NAME

ost::BaseObject −

BaseObject.

SYNOPSIS

`#include <persist.h>`

Public Member Functions

BaseObject ()

This constructor is used in serialisation processes.

virtual **˜BaseObject** ()

Default destructor.

virtual const char * **getPersistenceID** () const

This returns the ID of the persistent object (Its type).

virtual bool **write** (**Engine** &archive) const

This method is used to write to the Persistence::Engine It is not equivalent to the << operator as it writes only the data and not the object type etc.

virtual bool **read** (**Engine** &archive)

This method is used to read from a Persistence::Engine It is not equivalent to the >> operator as it does no typesafety or anything.

Detailed Description

BaseObject.

This object is the base for all Persistent data which is not natively serialised by the Persistence::Engine

It registers itself with the Persistence::TypeManager using a global constructor function. A matching deregister call is made in a global destructor, to allow DLL’s to use the Persistence::Engine in a main executable.

Persistable objects must never maintain bad pointers. If a pointer doesn’t point to something valid, it must be NULL. This is so the persistence engine knows whether to allocate memory for an object or whether the memory has been pre-allocated.

Author:

Daniel Silverstone Base class for classes that will be persistent.

Constructor & Destructor Documentation

ost::BaseObject::BaseObject ()

This constructor is used in serialisation processes. It is called in CreateNewInstance in order to create an instance of the class to have Read() called on it.

virtual ost::BaseObject::˜BaseObject () `[virtual]`

Default destructor.

Member Function Documentation

virtual const char* ost::BaseObject::getPersistenceID () const `[virtual]`

This returns the ID of the persistent object (Its type).

virtual bool ost::BaseObject::read (Engine & archive) `[virtual]`

This method is used to read from a Persistence::Engine It is not equivalent to the >> operator as it does no typesafety or anything.

virtual bool ost::BaseObject::write (Engine & archive) const `[virtual]`

This method is used to write to the Persistence::Engine It is not equivalent to the << operator as it writes only the data and not the object type etc.

Author

Generated automatically by Doxygen for GNU CommonC++ from the source code.