ZPI_Common

Generated by Doxygen 1.8.5

Tue Nov 19 2013 21:52:58

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

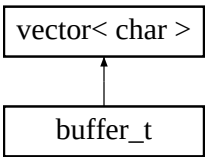
Class Documentation

3.1 buffer_t Class Reference

Byte buffer used for marshalling.

`#include <common.h>`

Inheritance diagram for buffer_t:



Public Member Functions

- template<typename T >
  bool append (const T &val)

  Append single data to buffer.
- template<typename T >
  bool append (const vector< T > &array)

  Append vector of data to buffer.
- template<typename T >
  bool fetch (T &val)

  Fetch parametrized data from buffer last position.
- bool fetch (string &val)

  Fetch string from buffer last position.
- void rewind ()

  Reset internal pointer.

3.1.1 Detailed Description

Byte buffer used for marshalling.

Implemented as stl vector of chars.

Definition at line 19 of file common.h.

3.1.2 Member Function Documentation

3.1.2.1 template<typename T > template< typename T > bool buffer_t::append (const T & *val*) [inline]

Append single data to buffer.

Parameters

val	Data to append.

Returns

If succeeded.

Definition at line 29 of file common.h.

3.1.2.2 template<typename T > template< typename T > bool buffer_t::append (const vector< T > & *array*) [inline]

Append vector of data to buffer.

Parameters

array	Data to append.

Returns

If succeeded.

Definition at line 41 of file common.h.

3.1.2.3 template<typename T > template< typename T > bool buffer_t::fetch (T & *val*) [inline]

Fetch parametrized data from buffer last position.

Parameters

out	*val*	Fetched value.

Returns

If succeeded.

Definition at line 60 of file common.h.

3.1.2.4 bool buffer_t::fetch (string & *val*) [inline]

Fetch string from buffer last position.

Parameters

out	*val*	Fetched string.

Returns

If succeeded.

Definition at line 74 of file common.h.

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

- common.h

3.2 Config Class Reference

Simple handler for configuration files.

`#include <config.h>`

Public Member Functions

- Config ()

  Default constructor, do nothing.
- ~Config ()

  Destructor, do nothing.
- bool fromFile (const string &path)

  Load configuration from given file.
- bool saveToFile (const string &path)

  Save configuration to given file.
- bool hasKey (const string &key)

  Check if configuration contains given key.
- string getString (const string &key, const string &def="")

  Get string value connected with given key.
- int getInt (const string &key, int def=0)

  Get int value connected with given key.
- void setString (const string &key, const string &value)

  Insert string value and assign it with given key.
- void setInt (const string &key, int val)

  Insert int value and assign it with given key.
- bool **fromFile** (const string &path)
- bool **saveToFile** (const string &path)
- bool **hasKey** (const string &key)
- string **getString** (const string &key, const string &def="")
- int **getInt** (const string &key, int def=0)
- void **setString** (const string &key, const string &value)
- void **setInt** (const string &key, int val)

3.2.1 Detailed Description

Simple handler for configuration files.

Definition at line 12 of file config.h.

3.2.2 Member Function Documentation

3.2.2.1 bool Config::fromFile (const string & *path*)

Load configuration from given file.

Parameters

path	Path pointing to configuration file.

Returns

If succeeded.

Definition at line 14 of file config.cpp.

3.2.2.2 int Config::getInt (const string & *key*, int *def* = 0)

Get int value connected with given key.

Parameters

key	Key connected with wanted value.
def	Default value in case of absence of given key.

Returns

Value connected with key or default value def otherwise.

Definition at line 74 of file config.cpp.

3.2.2.3 string Config::getString (const string & *key*, const string & *def* = " ")

Get string value connected with given key.

Parameters

key	Key connected with wanted value.
def	Default value in case of absence of given key.

Returns

Value connected with key or default value def otherwise.

Definition at line 67 of file config.cpp.

3.2.2.4 bool Config::hasKey (const string & *key*)

Check if configuration contains given key.

Parameters

key	Key to check.

Returns

If contained.

Definition at line 63 of file config.cpp.

3.2.2.5 bool Config::saveToFile (const string & *path*)

Save configuration to given file.

Parameters

path	Path pointing to configuration file.

Returns

If succeeded.

Definition at line 41 of file config.cpp.

3.2.2.6 void Config::setInt (const string & *key*, int *val*)

Insert int value and assign it with given key.

Parameters

key	Key connected with inserted value.
val	Inserted value.

Returns

None.

Parameters

key	Key connected with inserted value.
value	Inserted value.

Returns

None.

Definition at line 89 of file config.cpp.

3.2.2.7 void Config::setString (const string & *key*, const string & *value*)

Insert string value and assign it with given key.

Parameters

key	Key connected with inserted value.
value	Inserted value.

Returns

None.

Definition at line 85 of file config.cpp.

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

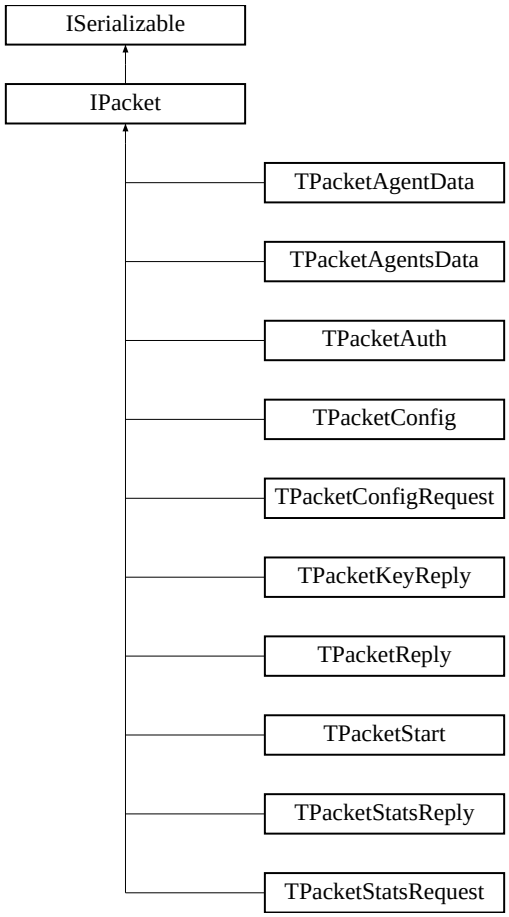
- config.h
- configWin.h
- config.cpp
- configWin.cpp

3.3 IPacket Class Reference

Packet data base class.

`#include <packets.h>`

Inheritance diagram for IPacket:



Public Member Functions

- virtual int getType ()=0
  Get packet type.

Additional Inherited Members

3.3.1 Detailed Description

Packet data base class.

Definition at line 55 of file packets.h.

3.3.2 Member Function Documentation

3.3.2.1 int IPacket::getType ( ) [pure virtual]

Get packet type.

Returns

Packet type.

Implemented in TPacketStatsReply, TPacketStatsRequest, TPacketConfigRequest, TPacketConfig, TPacketKeyReply, TPacketAgentsData, TPacketAgentData, TPacketStart, TPacketReply, and TPacketAuth.

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

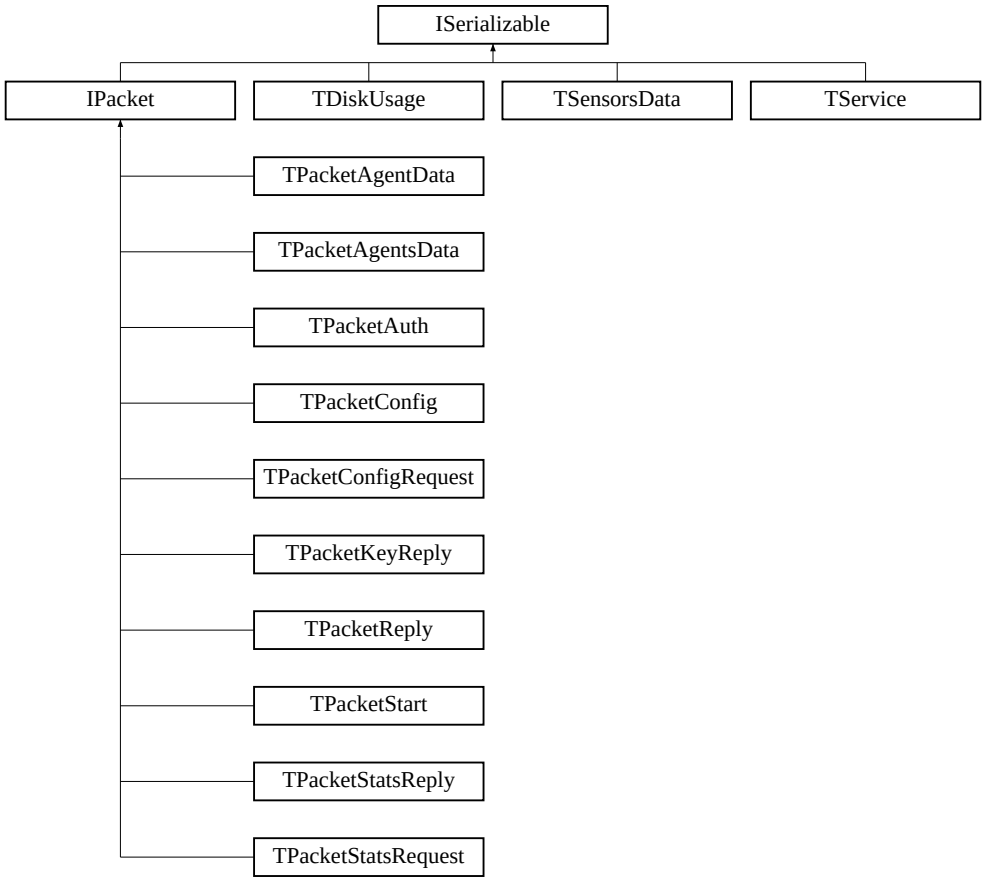
- packets.h

3.4 ISerializable Interface Reference

Object able to serialize himself into/from buffer_t class.

`#include <common.h>`

Inheritance diagram for ISerializable:



Public Member Functions

- virtual void toBuffer (buffer_t &buf)=0

  Convert object to byte buffer.
- virtual bool fromBuffer (buffer_t &buf)=0

  Fill object data with given buffer.
- template<typename T >
  bool append (buffer_t &buf, T &val)

  Append parametrized value to end of buffer.
- bool append (buffer_t &buf, string &val)

  Append string to end of buffer.
- template<typename T >
  bool fetch (buffer_t &buf, T &val)

  Fetch parametrized data from buffer last position.
- bool fetch (buffer_t &buf, string &val)

  Fetch string from buffer last position.

Protected Attributes

- int m_pos

  Buffer cursor.

3.4.1 Detailed Description

Object able to serialize himself into/from buffer_t class.

Definition at line 139 of file common.h.

3.4.2 Member Function Documentation

3.4.2.1 template<typename T > template< typename T > bool ISerializable::append (buffer_t & *buf*, T & *val*) [inline]

Append parametrized value to end of buffer.

Parameters

out	*buf*	Target buffer.
in	*val*	Value to append.

Returns

If succeeded.

Definition at line 165 of file common.h.

3.4.2.2 bool ISerializable::append (buffer_t & *buf*, string & *val*) [inline]

Append string to end of buffer.

Parameters

out	*buf*	Target buffer.
in	*val*	String to append.

Returns

If succeeded.

Definition at line 177 of file common.h.

3.4.2.3 template<typename T > template< typename T > bool ISerializable::fetch (buffer_t & *buf*, T & *val*) [inline]

Fetch parametrized data from buffer last position.

Parameters

in	*buf*	Source buffer.
out	*val*	Fetched value.

Returns

If succeeded.

Definition at line 194 of file common.h.

3.4.2.4 bool ISerializable::fetch (buffer_t & *buf,* string & *val*) [inline]

Fetch string from buffer last position.

Parameters

in	*buf*	Source buffer.
out	*val*	Fetched string.

Returns

If succeeded.

Definition at line 209 of file common.h.

3.4.2.5 bool ISerializable::fromBuffer (buffer_t & *buf*) [pure virtual]

Fill object data with given buffer.

Parameters

in	*buf*	Buffer with object data.

Returns

If succeeded.

Implemented in TPacketStatsReply, TPacketStatsRequest, TPacketConfigRequest, TPacketConfig, TPacketKeyReply, TPacketAgentsData, TPacketAgentData, TPacketStart, TPacketReply, TSensorsData, TPacketAuth, TService, and TDiskUsage.

3.4.2.6 void ISerializable::toBuffer (buffer_t & *buf*) [pure virtual]

Convert object to byte buffer.

Parameters

out	*buf*	Byte buffer representation of object.

Returns

None.

Implemented in TPacketStatsReply, TPacketStatsRequest, TPacketConfigRequest, TPacketConfig, TPacketKeyReply, TPacketAgentsData, TPacketAgentData, TPacketStart, TPacketReply, TSensorsData, TPacketAuth, TService, and TDiskUsage.

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

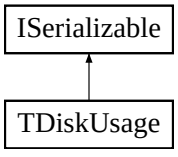
- common.h

3.5 TDiskUsage Class Reference

Information about disc usage.

`#include <sensors.h>`

Inheritance diagram for TDiskUsage:



Public Member Functions

- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- string name

  Disc name.
- uint64_t totalSpace

  Total available space on disc.
- uint64_t usedSpace

  Used space on disc.

Additional Inherited Members

3.5.1 Detailed Description

Information about disc usage.

Definition at line 18 of file sensors.h.

3.5.2 Member Function Documentation

3.5.2.1 bool TDiskUsage::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 46 of file sensors.h.

3.5.2.2 void TDiskUsage::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 34 of file sensors.h.

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

- sensors.h

3.6 THeader Struct Reference

Packet header.

`#include <packets.h>`

Public Attributes

- uint8_t type
  Packet type.
- uint16_t size
  Packet size.

3.6.1 Detailed Description

Packet header.

Definition at line 71 of file packets.h.

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

- packets.h

3.7 Timer Class Reference

Counts steps between the time interval.

`#include <kutils.h>`

Public Member Functions

- Timer ()
  Default constructor, zeroes time interval.
- void setInterval (uint32_t interval)
  Set interval to count to given value.
- bool process ()
  Check if interval passed.

3.7.1 Detailed Description

Counts steps between the time interval.

Definition at line 44 of file kutils.h.

3.7.2 Member Function Documentation

3.7.2.1 bool Timer::process ( ) [inline]

Check if interval passed.

Returns

If passed.

Definition at line 69 of file kutils.h.

3.7.2.2 void Timer::setInterval (uint32_t *interval*) [inline]

Set interval to count to given value.

Parameters

interval	Time interval.

Returns

None.

Definition at line 59 of file kutils.h.

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

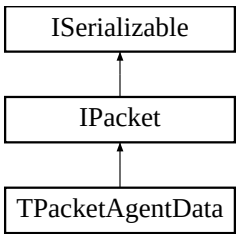
- kutils.h

3.8 TPacketAgentData Class Reference

Data from agent.

`#include <packets.h>`

Inheritance diagram for TPacketAgentData:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- uint16_t id

  Packet id.
- uint8_t oldData

  Old data.
- string name

  Agent name.
- TSensorsData data

  Sensors data.

Additional Inherited Members

3.8.1 Detailed Description

Data from agent.

Definition at line 208 of file packets.h.

3.8.2 Member Function Documentation

3.8.2.1 bool TPacketAgentData::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 247 of file packets.h.

3.8.2.2 int TPacketAgentData::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_AGENTDATA.

Implements IPacket.

Definition at line 225 of file packets.h.

3.8.2.3 void TPacketAgentData::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 232 of file packets.h.

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

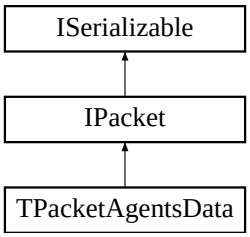
- packets.h

3.9 TPacketAgentsData Class Reference

Combined agents data.

`#include <packets.h>`

Inheritance diagram for TPacketAgentsData:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- vector< TPacketAgentData > agents

  Agents data.

Additional Inherited Members

3.9.1 Detailed Description

Combined agents data.

Definition at line 262 of file packets.h.

3.9.2 Member Function Documentation

3.9.2.1 bool TPacketAgentsData::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 297 of file packets.h.

3.9.2.2 int TPacketAgentsData::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_AGENTSDATA.

Implements IPacket.

Definition at line 273 of file packets.h.

3.9.2.3 void TPacketAgentsData::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 280 of file packets.h.

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

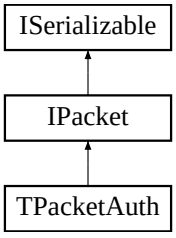
- packets.h

3.10 TPacketAuth Class Reference

Authentication packet.

`#include <packets.h>`

Inheritance diagram for TPacketAuth:



Public Member Functions

- virtual int getType ()
  Get packet type.
- virtual void toBuffer (buffer_t &buf)
  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)
  Fill object with buffer data.

Public Attributes

- char key [16]
  Key.
- uint8_t sendConfig
  Config data.

Additional Inherited Members

3.10.1 Detailed Description

Authentication packet.

Definition at line 84 of file packets.h.

3.10.2 Member Function Documentation

3.10.2.1 bool TPacketAuth::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 115 of file packets.h.

3.10.2.2 int TPacketAuth::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_AUTH.

Implements IPacket.

Definition at line 97 of file packets.h.

3.10.2.3 void TPacketAuth::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 104 of file packets.h.

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

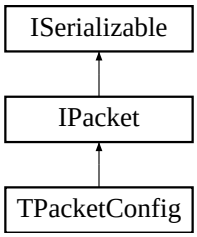
- packets.h

3.11 TPacketConfig Class Reference

Agent configuration.

`#include <packets.h>`

Inheritance diagram for TPacketConfig:



Classes

- struct TService

  Networ service data.

Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- uint16_t agentId

  Agent id.
- string tempPath

  Temperature path.
- uint16_t tempDivider

Temperature divider.

- vector< TPacketConfig::TService > services

  Agent services.

- uint16_t interval

  Time interval.

- string name

  Agent name.

Additional Inherited Members

3.11.1 Detailed Description

Agent configuration.

Definition at line 355 of file packets.h.

3.11.2 Member Function Documentation

3.11.2.1 bool TPacketConfig::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 418 of file packets.h.

3.11.2.2 int TPacketConfig::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_CONFIG.

Implements IPacket.

Definition at line 389 of file packets.h.

3.11.2.3 void TPacketConfig::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 396 of file packets.h.

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

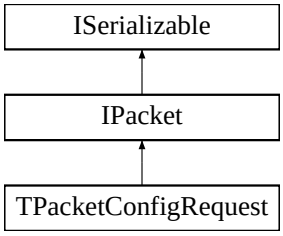
- packets.h

3.12 TPacketConfigRequest Class Reference

Requet for agent configuration.

`#include <packets.h>`

Inheritance diagram for TPacketConfigRequest:



Public Member Functions

- virtual int getType ()
  Get packet type.
- virtual void toBuffer (buffer_t &buf)
  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)
  Fill object with buffer data.

Public Attributes

- uint16_t agentId
  Requested agent id.

Additional Inherited Members

3.12.1 Detailed Description

Requet for agent configuration.

Definition at line 445 of file packets.h.

3.12.2 Member Function Documentation

3.12.2.1 bool TPacketConfigRequest::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 473 of file packets.h.

3.12.2.2 int TPacketConfigRequest::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_CONFIG_REQUEST.

Implements IPacket.

Definition at line 456 of file packets.h.

3.12.2.3 void TPacketConfigRequest::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 463 of file packets.h.

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

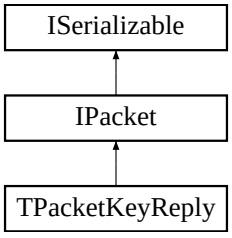
- packets.h

3.13 TPacketKeyReply Class Reference

Packet with authentication key.

`#include <packets.h>`

Inheritance diagram for TPacketKeyReply:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- char key [16]

  Key.

Additional Inherited Members

3.13.1 Detailed Description

Packet with authentication key.

Definition at line 315 of file packets.h.

3.13.2 Member Function Documentation

3.13.2.1 bool TPacketKeyReply::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 343 of file packets.h.

3.13.2.2 int TPacketKeyReply::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_KEY_REPLY.

Implements IPacket.

Definition at line 326 of file packets.h.

3.13.2.3 void TPacketKeyReply::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 333 of file packets.h.

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

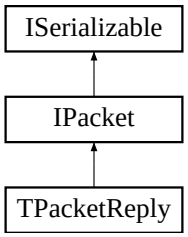
- packets.h

3.14 TPacketReply Class Reference

Reply packet.

`#include <packets.h>`

Inheritance diagram for TPacketReply:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- int value

  Reply value.

Additional Inherited Members

3.14.1 Detailed Description

Reply packet.

Definition at line 128 of file packets.h.

3.14.2 Member Function Documentation

3.14.2.1 bool TPacketReply::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 156 of file packets.h.

3.14.2.2 int TPacketReply::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_REPLY.

Implements IPacket.

Definition at line 139 of file packets.h.

3.14.2.3 void TPacketReply::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 146 of file packets.h.

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

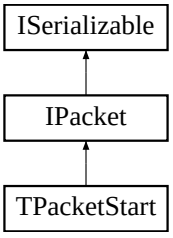
- packets.h

3.15 TPacketStart Class Reference

Starting packet.

`#include <packets.h>`

Inheritance diagram for TPacketStart:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- uint16_t interval

  Time interval.

Additional Inherited Members

3.15.1 Detailed Description

Starting packet.

Definition at line 168 of file packets.h.

3.15.2 Member Function Documentation

3.15.2.1 bool TPacketStart::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 196 of file packets.h.

3.15.2.2 int TPacketStart::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_START.

Implements IPacket.

Definition at line 179 of file packets.h.

3.15.2.3 void TPacketStart::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 186 of file packets.h.

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

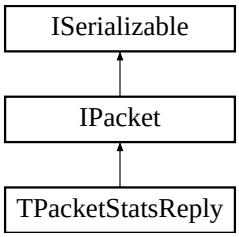
- packets.h

3.16 TPacketStatsReply Class Reference

Reply for stats request.

`#include <packets.h>`

Inheritance diagram for TPacketStatsReply:



Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- vector< float > points

  Given points.

Additional Inherited Members

3.16.1 Detailed Description

Reply for stats request.

Definition at line 552 of file packets.h.

3.16.2 Member Function Documentation

3.16.2.1 bool TPacketStatsReply::fromBuffer (buffer_t & *buf*) [inline],[virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 580 of file packets.h.

3.16.2.2 int TPacketStatsReply::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_STATS_REPLY.

Implements IPacket.

Definition at line 563 of file packets.h.

3.16.2.3 void TPacketStatsReply::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 570 of file packets.h.

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

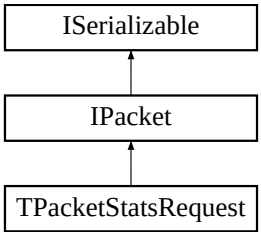
- packets.h

3.17 TPacketStatsRequest Class Reference

Stats request from agent.

`#include <packets.h>`

Inheritance diagram for TPacketStatsRequest:



Public Types

- enum EType { **CPU** = 0, **RAM**, **TEMP**, **DISK** }

  Statistic types based on origin.

Public Member Functions

- virtual int getType ()

  Get packet type.
- virtual void toBuffer (buffer_t &buf)

  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- uint16_t agentId

  Agent id.
- uint32_t startDate

  Start date.
- uint32_t endDate

  End date.
- uint16_t points

  Number of points.
- EType type

  Stats type.
- string diskName

  Disc name.

Additional Inherited Members

3.17.1 Detailed Description

Stats request from agent.

Definition at line 485 of file packets.h.

3.17.2 Member Function Documentation

3.17.2.1 bool TPacketStatsRequest::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 533 of file packets.h.

3.17.2.2 int TPacketStatsRequest::getType ( ) [inline], [virtual]

Get packet type.

Returns

PACKET_STATS_REQUEST.

Implements IPacket.

Definition at line 511 of file packets.h.

3.17.2.3 void TPacketStatsRequest::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 518 of file packets.h.

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

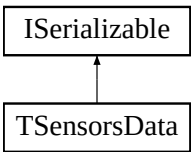
- packets.h

3.18 TSensorsData Class Reference

Data from agent sensors.

`#include <sensors.h>`

Inheritance diagram for TSensorsData:



Public Member Functions

- virtual void toBuffer (buffer_t &buf)
  Get byte buffer representation.
- virtual bool fromBuffer (buffer_t &buf)
  Fill object with buffer data.

Public Attributes

- uint32_t timestamp

Time of data acquisition.

- float temp

  Cpu temperature.

- bool tempValid

  Cpu temperature correctness.

- float cpuUsage

  Cpu percent usage.

- uint64_t totalRam

  Available RAM.

- uint64_t freeRam

  Not used RAM.

- uint32_t uptime

  System uptime.

- vector< TDiskUsage > disksUsage

  List of discs usage.

- vector< TService > services

  List of running services.

Additional Inherited Members

3.18.1 Detailed Description

Data from agent sensors.

Definition at line 91 of file sensors.h.

3.18.2 Member Function Documentation

3.18.2.1 bool TSensorsData::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 153 of file sensors.h.

3.18.2.2 void TSensorsData::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 119 of file sensors.h.

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

- sensors.h

3.19 TPacketConfig::TService Struct Reference

Networ service data.

`#include <packets.h>`

Public Attributes

- string name
  Service name.
- bool tcp
  Service transport type (TCP/UDP)
- uint16_t port
  Service port.

3.19.1 Detailed Description

Networ service data.

Definition at line 362 of file packets.h.

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

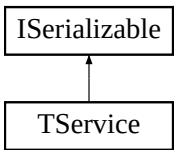
- packets.h

3.20 TService Class Reference

Information about service.

`#include <sensors.h>`

Inheritance diagram for TService:



Public Member Functions

- virtual void toBuffer (buffer_t &buf)
  Get byte buffer representation.

- virtual bool fromBuffer (buffer_t &buf)

  Fill object with buffer data.

Public Attributes

- string name

  Service name.
- bool available

  Service availability.

Additional Inherited Members

3.20.1 Detailed Description

Information about service.

Definition at line 56 of file sensors.h.

3.20.2 Member Function Documentation

3.20.2.1 bool TService::fromBuffer (buffer_t & *buf*) [inline], [virtual]

Fill object with buffer data.

Parameters

in	*buf*	Buffer data.

Returns

If succeeded.

Implements ISerializable.

Definition at line 81 of file sensors.h.

3.20.2.2 void TService::toBuffer (buffer_t & *buf*) [inline], [virtual]

Get byte buffer representation.

Parameters

out	*buf*	Target buffer.

Returns

None.

Implements ISerializable.

Definition at line 70 of file sensors.h.

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

- sensors.h

Index