

ZPI_AgentWindows

Generated by Doxygen 1.8.5

Tue Nov 19 2013 21:50:44

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	SystemInfo Namespace Reference	7
4.1.1	Detailed Description	7
4.2	WinAgent Namespace Reference	7
4.2.1	Detailed Description	8
4.2.2	Function Documentation	8
4.2.2.1	explode	8
4.2.2.2	getTicks	8
4.2.2.3	recvallWin	8
4.2.2.4	usleep	9
5	Class Documentation	11
5.1	SystemInfo::Cpu Class Reference	11
5.1.1	Detailed Description	12
5.1.2	Member Function Documentation	12
5.1.2.1	displayCombinedInfo	12
5.1.2.2	displayDetailsInfo	12
5.1.2.3	displayTimesInfo	12
5.2	SystemInfo::Cpu::Details Struct Reference	12
5.2.1	Detailed Description	13
5.3	SystemInfo::FileSystem::Details Struct Reference	13
5.3.1	Detailed Description	14
5.4	SystemInfo::DiagnosticMgr Class Reference	14
5.4.1	Detailed Description	14

5.4.2	Member Function Documentation	15
5.4.2.1	getCpuInfo	15
5.4.2.2	getCpuTemp	15
5.4.2.3	getFileSystemInfo	15
5.4.2.4	getInstance	15
5.4.2.5	getResourcesInfo	15
5.5	SystemInfo::FileSystem Class Reference	16
5.5.1	Detailed Description	16
5.5.2	Member Function Documentation	17
5.5.2.1	displayCombinedInfo	17
5.5.2.2	displayDetailsInfo	17
5.5.2.3	displayUsageInfo	17
5.5.2.4	getFullFsTypeName	17
5.6	SystemInfo::InfoMapper Class Reference	17
5.6.1	Detailed Description	18
5.6.2	Member Function Documentation	18
5.6.2.1	sigarCpuInfoToDetails	18
5.6.2.2	sigarCpuToCpuTimes	18
5.6.2.3	sigarFileSystemToFsDetails	18
5.6.2.4	sigarFileSystemUsageToFsUsage	19
5.6.2.5	sigarResourcesLimitToResources	19
5.7	SystemInfo::ModuleInfo Class Reference	19
5.7.1	Detailed Description	20
5.7.2	Member Function Documentation	20
5.7.2.1	displayCombinedInfo	20
5.7.2.2	displayGroupHeader	20
5.7.2.3	displayModuleHeader	20
5.7.2.4	displayPairInfo	21
5.7.2.5	displaySingleInfo	21
5.8	SystemInfo::Resources Class Reference	21
5.8.1	Detailed Description	23
5.8.2	Member Function Documentation	23
5.8.2.1	displayCombinedInfo	23
5.9	WinAgent::Server Class Reference	23
5.9.1	Detailed Description	24
5.9.2	Member Function Documentation	24
5.9.2.1	configApplied	24
5.9.2.2	configChanged	24
5.9.2.3	connectServer	24
5.9.2.4	getConfig	24

5.9.2.5	isValid	24
5.9.2.6	process	25
5.9.2.7	sendPacket	25
5.9.2.8	setup	25
5.10	SystemInfo::Cpu::Times Struct Reference	25
5.10.1	Detailed Description	26
5.11	SystemInfo::FileSystem::Usage Struct Reference	26
5.11.1	Detailed Description	27

Index**28**

Chapter 1

Namespace Index

1.1 Namespace List

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

SystemInfo		
	System information gathering module	7
WinAgent		
	Windows agent module	7

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

SystemInfo::Cpu::Details	12
SystemInfo::FileSystem::Details	13
SystemInfo::DiagnosticMgr	14
SystemInfo::InfoMapper	17
SystemInfo::ModuleInfo	19
SystemInfo::Cpu	11
SystemInfo::FileSystem	16
SystemInfo::Resources	21
WinAgent::Server	23
SystemInfo::Cpu::Times	25
SystemInfo::FileSystem::Usage	26

Chapter 3

Class Index

3.1 Class List

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

SystemInfo::Cpu	
Contains Cpu related information (times, cpu model details)	11
SystemInfo::Cpu::Details	
Represents basic Cpu/core information (vendor, mhz etc.)	12
SystemInfo::FileSystem::Details	
General information about device/disk	13
SystemInfo::DiagnosticMgr	
Singleton manager for information gathering and visualization	14
SystemInfo::FileSystem	
Class contains file system informations	16
SystemInfo::InfoMapper	
Utils for easy Sigar to SystemInfo data mapping	17
SystemInfo::ModuleInfo	
Pure abstract class for module info (Cpu , Memory etc.)	19
SystemInfo::Resources	
Contains current resources status and its limits	21
WinAgent::Server	
Represents server connected to	23
SystemInfo::Cpu::Times	
Represents times that Cpu used on various operations and states	25
SystemInfo::FileSystem::Usage	
Informations about usage of directory/devie	26

Chapter 4

Namespace Documentation

4.1 SystemInfo Namespace Reference

System information gathering module.

Classes

- class [Cpu](#)
Contains [Cpu](#) related information (times, cpu model details).
- class [DiagnosticMgr](#)
Singleton manager for information gathering and visualization.
- class [FileSystem](#)
Class contains file system informations.
- class [InfoMapper](#)
Utils for easy Sigar to [SystemInfo](#) data mapping.
- class [ModuleInfo](#)
Pure abstract class for module info ([Cpu](#), Memory etc.).
- class [Resources](#)
Contains current resources status and its limits.

4.1.1 Detailed Description

System information gathering module.

4.2 WinAgent Namespace Reference

Windows agent module.

Classes

- class [Server](#)
Represents server connected to.

Functions

- int [recvallWin](#) (SOCKET sockfd, void *data, int size, int timeout)
Receive data from given socket.
- uint32_t [getTicks](#) ()
Get time elapsed since Epoch in milliseconds.
- vector< string > [explode](#) (const string &str, const string &delim)
Split given text by delimiter.
- void [usleep](#) (__int64 usec)
holds for usec microseconds

4.2.1 Detailed Description

Windows agent module.

4.2.2 Function Documentation

4.2.2.1 vector< string > WinAgent::explode (const string & str, const string & delim)

Split given text by delimiter.

Parameters

<i>str</i>	Text to explode.
<i>delim</i>	Delimiter.

Returns

Splitted strings.

Definition at line 19 of file kutilsWin.cpp.

4.2.2.2 uint32_t WinAgent::getTicks ()

Get time elapsed since Epoch in milliseconds.

Returns

Time elapsed in milliseconds.

Definition at line 12 of file kutilsWin.cpp.

4.2.2.3 int WinAgent::recvallWin (SOCKET sockfd, void * data, int size, int timeout)

Receive data from given socket.

Parameters

<i>sockfd</i>	Socket descriptor.
<i>data</i>	Pointer to received data.
<i>size</i>	Number of bytes to be read.

<i>timeout</i>	Time to wait for data.
----------------	------------------------

Returns

Size of received data.

Definition at line 11 of file commonWin.cpp.

4.2.2.4 void WinAgent::usleep (__int64 *usec*)

holds for usec microseconds

Parameters

<i>usec</i>	microseconds
-------------	--------------

Definition at line 43 of file kutilsWin.cpp.

Chapter 5

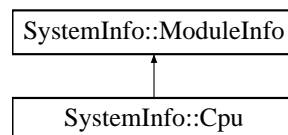
Class Documentation

5.1 SystemInfo::Cpu Class Reference

Contains [Cpu](#) related information (times, cpu model details).

```
#include <Cpu.h>
```

Inheritance diagram for SystemInfo::Cpu:



Classes

- struct [Details](#)
Represents basic Cpu/core information (vendor, mhz etc.).
- struct [Times](#)
Represents times that [Cpu](#) used on various operations and states.

Public Member Functions

- [Cpu](#) ()
Default constructor, initialize general times information and core times / details info.
- void [displayCombinedInfo](#) ()
Display all information about cpu/cores using standard output (std::cout).
- void [displayTimesInfo](#) ([Times](#) *times)
Display information about times using standard output (std::cout).
- void [displayDetailsInfo](#) ([Details](#) *details)
Display information about details using standard output (std::cout).

Public Attributes

- [Times](#) * [generalTimes](#)
[Times](#) combined with all the cores.
- std::vector< [Times](#) * > [coreTimes](#)

Times information per core.

- `std::vector< Details * > coreDetails`

Model *Details* per core.

5.1.1 Detailed Description

Contains [Cpu](#) related information (times, cpu model details).

Definition at line 14 of file Cpu.h.

5.1.2 Member Function Documentation

5.1.2.1 `void SystemInfo::Cpu::displayCombinedInfo () [inline],[virtual]`

Display all information about cpu/cores using standard output (std::cout).

Returns

None

Implements [SystemInfo::ModuleInfo](#).

Definition at line 105 of file Cpu.h.

5.1.2.2 `void SystemInfo::Cpu::displayDetailsInfo (Details * details) [inline]`

Display information about details using standard output (std::cout).

Returns

None

Definition at line 147 of file Cpu.h.

5.1.2.3 `void SystemInfo::Cpu::displayTimesInfo (Times * times) [inline]`

Display information about times using standard output (std::cout).

Returns

None

Definition at line 129 of file Cpu.h.

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

- `src/SystemInfo/Cpu.h`

5.2 SystemInfo::Cpu::Details Struct Reference

Represents basic Cpu/core information (vendor, mhz etc.).

```
#include <Cpu.h>
```

Public Member Functions

- [Details \(\)](#)
Default constructor, zeroes details variables.

Public Attributes

- std::string [vendor](#)
Vendor name.
- std::string [model](#)
Cpu model.
- uint16 [mhz](#)
Clock rate in MHZ.
- uint64 [cache_size](#)
Cache size in bits.
- uint16 [total_sockets](#)
Total number of available sockets.
- uint16 [total_cores](#)
Total number of available cores.
- uint16 [cores_per_socket](#)
Number of cores per available sockets.

5.2.1 Detailed Description

Represents basic Cpu/core information (vendor, mhz etc.).

Definition at line 69 of file Cpu.h.

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

- src/SystemInfo/Cpu.h

5.3 SystemInfo::FileSystem::Details Struct Reference

General information about device/disk.

```
#include <FileSystem.h>
```

Public Member Functions

- [Details \(\)](#)
Default constructor, initialize general directory/device information.

Public Attributes

- std::string [dir](#)
Directory name.
- std::string [dev](#)
Device name.
- std::string [type](#)
Device type.

- `std::string sys_type`
Device system type.
- `std::string options`
Additional options.
- `std::string fsType`
File system type.
- `uint32 flags`
Additional flags.

5.3.1 Detailed Description

General information about device/disk.

Definition at line 28 of file `FileSystem.h`.

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

- `src/SystemInfo/FileSystem.h`

5.4 SystemInfo::DiagnosticMgr Class Reference

Singleton manager for information gathering and visualization.

```
#include <DiagnosticMgr.h>
```

Public Member Functions

- `Resources * getResourcesInfo ()`
Get informations about resources used and its limits.
- `Cpu * getCpuInfo ()`
Get informations about CPUs/Cores.
- `FileSystem * getFileSystemInfo ()`
Get informations about file system (HDD, virtual, external drives).

Static Public Member Functions

- `static double getCpuTemp ()`
Get CPU temepature.
- `static DiagnosticMgr & getInstance ()`
*Get instance of *DiagnosticMgr* singleton class.*

5.4.1 Detailed Description

Singleton manager for information gathering and visualization.

Simple wrapper around Sigar and WMI libraries.

Definition at line 20 of file `DiagnosticMgr.h`.

5.4.2 Member Function Documentation

5.4.2.1 `Cpu * DiagnosticMgr::getCpuInfo ()`

Get informations about CPUs/Cores.

Returns

[Cpu](#) and cores informations.

Definition at line 43 of file DiagnosticMgr.cpp.

5.4.2.2 `double SystemInfo::DiagnosticMgr::getCpuTemp () [static]`

Get CPU temeperature.

Returns

Current CPU temperature in celsius degree.

Definition at line 119 of file DiagnosticMgr.cpp.

5.4.2.3 `FileSystem * SystemInfo::DiagnosticMgr::getFileSystemInfo ()`

Get informations about file system (HDD, virtual, external drives).

Returns

[FileSystem](#) informations.

Definition at line 79 of file DiagnosticMgr.cpp.

5.4.2.4 `static DiagnosticMgr& SystemInfo::DiagnosticMgr::getInstance () [inline],[static]`

Get instance of [DiagnosticMgr](#) singleton class.

Returns

Instance of [DiagnosticMgr](#).

Definition at line 48 of file DiagnosticMgr.h.

5.4.2.5 `Resources * DiagnosticMgr::getResourcesInfo ()`

Get informations about resources used and its limits.

Returns

[Resources](#) informations.

Definition at line 28 of file DiagnosticMgr.cpp.

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

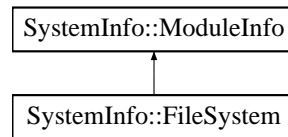
- src/SystemInfo/DiagnosticMgr.h
- src/SystemInfo/DiagnosticMgr.cpp

5.5 SystemInfo::FileSystem Class Reference

Class contains file system informations.

```
#include <FileSystem.h>
```

Inheritance diagram for SystemInfo::FileSystem:



Classes

- struct [Details](#)
General information about device/disk.
- struct [Usage](#)
Informations about usage of directory/devie.

Public Member Functions

- [FileSystem](#) ()
Default constructor, initialize sub-structures for details/usage information.
- void [displayCombinedInfo](#) ()
Display all informations about file system on standard output (std::cout)
- void [displayDetailsInfo](#) ([Details](#) *details)
Display informations from [Details](#) struct over standard output (std::cout)
- void [displayUsageInfo](#) ([Usage](#) *usage)
Display informations from [Usage](#) struct over standard output (std::cout)

Static Public Member Functions

- static std::string [getFullFsTypeName](#) (sigar_file_system_type_e type)
Map file system type to human-readable string.

Public Attributes

- std::vector< [Details](#) * > [dirDetails](#)
List of details per directory.
- std::vector< [Usage](#) * > [dirUsages](#)
List with usage information for every disc.

5.5.1 Detailed Description

Class contains file system informations.

Definition at line 14 of file FileSystem.h.

5.5.2 Member Function Documentation

5.5.2.1 void SystemInfo::FileSystem::displayCombinedInfo () [inline], [virtual]

Display all informations about file system on standard output (std::cout)

Returns

None

Implements [SystemInfo::ModuleInfo](#).

Definition at line 140 of file FileSystem.h.

5.5.2.2 void SystemInfo::FileSystem::displayDetailsInfo (Details * details) [inline]

Display informations from [Details](#) struct over standard output (std::cout)

Returns

None

Definition at line 160 of file FileSystem.h.

5.5.2.3 void SystemInfo::FileSystem::displayUsageInfo (Usage * usage) [inline]

Display informations from [Usage](#) struct over standard output (std::cout)

Returns

None

Definition at line 176 of file FileSystem.h.

5.5.2.4 SystemInfo::FileSystem::getFullFsTypeName (sigar_file_system_type_e type) [inline], [static]

Map file system type to human-readable string.

Returns

Human-readable file system type

Definition at line 122 of file FileSystem.h.

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

- src/SystemInfo/FileSystem.h

5.6 SystemInfo::InfoMapper Class Reference

Utils for easy Sigar to [SystemInfo](#) data mapping.

```
#include <InfoMapper.h>
```

Static Public Member Functions

- static [Cpu::Times](#) * [sigarCpuToCpuTimes](#) (sigar_cpu_t *cpuData)
Mapping cpu times data.
- static [Cpu::Details](#) * [sigarCpuInfoToDetails](#) (sigar_cpu_info_t *cpuData)
Mapping cpu info data.
- static [Resources](#) * [sigarResourcesLimitToResources](#) (sigar_resource_limit_t *resData)
Mapping resources limits data.
- static [FileSystem::Details](#) * [sigarFileSystemToFsDetails](#) (sigar_file_system_t *fsData)
Mapping file system data.
- static [FileSystem::Usage](#) * [sigarFileSystemUsageToFsUsage](#) (sigar_file_system_usage_t *fsData, std::string dirName)
Mapping file system usage data.

5.6.1 Detailed Description

Utils for easy Sigar to [SystemInfo](#) data mapping.

Definition at line 14 of file InfoMapper.h.

5.6.2 Member Function Documentation

5.6.2.1 static [Cpu::Details](#) * [SystemInfo::InfoMapper::sigarCpuInfoToDetails](#) (sigar_cpu_info_t * *cpuData*) [static]

Mapping cpu info data.

Parameters

<i>cpuData</i>	Sigar Cpu info data.
----------------	--------------------------------------

Returns

[Cpu::Details](#) data.

Definition at line 22 of file InfoMapper.cpp.

5.6.2.2 static [Cpu::Times](#) * [SystemInfo::InfoMapper::sigarCpuToCpuTimes](#) (sigar_cpu_t * *cpuData*) [static]

Mapping cpu times data.

Parameters

<i>cpuData</i>	Sigar Cpu times data.
----------------	---------------------------------------

Returns

[Cpu::Times](#) data.

Definition at line 5 of file InfoMapper.cpp.

5.6.2.3 [FileSystem::Details](#) * [SystemInfo::InfoMapper::sigarFileSystemToFsDetails](#) (sigar_file_system_t * *fsData*)
[static]

Mapping file system data.

Parameters

<i>fsData</i>	Sigar file system data.
---------------	-------------------------

Returns

[FileSystem::Details](#) data.

Definition at line 65 of file InfoMapper.cpp.

5.6.2.4 `FileSystem::Usage * SystemInfo::InfoMapper::sigarFileSystemUsageToFsUsage (sigar_file_system_usage_t * fsData, std::string dirName) [static]`

Mapping file system usage data.

Parameters

<i>fsData</i>	Sigar file system usage data.
<i>dirName</i>	Directory name.

Returns

[FileSystem::Usage](#) data.

Definition at line 80 of file InfoMapper.cpp.

5.6.2.5 `static Resources * SystemInfo::InfoMapper::sigarResourcesLimitToResources (sigar_resource_limit_t * resData) [static]`

Mapping resources limits data.

Parameters

<i>resData</i>	Sigar resources data.
----------------	-----------------------

Returns

[Resources](#) data.

Definition at line 37 of file InfoMapper.cpp.

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

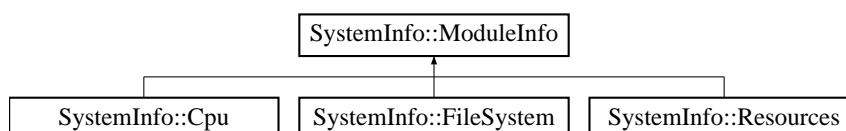
- src/SystemInfo/InfoMapper.h
- src/SystemInfo/InfoMapper.cpp

5.7 SystemInfo::ModuleInfo Class Reference

Pure abstract class for module info ([Cpu](#), Memory etc.).

```
#include <ModuleInfo.h>
```

Inheritance diagram for SystemInfo::ModuleInfo:



Public Member Functions

- virtual void [displayCombinedInfo](#) ()=0
Display full informations from module instance.
- void [displayModuleHeader](#) (std::string name)
Display given module header using standard output (std::cout).
- void [displayGroupHeader](#) (std::string name)
Display given group header using standard output (std::cout).
- template<typename T >
void [displaySingleInfo](#) (std::string description, T value)
Display single information as pair of label and parameterized value.
- template<typename T , typename V >
void [displayPairInfo](#) (std::string description, T current, V max)
Display signle information as pair of label and parameterized pair value of current/max status.

5.7.1 Detailed Description

Pure abstract class for module info ([Cpu](#), Memory etc.).

Provides method for easy displaying / forming system informations.

Definition at line 14 of file ModuleInfo.h.

5.7.2 Member Function Documentation

5.7.2.1 void SystemInfo::ModuleInfo::displayCombinedInfo () [pure virtual]

Display full informations from module instance.

Returns

None.

Implemented in [SystemInfo::FileSystem](#), [SystemInfo::Cpu](#), and [SystemInfo::Resources](#).

5.7.2.2 void SystemInfo::ModuleInfo::displayGroupHeader (std::string name) [inline]

Display given group header using standard output (std::cout).

Parameters

<i>name</i>	Group header to display.
-------------	--------------------------

Returns

None.

Definition at line 41 of file ModuleInfo.h.

5.7.2.3 void SystemInfo::ModuleInfo::displayModuleHeader (std::string name) [inline]

Display given module header using standard output (std::cout).

Parameters

<i>name</i>	Header to display.
-------------	--------------------

Returns

None.

Definition at line 31 of file ModuleInfo.h.

5.7.2.4 `template<typename T , typename V > template< typename T, typename V > void SystemInfo::ModuleInfo::displayPairInfo (std::string description, T current, V max) [inline]`

Display single information as pair of label and parameterized pair value of current/max status.

Parameters

<i>description</i>	Description label for value.
<i>current</i>	Parameterized current value to display.
<i>max</i>	Parameterized maximum value to display.

Returns

None.

Definition at line 67 of file ModuleInfo.h.

5.7.2.5 `template<typename T > template< typename T > void SystemInfo::ModuleInfo::displaySingleInfo (std::string description, T value) [inline]`

Display single information as pair of label and parameterized value.

Parameters

<i>description</i>	Description label for value.
<i>value</i>	Parameterized value to display.

Returns

None;

Definition at line 53 of file ModuleInfo.h.

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

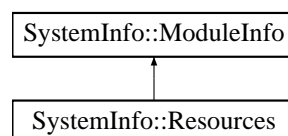
- src/SystemInfo/ModuleInfo.h

5.8 SystemInfo::Resources Class Reference

Contains current resources status and its limits.

```
#include <Resources.h>
```

Inheritance diagram for SystemInfo::Resources:



Public Member Functions

- [Resources](#) ()
Default constructor, zeroes resource variables.
- void [displayCombinedInfo](#) ()
Display all informations about resources using standard output (std::cout).

Public Attributes

- uint64 [cpuCurrent](#)
Current cpu resource.
- uint64 [cpuMax](#)
Limit of cpu resource.
- uint64 [fileSizeCurrent](#)
Current file size.
- uint64 [fileSizeMax](#)
Limit of file size.
- uint64 [pipeSizeCurrent](#)
Current pipe buffer size.
- uint64 [pipeSizeMax](#)
Limit of pipe buffer size.
- uint64 [dataCurrent](#)
Current data size.
- uint64 [dataMax](#)
Limit of data size.
- uint64 [stackCurrent](#)
Current stack use.
- uint64 [stackMax](#)
Limit of use of stack.
- uint64 [coreCurrent](#)
Current core resource.
- uint64 [coreMax](#)
Limit of core resource.
- uint64 [memoryCurrent](#)
Current memory use.
- uint64 [memoryMax](#)
Limit of memory use.
- uint64 [processesCurrent](#)
Current processes resource.
- uint64 [processesMax](#)
Limit of processes resource.
- uint64 [openFilesCurrent](#)
Current opened files resource.
- uint64 [openFilesMax](#)
Limit of opened files resource at once.
- uint64 [virtualMemoryCurrent](#)
Current virtual memory resource.
- uint64 [virtualMemoryMax](#)
Limit of virtual memory resource.

5.8.1 Detailed Description

Contains current resources status and its limits.

Definition at line 12 of file Resources.h.

5.8.2 Member Function Documentation

5.8.2.1 void SystemInfo::Resources::displayCombinedInfo () [inline], [virtual]

Dispaly all informations about resources using standard output (std::cout).

Returns

None

Implements [SystemInfo::ModuleInfo](#).

Definition at line 75 of file Resources.h.

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

- src/SystemInfo/Resources.h

5.9 WinAgent::Server Class Reference

Represents server connected to.

```
#include <serverWin.h>
```

Public Member Functions

- [Server](#) ()
Constructor, zeroes data.
- [~Server](#) ()
Destructor, do nothing.
- void [setup](#) (const string &host, int port, const string &key)
Set server connection data and authentication key.
- void [process](#) ()
Process connection events.
- bool [isValid](#) ()
Check if connection is valid (agent connected to server).
- bool [configChanged](#) ()
Check if configuration change is needed.
- void [configApplied](#) ()
Set configuration change flag if new config applied.
- bool [sendPacket](#) (IPacket &packet)
Send given packet to server.
- void [connectServer](#) ()
Connect to server.
- TPacketConfig & [getConfig](#) ()
Get current agent-server configuration.

5.9.1 Detailed Description

Represents server connected to.

Definition at line 27 of file serverWin.h.

5.9.2 Member Function Documentation

5.9.2.1 void WinAgent::Server::configApplied () [inline]

Set configuration change flag if new config applied.

Returns

None.

Definition at line 73 of file serverWin.h.

5.9.2.2 bool WinAgent::Server::configChanged () [inline]

Check if configuration change is needed.

Returns

If change needed.

Definition at line 67 of file serverWin.h.

5.9.2.3 void WinAgent::Server::connectServer ()

Connect to server.

Returns

None.

Definition at line 98 of file serverWin.cpp.

5.9.2.4 TPacketConfig & WinAgent::Server::getConfig () [inline]

Get current agent-server configuration.

Returns

Current configuration.

Definition at line 93 of file serverWin.h.

5.9.2.5 bool WinAgent::Server::isValid () [inline]

Check if connection is valid (agent connected to server).

Returns

If connected.

Definition at line 61 of file serverWin.h.

5.9.2.6 void WinAgent::Server::process ()

Process connection events.

Returns

None.

Definition at line 27 of file serverWin.cpp.

5.9.2.7 bool WinAgent::Server::sendPacket (IPacket & packet)

Send given packet to server.

Parameters

<i>packet</i>	Packet to send.
---------------	-----------------

Returns

If succeeded.

Definition at line 186 of file serverWin.cpp.

5.9.2.8 void WinAgent::Server::setup (const string & host, int port, const string & key)

Set server connection data and authentication key.

Parameters

<i>host</i>	Host address.
<i>port</i>	Host port.
<i>key</i>	Authentication key.

Returns

None.

Definition at line 21 of file serverWin.cpp.

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

- src/serverWin.h
- src/serverWin.cpp

5.10 SystemInfo::Cpu::Times Struct Reference

Represents times that [Cpu](#) used on various operations and states.

```
#include <Cpu.h>
```

Public Member Functions

- [Times](#) ()

Default constructor, zeroes times variables.

Public Attributes

- uint64 [user](#)
Time spent outside kernel.
- uint64 [sys](#)
Time spent inside kernel.
- uint64 [nice](#)
Nice time.
- uint64 [idle](#)
Time spent idleing, without jobs.
- uint64 [wait](#)
Time spent waiting for other tasks.
- uint64 [irq](#)
Time spent on handling hardware irq (interrupt).
- uint64 [soft_irq](#)
Time spent on handling software irq (interrupt).
- uint64 [stolen](#)
Time stolen.
- uint64 [total](#)
Total time spent on every tasks.

5.10.1 Detailed Description

Represents times that [Cpu](#) used on various operations and states.

Definition at line 30 of file Cpu.h.

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

- src/SystemInfo/Cpu.h

5.11 SystemInfo::FileSystem::Usage Struct Reference

Informations about usage of directory/devie.

```
#include <FileSystem.h>
```

Public Member Functions

- [Usage](#) ()
Default constructor, zeroing usage informations.

Public Attributes

- std::string [dir](#)
Directory/disc name for usage informations.
- double [percent_use](#)
Percent use of directory/disc.
- uint64 [total](#)
Total memory assigned to disc.
- uint64 [free](#)

- Free memory on disc.*
- uint64 [used](#)
- Memory used on disc.*
- uint64 [avail](#)
- Memory available on disc.*
- uint64 [files](#)
- Memory used by files.*
- uint64 [free_files](#)
- Memory not used by files.*
- uint64 [disc_reads](#)
- Number of disc reads operation.*
- uint64 [disc_writes](#)
- Number of disc writes operation.*
- uint64 [disc_write_bytes](#)
- Number of bytes writed.*
- uint64 [disc_read_bytes](#)
- Number of bytes readed.*
- uint64 [disc_rtime](#)
- Time spent on read.*
- uint64 [disc_wtime](#)
- Time spent on write.*
- uint64 [disc_qtime](#)
- Time spent in queue.*
- uint64 [disc_time](#)
- Total time.*
- uint64 [disc_snaptime](#)
- Snap time.*
- double [disc_service_time](#)
- Disc service time.*
- double [disc_queue](#)
- Number of disc queues.*

5.11.1 Detailed Description

Informations about usage of directory/devie.

Definition at line 60 of file FileSystem.h.

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

- src/SystemInfo/FileSystem.h

Index

- configApplied
 - WinAgent::Server, [24](#)
- configChanged
 - WinAgent::Server, [24](#)
- connectServer
 - WinAgent::Server, [24](#)
- displayCombinedInfo
 - SystemInfo::Cpu, [12](#)
 - SystemInfo::FileSystem, [17](#)
 - SystemInfo::ModuleInfo, [20](#)
 - SystemInfo::Resources, [23](#)
- displayDetailsInfo
 - SystemInfo::Cpu, [12](#)
 - SystemInfo::FileSystem, [17](#)
- displayGroupHeader
 - SystemInfo::ModuleInfo, [20](#)
- displayModuleHeader
 - SystemInfo::ModuleInfo, [20](#)
- displayPairInfo
 - SystemInfo::ModuleInfo, [21](#)
- displaySingleInfo
 - SystemInfo::ModuleInfo, [21](#)
- displayTimesInfo
 - SystemInfo::Cpu, [12](#)
- displayUsageInfo
 - SystemInfo::FileSystem, [17](#)
- explode
 - WinAgent, [8](#)
- getConfig
 - WinAgent::Server, [24](#)
- getCpuInfo
 - SystemInfo::DiagnosticMgr, [15](#)
- getCpuTemp
 - SystemInfo::DiagnosticMgr, [15](#)
- getFileSystemInfo
 - SystemInfo::DiagnosticMgr, [15](#)
- getFullFsTypeName
 - SystemInfo::FileSystem, [17](#)
- getInstance
 - SystemInfo::DiagnosticMgr, [15](#)
- getResourcesInfo
 - SystemInfo::DiagnosticMgr, [15](#)
- getTicks
 - WinAgent, [8](#)
- isValid
 - WinAgent::Server, [24](#)

- process
 - WinAgent::Server, [24](#)
- recvallWin
 - WinAgent, [8](#)
- sendPacket
 - WinAgent::Server, [25](#)
- setup
 - WinAgent::Server, [25](#)
- sigarCpuInfoToDetails
 - SystemInfo::InfoMapper, [18](#)
- sigarCpuToCpuTimes
 - SystemInfo::InfoMapper, [18](#)
- sigarFileSystemToFsDetails
 - SystemInfo::InfoMapper, [18](#)
- sigarFileSystemUsageToFsUsage
 - SystemInfo::InfoMapper, [19](#)
- sigarResourcesLimitToResources
 - SystemInfo::InfoMapper, [19](#)
- SystemInfo, [7](#)
- SystemInfo::Cpu, [11](#)
 - displayCombinedInfo, [12](#)
 - displayDetailsInfo, [12](#)
 - displayTimesInfo, [12](#)
- SystemInfo::Cpu::Details, [12](#)
- SystemInfo::Cpu::Times, [25](#)
- SystemInfo::DiagnosticMgr, [14](#)
 - getCpuInfo, [15](#)
 - getCpuTemp, [15](#)
 - getFileSystemInfo, [15](#)
 - getInstance, [15](#)
 - getResourcesInfo, [15](#)
- SystemInfo::FileSystem, [16](#)
 - displayCombinedInfo, [17](#)
 - displayDetailsInfo, [17](#)
 - displayUsageInfo, [17](#)
 - getFullFsTypeName, [17](#)
- SystemInfo::FileSystem::Details, [13](#)
- SystemInfo::FileSystem::Usage, [26](#)
- SystemInfo::InfoMapper, [17](#)
 - sigarCpuInfoToDetails, [18](#)
 - sigarCpuToCpuTimes, [18](#)
 - sigarFileSystemToFsDetails, [18](#)
 - sigarFileSystemUsageToFsUsage, [19](#)
 - sigarResourcesLimitToResources, [19](#)
- SystemInfo::ModuleInfo, [19](#)
 - displayCombinedInfo, [20](#)
 - displayGroupHeader, [20](#)
 - displayModuleHeader, [20](#)

- displayPairInfo, [21](#)
- displaySingleInfo, [21](#)
- SystemInfo::Resources, [21](#)
 - displayCombinedInfo, [23](#)
- usleep
 - WinAgent, [9](#)
- WinAgent, [7](#)
 - explode, [8](#)
 - getTicks, [8](#)
 - recvallWin, [8](#)
 - usleep, [9](#)
- WinAgent::Server, [23](#)
 - configApplied, [24](#)
 - configChanged, [24](#)
 - connectServer, [24](#)
 - getConfig, [24](#)
 - isValid, [24](#)
 - process, [24](#)
 - sendPacket, [25](#)
 - setup, [25](#)