

StatisticsCapability Reference Manual

Generated by Doxygen 1.5.3

Mon Feb 18 14:37:28 2008

Contents

1	Statistics Capability - Developer Manual	1
1.1	Introduction	1
1.2	Interfaces associated with the capability	1
1.3	Services associated with the capability	1
1.4	Clients associated with the capability	1
1.5	User Manual	1
2	StatisticsCapability Class Index	3
2.1	StatisticsCapability Class List	3
3	StatisticsCapability Class Documentation	5
3.1	StatisticInterface Class Reference	5
3.2	StatisticRepository Class Reference	6
3.3	StatisticService Class Reference	8
3.4	UnknownStatisticException Struct Reference	10

Chapter 1

Statistics Capability - Developer Manual

1.1 Introduction

The purpose of simulators is not only to compute timing information, but also to gather information about the behavior of the different hardware mechanism during the simulation. The Statistic Capability allow a standardize access to event counters within the different modules.

1.2 Interfaces associated with the capability

- [StatisticInterface](#)

1.3 Services associated with the capability

- [StatisticService](#)

1.4 Clients associated with the capability

- none

1.5 User Manual

Corresponding user manual can be found on the [unisim.org](http://unisim.org/wiki) wiki at <http://unisim.org/site/services/statistics/start>

Chapter 2

StatisticsCapability Class Index

2.1 StatisticsCapability Class List

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

StatisticInterface (The interface of the Statistic capability)	5
StatisticRepository (Statistic repository where all the statistics are referenced)	6
StatisticService (Service to inherit from to use the Statistics capability)	8
UnknownStatisticException (Exception raised when searching for a non existing statistic) . . .	10

Chapter 3

StatisticsCapability Class Documentation

3.1 StatisticInterface Class Reference

The interface of the Statistic capability.

```
#include <StatisticService.h>
```

Public Member Functions

- virtual uint64_t [get_statistic](#) (const string &name)=0
Retrieve a statistic from the repository.

3.1.1 Detailed Description

The interface of the Statistic capability.

Definition at line 91 of file StatisticService.h.

3.1.2 Member Function Documentation

3.1.2.1 virtual uint64_t StatisticInterface::get_statistic (const string & name) [pure virtual]

Retrieve a statistic from the repository.

Parameters:

- *name* Name of the statistic to be retrieved

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

- StatisticService.h

3.2 StatisticRepository Class Reference

Statistic repository where all the statistics are referenced.

```
#include <StatisticService.h>
```

Public Member Functions

- [StatisticRepository](#) (const char *name)
Creates a repository associated with a module.
- void [add](#) (const string &name, uint64_t &value)
Add a new statistic to the repository.
- uint64_t [operator\[\]](#) (const string &name)
Retrieve a statistic from the repository.

Friends

- ostream & [operator<<](#) (ostream &os, const [StatisticRepository](#) &r)
Pretty printer of the [StatisticRepository](#) class.

3.2.1 Detailed Description

Statistic repository where all the statistics are referenced.

Definition at line 103 of file StatisticService.h.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 [StatisticRepository::StatisticRepository](#) (const char * *name*) `[inline]`

Creates a repository associated with a module.

Parameters:

name The module name

Definition at line 109 of file StatisticService.h.

3.2.3 Member Function Documentation

3.2.3.1 [uint64_t StatisticRepository::operator\[\]](#) (const string & *name*) `[inline]`

Retrieve a statistic from the repository.

Parameters:

name Name of the statistic to be retrieved

Definition at line 122 of file `StatisticService.h`.

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

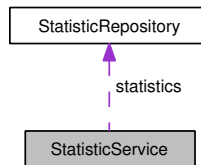
- `StatisticService.h`

3.3 StatisticService Class Reference

Service to inherit from to use the Statistics capability.

```
#include <StatisticService.h>
```

Collaboration diagram for StatisticService:



Public Member Functions

- [StatisticService](#) (const char *name, Object *parent)
Creates a new [StatisticService](#).
- virtual uint64_t [get_statistic](#) (const string &name)
Retrieve a statistic from the repository.

Static Public Member Functions

- static void [dump_statistics](#) (ostream &os)
Dumps all the statistics to an output stream.

Public Attributes

- [StatisticRepository](#) [statistics](#)
Statistics repository.
- ServiceExport< [StatisticInterface](#) > [statistic_export](#)
service port

3.3.1 Detailed Description

Service to inherit from to use the Statistics capability.

Definition at line 148 of file StatisticService.h.

3.3.2 Member Function Documentation

3.3.2.1 `virtual uint64_t StatisticService::get_statistic (const string & name)` [`inline`, `virtual`]

Retrieve a statistic from the repository.

Parameters:

name Name of the statistic to be retrieved

Definition at line 168 of file StatisticService.h.

References statistics.

3.3.2.2 `static void StatisticService::dump_statistics (ostream & os)` [`inline`, `static`]

Dumps all the statistics to an output stream.

Parameters:

os The output stream to dump the statistics to

Definition at line 176 of file StatisticService.h.

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

- StatisticService.h

3.4 UnknownStatisticException Struct Reference

Exception raised when searching for a non existing statistic.

```
#include <StatisticService.h>
```

Public Member Functions

- [UnknownStatisticException](#) (const string &s)
Exception raised when searching for a non existing statistic.

3.4.1 Detailed Description

Exception raised when searching for a non existing statistic.

Definition at line 81 of file StatisticService.h.

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

- StatisticService.h

Index

- dump_statistics
 - StatisticService, [9](#)
- get_statistic
 - StatisticInterface, [5](#)
 - StatisticService, [9](#)
- operator[]
 - StatisticRepository, [6](#)
- StatisticInterface, [5](#)
 - get_statistic, [5](#)
- StatisticRepository, [6](#)
 - operator[], [6](#)
 - StatisticRepository, [6](#)
- StatisticService, [8](#)
 - dump_statistics, [9](#)
 - get_statistic, [9](#)
- UnknownStatisticException, [10](#)