include dependency graph for ABEventQueue.cpp:

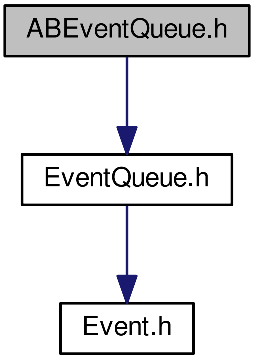


5.2 ABEventQueue.h File Reference

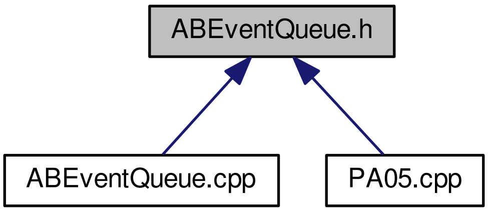
Declares the Array-Based Event Queue class.

```
#include "EventQueue.h"
```

Include dependency graph for ABEventQueue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class ABEventQueue

  An Array-Based Event Queue.

5.2.1 Detailed Description

Declares the Array-Based Event Queue class.

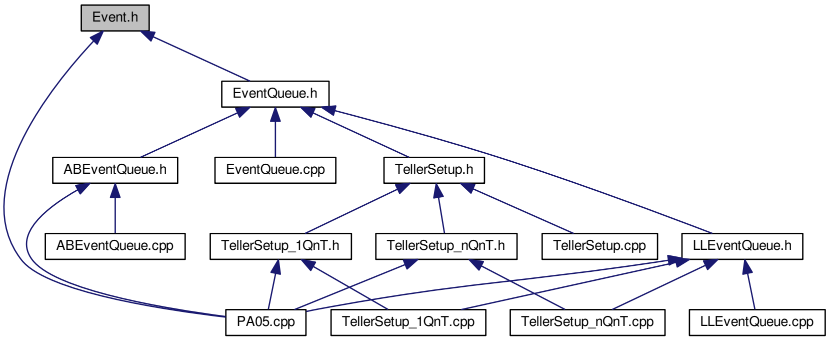
Author

Matthew Bauer

5.3 Event.h File Reference

Declares the Event structure.

This graph shows which files directly or indirectly include this file:



Classes

- struct Event

5.3.1 Detailed Description

Declares the Event structure.

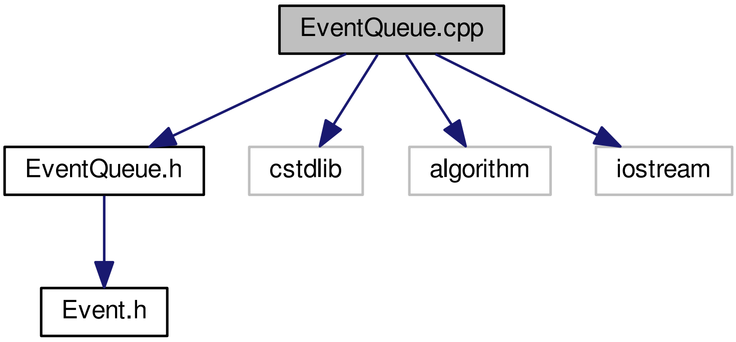
Author

Matthew Bauer

5.4 EventQueue.cpp File Reference

```
#include "EventQueue.h"
#include <cstdlib>
#include <algorithm>
#include <iostream>
```

Include dependency graph for EventQueue.cpp:

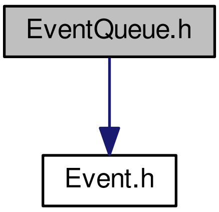


5.5 EventQueue.h File Reference

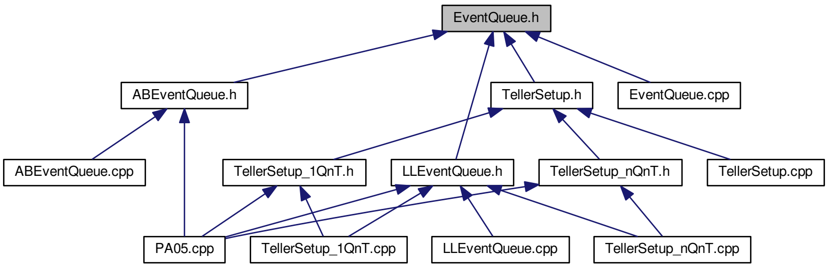
Declares the EventQueue abstract class.

```
#include "Event.h"
```

Include dependency graph for EventQueue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class EventQueue

5.5.1 Detailed Description

Declares the EventQueue abstract class.

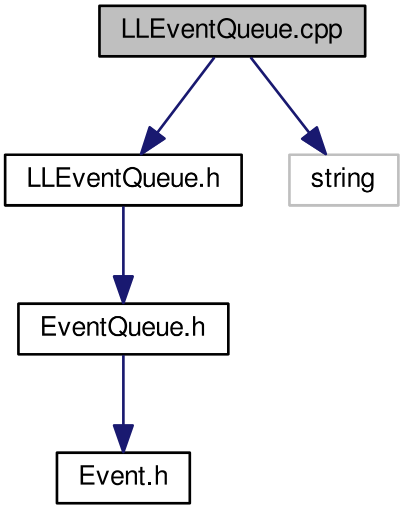
Author

Matthew Bauer

5.6 LLEventQueue.cpp File Reference

```
#include "LLEventQueue.h"
#include <string>
```

Include dependency graph for LLEventQueue.cpp:

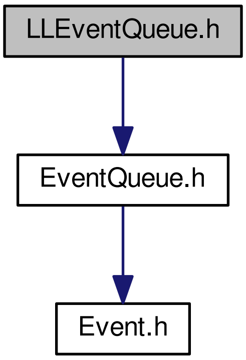


5.7 LLEventQueue.h File Reference

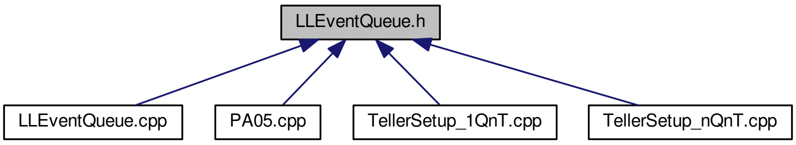
Declares the Linked-List-Based Event Queue class.

```
#include "EventQueue.h"
```

Include dependency graph for LLEventQueue.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct LLEQNode

  A node in an LLEventQueue.
- class LLEventQueue

  A Linked-List-Based Event Queue.

5.7.1 Detailed Description

Declares the Linked-List-Based Event Queue class.

Author

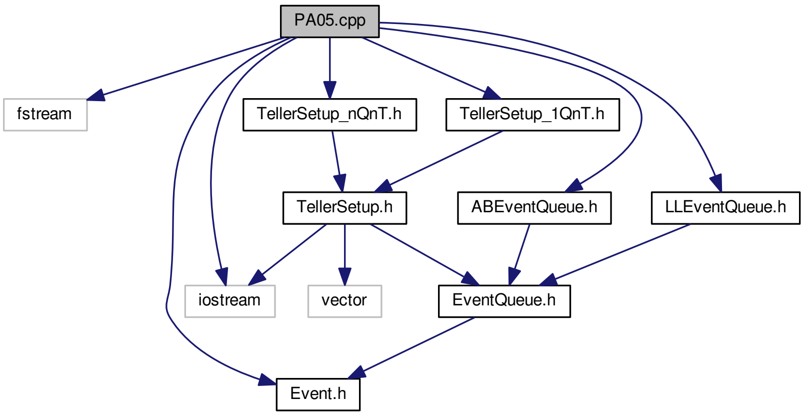
Matthew Bauer

5.8 PA05.cpp File Reference

Main file for CS302/PA05.

```
#include <fstream>
#include <iostream>
#include "Event.h"
#include "ABEventQueue.h"
#include "LLEventQueue.h"
#include "TellerSetup_1QnT.h"
#include "TellerSetup_nQnT.h"
```

Include dependency graph for PA05.cpp:



Functions

- void TestSetup (TellerSetup ∗setup, std::ostream &out)

  Test the given TellerSetup on 10 samples.
- int main ()

  Program entry point.

Variables

- const int NUM_EVENTS = 99999

  The number of events to use per simulation.
- const int MIN_ARRIVAL = 0
- const int MAX_ARRIVAL = 100000
- const int MIN_DURATION = 1
- const int MAX_DURATION = 100

5.8.1 Detailed Description

Main file for CS302/PA05.

Author

Matthew Bauer

5.8.2 Function Documentation

5.8.2.1 int main ( )

Program entry point.

5.8.2.2 void TestSetup (TellerSetup ∗ *setup,* std::ostream & *out*)

Test the given TellerSetup on 10 samples.

Uses array-based queues.

Parameters

pEvQ	A pointer to the EventQueue to test.
out	The stream to send output to.

5.8.3 Variable Documentation

5.8.3.1 const int MAX_ARRIVAL = 100000

5.8.3.2 const int MAX_DURATION = 100

5.8.3.3 const int MIN_ARRIVAL = 0

5.8.3.4 const int MIN_DURATION = 1

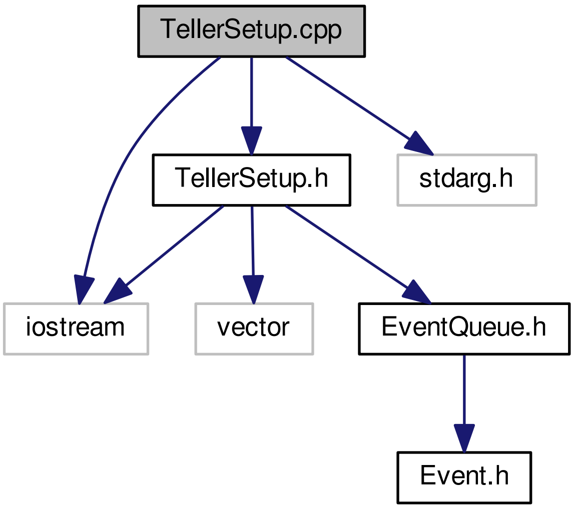
5.8.3.5 const int NUM_EVENTS = 99999

The number of events to use per simulation.

5.9 TellerSetup.cpp File Reference

```
#include "TellerSetup.h"
#include <stdarg.h>
#include <iostream>
```

Include dependency graph for TellerSetup.cpp:



Functions

- int getEarliestTime (const std::vector< int > &times)

  Take the min of the given numbers, with -1 values excluded.

5.9.1 Function Documentation

5.9.1.1 int getEarliestTime (const std::vector< int > & *times*)

Take the min of the given numbers, with -1 values excluded.

For use by TellerSetup.

Parameters

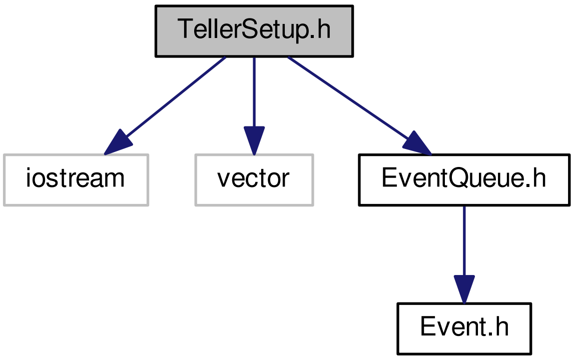
times	The times.

5.10 TellerSetup.h File Reference

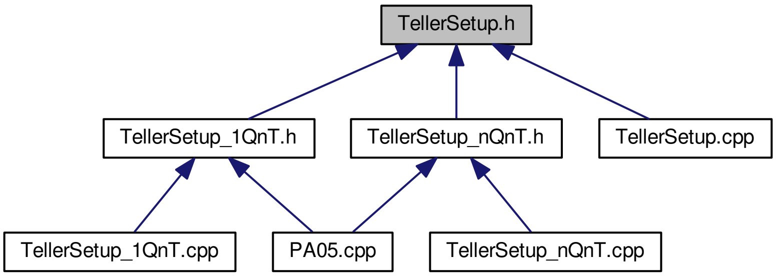
Declares the TellerSetup abstract class.

```
#include <iostream>
#include <vector>
#include "EventQueue.h"
```

Include dependency graph for TellerSetup.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TellerSetup
- class Tickable

  A Tickable object.
- class Teller

  Models a bank teller.
- class TickableQueue

  An EventQueue wrapper that enables tick functionality.

Functions

- int getEarliestTime (const std::vector< int > &times)

  Take the min of the given numbers, with -1 values excluded.

5.10.1 Detailed Description

Declares the TellerSetup abstract class.

Author

Matthew Bauer

5.10.2 Function Documentation

5.10.2.1 int getEarliestTime (const std::vector< int > & *times*)

Take the min of the given numbers, with -1 values excluded.

For use by TellerSetup.

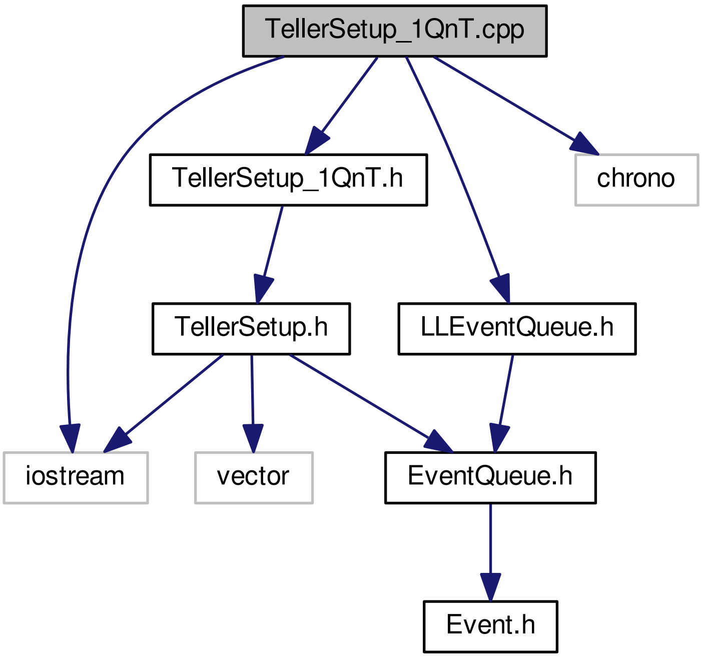
Parameters

times	The times.

5.11 TellerSetup_1QnT.cpp File Reference

```
#include "TellerSetup_1QnT.h"
#include "LLEventQueue.h"
#include <chrono>
#include <iostream>
```

Include dependency graph for TellerSetup_1QnT.cpp:

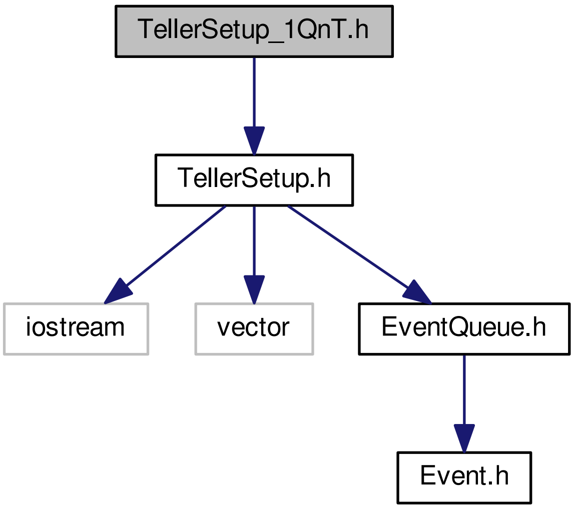


5.12 TellerSetup_1QnT.h File Reference

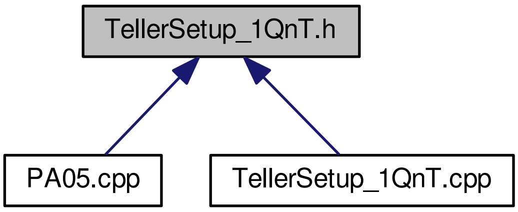
Declares a TellerSetup with 1 Queue and N Tellers.

```
#include "TellerSetup.h"
```

Include dependency graph for TellerSetup_1QnT.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TellerSetup_1QnT

5.12.1 Detailed Description

Declares a TellerSetup with 1 Queue and N Tellers.

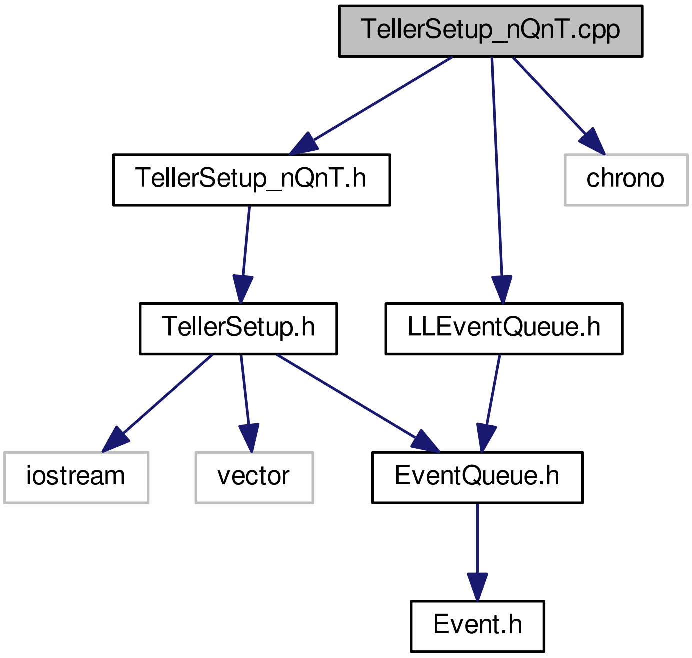
Author

Matthew Bauer

5.13 TellerSetup_nQnT.cpp File Reference

```
#include "TellerSetup_nQnT.h"
#include "LLEventQueue.h"
#include <chrono>
```

Include dependency graph for TellerSetup_nQnT.cpp:

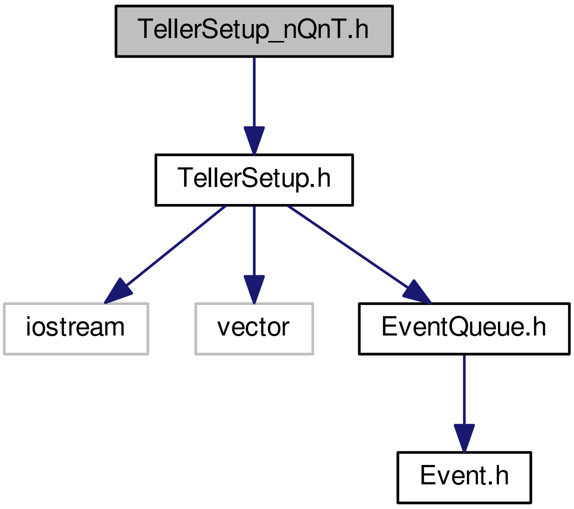


5.14 TellerSetup_nQnT.h File Reference

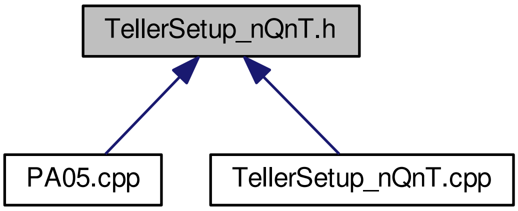
Declares a TellerSetup with N Queues and N Tellers.

```
#include "TellerSetup.h"
```

Include dependency graph for TellerSetup_nQnT.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TellerSetup_nQnT

5.14.1 Detailed Description

Declares a TellerSetup with N Queues and N Tellers.

Author

Matthew Bauer

Index