SADB-Client

0.99

Generated by Doxygen 1.8.8

Wed Dec 10 2014 21:32:09

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 Blackboard Class Reference

Public Member Functions

- void **Initialize** (void)
- void **Free** (void)
- int **ConnectToServer** (void)
- int **ConnectToServer** (const char *address, int port=SADBPort)
- void **ConnectToServerWithRetry** (void)
- void **ConnectToServerWithRetry** (const char *address, int port=SADBPort)
- void **GenerateHeader** (SADBu8 *Header, SADBu8 CMD, SADBu64 OUID1, SADBu64 OUID2, SADBu64 Second, SADBu16 Millisecond, SADBu32 DataLength)
- void **Disconnect** (void)
- SADBs64 **Ping** (void)
- SADBu64 **CreateObject** (void)
- void **DeleteObject** (SADBu64 OUID)
- int **IsValidObject** (SADBu64 OUID)
- void **SetObjectName** (SADBu64 OUID, const char *Name)
- void **GetObjectName** (SADBu64 OUID, SADBu8 **Name)
- void **SetObjectDescription** (SADBu64 OUID, const char *Description)
- void **GetObjectDescription** (SADBu64 OUID, SADBu8 **Description)
- SADBu64 **GetObjectByName** (const char *Name)
- int **SendObjectValue** (SADBu64 OUID, BlackboardObject *Object)
- int **GetLatestObjectValue** (SADBu64 OUID, BlackboardObject *Object)
- int **GetLatestObjectTimestamp** (SADBu64 OUID, Timestamp *time)
- int **GetNextObjectTimestamp** (SADBu64 OUID, Timestamp *Current, Timestamp *Next)
- int **GetPreviousObjectTimestamp** (SADBu64 OUID, Timestamp *Current, Timestamp *Previous)
- int **GetNearestObjectTimestamp** (SADBu64 OUID, Timestamp *Target, Timestamp *Result)
- int **GetObjectValueAtTimestamp** (SADBu64 OUID, BlackboardObject *Object, Timestamp *Time, int Interpolation, int nPoints)
- int **GetObjectValueAtNearestTimestamp** (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- int **RemoveOldestValue** (SADBu64 OUID)
- int **KeepOnlyLatestValues** (SADBu64 OUID, int nValues)
- int **GetObjectValueAtNextTimestamp** (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- int **GetObjectValueAtPreviousTimestamp** (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- SADBu64 **CreateCategory** (void)
- SADBu64 **CreateCategoryIfNonExistant** (const char *Name)
- void **DeleteCategory** (SADBu64 CUID)
- int **IsValidCategory** (SADBu64 CUID)

- void **SetCategoryName** (SADBu64 CUID, const char *Name)
- void **GetCategoryName** (SADBu64 CUID, SADBu8 **Name)
- void **SetCategoryDescription** (SADBu64 CUID, const char *Description)
- void **GetCategoryDescription** (SADBu64 CUID, SADBu8 **Description)
- SADBu64 **GetCategoryByName** (const char *Name)
- void **AddObjectToCategory** (SADBu64 CUID, SADBu64 OUID)
- void **RemoveObjectFromCategory** (SADBu64 CUID, SADBu64 OUID)
- int **FindObjects** (const char *Request, SADBu64 **Objects)
- SADBu64 **GetNextObjectID** (void)
- SADBu64 **GetNextCategoryID** (void)
- void **Reset** (void)

Public Attributes

- char * **Address**
- int **Port**

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

- Blackboard.h

2.2 SADB::Blackboard Class Reference

Public Member Functions

- Blackboard (void)
- ~Blackboard (void)
- void Initialize (void)

  Initializes object.
- void Free (void)

  Frees allocated memory.
- int ConnectToServer (void)

  Connect to the SADB Server.
- int ConnectToServer (const char *address, int port=SADBPort)
- void ConnectToServerWithRetry (void)

  Connect to the SADB Server and retry to connect until the connection is established.
- void ConnectToServerWithRetry (const char *address, int port=SADBPort)
- void **GenerateHeader** (SADBu8 *Header, SADBu8 CMD, SADBu64 OUID1, SADBu64 OUID2, SADBu64 Second, SADBu16 Millisecond, SADBu32 DataLength)
- void Disconnect (void)

  Disconnect from the SADB Server.
- SADBs64 Ping (void)
- SADBu64 CreateObject (void)
- void DeleteObject (SADBu64 OUID)
- int IsValidObject (SADBu64 OUID)
- void SetObjectName (SADBu64 OUID, const char *Name)
- void GetObjectName (SADBu64 OUID, SADBu8 **Name)
- void SetObjectDescription (SADBu64 OUID, const char *Description)
- void GetObjectDescription (SADBu64 OUID, SADBu8 **Description)
- SADBu64 GetObjectByName (const char *Name)
- int SendObjectValue (SADBu64 OUID, BlackboardObject *Object)
- int GetLatestObjectValue (SADBu64 OUID, BlackboardObject *Object)

- int GetLatestObjectTimestamp (SADBu64 OUID, Timestamp *time)
- int GetNextObjectTimestamp (SADBu64 OUID, Timestamp *Current, Timestamp *Next)
- int GetPreviousObjectTimestamp (SADBu64 OUID, Timestamp *Current, Timestamp *Previous)
- int GetNearestObjectTimestamp (SADBu64 OUID, Timestamp *Target, Timestamp *Result)
- int GetObjectValueAtTimestamp (SADBu64 OUID, BlackboardObject *Object, Timestamp *Time, int Interpolation, int nPoints)
- int GetObjectValueAtNearestTimestamp (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- int RemoveOldestValue (SADBu64 OUID)
- int KeepOnlyLatestValues (SADBu64 OUID, int nValues)
- int GetObjectValueAtNextTimestamp (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- int GetObjectValueAtPreviousTimestamp (SADBu64 OUID, Timestamp *Time, BlackboardObject *Object)
- SADBu64 CreateCategory (void)
- SADBu64 CreateCategoryIfNonExistant (const char *Name)
- void DeleteCategory (SADBu64 CUID)
- int IsValidCategory (SADBu64 CUID)
- void SetCategoryName (SADBu64 CUID, const char *Name)
- void GetCategoryName (SADBu64 CUID, SADBu8 **Name)
- void SetCategoryDescription (SADBu64 CUID, const char *Description)
- void GetCategoryDescription (SADBu64 CUID, SADBu8 **Description)
- SADBu64 GetCategoryByName (const char *Name)
- void AddObjectToCategory (SADBu64 CUID, SADBu64 OUID)
- void RemoveObjectFromCategory (SADBu64 CUID, SADBu64 OUID)
- int FindObjects (const char *Request, SADBu64 **Objects)
- SADBu64 GetNextObjectID (void)
- SADBu64 GetNextCategoryID (void)
- void Reset (void)

Public Attributes

- char * **Address**
- int **Port**

2.2.1 Constructor & Destructor Documentation

2.2.1.1 Blackboard::Blackboard (void)

Constructor, initializes object

See also

Blackboard::Initialize

2.2.1.2 Blackboard::~Blackboard (void)

Destructor, deinitializes object

See also

Blackboard::Free

2.2.2 Member Function Documentation

2.2.2.1 void Blackboard::AddObjectToCategory (SADBu64 *CUID*, SADBu64 *OUID*)

Add an object to a category

Parameters

CUID	ID of the category
OUID	ID of the object

2.2.2.2 int Blackboard::ConnectToServer (const char * *address*, int *port* = SADBPort)

Connect to the SADB Server

Parameters

Address	Address of the server
Port	Port of the server

2.2.2.3 void Blackboard::ConnectToServerWithRetry (const char * *address*, int *port* = SADBPort)

Connect to the SADB Server and retry to connect until the connection is established

Parameters

Address	Address of the server
Port	Port of the server

2.2.2.4 SADBu64 Blackboard::CreateCategory (void)

Create a new category on the blackboard

Returns

CUID of the new category

2.2.2.5 SADBu64 Blackboard::CreateCategoryIfNonExistant (const char * *Name*)

Get the ID of a category given its name. If the category does not exist, create a new one and return its ID.

Parameters

Name	Name of the category

Returns

CUID of the (possibly new) category

2.2.2.6 SADBu64 Blackboard::CreateObject (void)

Create a new object on the blackboard

Returns

OUID of the new object

2.2.2.7 void Blackboard::DeleteCategory (SADBu64 *CUID*)

Delete an existing category from the blackboard

Parameters

CUID	ID of the category to remove

2.2.2.8 void Blackboard::DeleteObject (SADBu64 *OUID*)

delete an existing object from the blackboard

Parameters

OUID	ID of the object to remove

2.2.2.9 int Blackboard::FindObjects (const char * *Request,* SADBu64 ** *Objects*)

Find IDs of objects matching a query

Note

Make sure the pointer to the list is initialized to NULL

Parameters

Request	String representation of the request
Objects	Pointer to a list of objet IDs

Returns

Number of Objects matching the query

2.2.2.10 SADBu64 Blackboard::GetCategoryByName (const char * *Name*)

Get the CUID of a category given its name

Parameters

Name	Name of the category CUID of the object, or 0 if not found

2.2.2.11 void Blackboard::GetCategoryDescription (SADBu64 *CUID,* SADBu8 ** *Description*)

Get the Description of a category on the blackboard

Parameters

CUID	ID of the target category
Name	Return address

2.2.2.12 void Blackboard::GetCategoryName (SADBu64 *CUID,* SADBu8 ** *Name*)

Get the name of a category on the blackboard

Parameters

CUID	ID of the target category
Name	Return address

2.2.2.13 int Blackboard::GetLatestObjectTimestamp (SADBu64 *OUID*, Timestamp * *time*)

Get the latest timestamp of an object

Note

This method does not get the value of the object, only the timestamp

Parameters

OUID	ID of the object in SADB
time	Pointer to a Timestamp object

Returns

0 on success, 1 otherwise

2.2.2.14 int Blackboard::GetLatestObjectValue (SADBu64 *OUID*, BlackboardObject * *Object*)

Get the latest value of an object

Parameters

OUID	ID of the object in SADB
Object	Pointer to an object that will contain the latest value

Returns

0 on success, 1 otherwise

2.2.2.15 int Blackboard::GetNearestObjectTimestamp (SADBu64 *OUID*, Timestamp * *Target*, Timestamp * *Result*)

Get the nearest timestamp of an object to the target

Note

This method does not get the value of the object, only the timestamp

Parameters

	OUID	ID of the object in SADB
	Target	Pointer to a Timestamp object
out	*Result*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.2.2.16 SADBu64 Blackboard::GetNextCategoryID (void)

Get the next Category ID.

Returns

Next available Category ID, or 0 on error (0 is the ID number reserved for no category)

2.2.2.17 SADBu64 Blackboard::GetNextObjectID (void)

Get the next Object ID.

Returns

Next available Object ID, or 0 on error (0 is the ID number reserved for no object)

2.2.2.18 int Blackboard::GetNextObjectTimestamp (SADBu64 *OUID,* Timestamp * *Current,* Timestamp * *Next*)

Get the next timestamp of an object

Note

This method does not get the value of the object, only the timestamp

Parameters

	OUID	ID of the object in SADB
	Current	Pointer to a Timestamp object
out	*Next*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.2.2.19 SADBu64 Blackboard::GetObjectByName (const char * *Name*)

Get the OUID of an object given its name

Parameters

Name	Name of the object OUID of the object, or 0 if not found

2.2.2.20 void Blackboard::GetObjectDescription (SADBu64 *OUID,* SADBu8 ** *Description*)

Get the Description of an object on the blackboard

Parameters

OUID	ID of the target object
Name	Return address

2.2.2.21 void Blackboard::GetObjectName (SADBu64 *OUID,* SADBu8 ** *Name*)

Get the name of an object on the blackboard

Parameters

OUID	ID of the target object
Name	Return address

2.2.2.22 int Blackboard::GetObjectValueAtNearestTimestamp (SADBu64 *OUID*, Timestamp * *Time*, BlackboardObject * *Object*)

Get the value of an object at the nearest Timestamp

Parameters

OUID	ID of the object in SADB
Time	Requested Timestamp
Object	Pointer to an object that will contain the latest value

Returns

0 on success, 1 otherwise

2.2.2.23 int Blackboard::GetObjectValueAtNextTimestamp (SADBu64 *OUID*, Timestamp * *Time*, BlackboardObject * *Object*)

Get the value of an object at the next Timestamp after the target time

Note

if there is no timestamp after the target time, the last value (latest) is returned

Parameters

OUID	ID of the object in SADB
Time	Target Timestamp
Object	Pointer to an object that will contain the value

Returns

0 on success, 1 otherwise

2.2.2.24 int Blackboard::GetObjectValueAtPreviousTimestamp (SADBu64 *OUID*, Timestamp * *Time*, BlackboardObject * *Object*)

Get the value of an object at the next Timestamp after the target time

Note

if there is no timestamp before the target time, the first value (earliest) is returned

Parameters

OUID	ID of the object in SADB

Time	Target Timestamp
Object	Pointer to an object that will contain the value

Returns

0 on success, 1 otherwise

2.2.2.25 int Blackboard::GetObjectValueAtTimestamp (SADBu64 *OUID*, BlackboardObject * *Object*, Timestamp * *Time*, int *Interpolation*, int *nPoints*)

Get the latest value of an object at a given timestamp

Parameters

OUID	ID of the object in SADB
Object	Pointer to an object that will contain the value
Time	Requested timestamp
Interpolation	Requested interpolation method
nPoints	Number of points to use for interpolation

Returns

0 on success, 1 otherwise

2.2.2.26 int Blackboard::GetPreviousObjectTimestamp (SADBu64 *OUID*, Timestamp * *Current*, Timestamp * *Previous*)

Get the previous timestamp of an object

Note

This method does not get the value of the object, only the timestamp

Parameters

	OUID	ID of the object in SADB
	Current	Pointer to a Timestamp object
`out`	*Previous*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.2.2.27 int Blackboard::IsValidCategory (SADBu64 *CUID*)

Check if a Category exists (has been created and not deleted)

Parameters

CUID	ID of the category

Returns

1 if CUID is valid, 0 otherwise

2.2.2.28 int Blackboard::IsValidObject (SADBu64 *OUID*)

Check if an Object exists (has been created and not deleted)

Parameters

OUID	ID of the Object

Returns

1 if CUID is valid, 0 otherwise

2.2.2.29 int Blackboard::KeepOnlyLatestValues (SADBu64 *OUID,* int *nValues*)

Tell the SADB server to discard older values of an object and to keep only the nValues latest values.

Parameters

OUID	ID of the target object
nValues	number of values to keep

Returns

0 on success, 1 otherwise

2.2.2.30 SADBs64 Blackboard::Ping (void)

Ping the SADB Server for testing purposes and to determine latency

Returns

number of milliseconds between the ping and the pong

2.2.2.31 void Blackboard::RemoveObjectFromCategory (SADBu64 *CUID,* SADBu64 *OUID*)

Remove an object from a category

Parameters

CUID	ID of the category
OUID	ID of the object

2.2.2.32 int Blackboard::RemoveOldestValue (SADBu64 *OUID*)

Tell the SADB server to discard the oldest value of an object.

Parameters

OUID	ID of the target object

Returns

0 on success, 1 otherwise

2.2.2.33 void Blackboard::Reset (void)

Reset the SADB Server

Note

Should probably never be used

2.2.2.34 int Blackboard::SendObjectValue (SADBu64 *OUID,* BlackboardObject * *Object*)

Send the value of an object

Parameters

OUID	ID of the object in SADB
Object	Pointer to an object

Returns

Returns 0 on success

2.2.2.35 void Blackboard::SetCategoryDescription (SADBu64 *CUID,* const char * *Description*)

Set the description of a category on the blackboard

Parameters

CUID	ID of the category to edit
Description	Null-terminated string to set

2.2.2.36 void Blackboard::SetCategoryName (SADBu64 *CUID,* const char * *Name*)

Set the name of a category on the blackboard

Parameters

CUID	ID of the category to edit
Name	Null-terminated string to set

2.2.2.37 void Blackboard::SetObjectDescription (SADBu64 *OUID,* const char * *Description*)

Set the description of an object on the blackboard

Parameters

OUID	ID of the object to edit
Description	Null-terminated string to set

2.2.2.38 void Blackboard::SetObjectName (SADBu64 *OUID,* const char * *Name*)

Set the name of an object on the blackboard

Parameters

OUID	ID of the object to edit
Name	Null-terminated string to set

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

- SADB.h
- Blackboard.cpp

2.3 BlackboardObject Class Reference

Public Member Functions

- void **Initialize** (void)
- void **Free** (void)
- void **Create** (Blackboard ∗board, const char ∗name)
- void **Create** (Blackboard ∗board, const char ∗name, const char ∗description)
- void **Create** (Blackboard ∗board, const char ∗name, const char ∗description, enum BODataType type)
- void **Create** (Blackboard ∗board, const char ∗name, enum BODataType)
- void **SetBlackboard** (Blackboard ∗board)
- void **AddToBlackboard** (void)
- int **Send** (void)
- void **AddToCategory** (SADBu64 CUID)
- void **AddToCategory** (const char ∗Cat)
- void **RemoveFromCategory** (SADBu64 CUID)
- void **RemoveFromCategory** (const char ∗Cat)
- void **CreateIfNonExistant** (const char ∗name)
- void **GetByName** (const char ∗name)
- int **GetLatestValue** (void)
- int **GetValueAtNearestTimestamp** (Timestamp ∗Time)
- int **GetValueAtTimestampNoInterpolation** (Timestamp ∗Time)
- int **GetValueAtTimestamp** (Timestamp ∗Time)
- int **GetValueAtTimestamp** (Timestamp ∗Time, int Interpolation)
- int **GetValueAtTimestamp** (Timestamp ∗Time, int Interpolation, int nPoints)
- int **GetValueAtNextTimestamp** (Timestamp ∗Time)
- int **GetValueAtPreviousTimestamp** (Timestamp ∗Time)
- int **GetLatestTimestamp** (Timestamp ∗Time)
- int **GetPreviousTimestamp** (Timestamp ∗Current, Timestamp ∗Previous)
- int **GetNextTimestamp** (Timestamp ∗Current, Timestamp ∗Next)
- int **GetNearestTimestamp** (Timestamp ∗Target, Timestamp ∗Result)
- void **SetDataType** (enum BODataType)
- int **SetDimensionsf** (SADBu32 D,...)
- int **OnlySetDimensionsf** (SADBu32 D,...)
- int **SetDimensions** (SADBu32 D, SADBu32 ∗Ds)
- SADBu32 **GetDimension** (SADBu32 D)
- int **SetName** (const char ∗name)
- void **GetName** (void)
- int **SetDescription** (const char ∗desc)
- void **GetDescription** (void)
- void **CreateWithInitialValueFromCSV** (Timestamp ∗Time, const char ∗Filename, enum BODataType DataType, char Delimiter)
- void **CreateWithInitialValueFromCSV** (const char ∗Filename, enum BODataType DataType)
- void **CreateWithInitialValueFromCSV** (const char ∗Filename, enum BODataType DataType, char Delimiter)
- void **CreateWithValuesFromCSV** (const char ∗Filename, enum BODataType DataType)
- void **CreateWithValuesFromCSV** (const char ∗Filename, enum BODataType DataType, char Delimiter)
- void **ExportAllValuesToCSV** (const char ∗Filename)
- void **ExportRaw** (const char ∗Filename)
- void **ImportRaw** (const char ∗Filename)
- void **ImportRaw** (const char ∗Filename, float A, float B)
- void **SetByte** (SADBu8 Value,...)
- void **SetInteger** (SADBu32 Value,...)
- void **SetFloat** (SADBfl Value,...)
- void **SetDouble** (SADBdfl Value,...)

- void **SetString** (const char *str)
- void **SetStringElement** (const char *str,...)
- void **SetBoolean** (bool Value,...)
- SADBu8 **GetByte** (int X,...)
- SADBu32 **GetInteger** (int X,...)
- SADBfl **GetFloat** (int X,...)
- SADBdfl **GetDouble** (int X,...)
- const char * **GetString** (void)
- char * **GetNewString** (void)
- bool **GetBoolean** (int X,...)
- char * **GetStringElement** (int X,...)
- void **RemoveOldestValue** (void)
- void **KeepOnlyLatestValues** (int nValues)
- int **CheckForNewValue** (void)
- void **Information** (void)
- void **SetTimestamp** (Timestamp *time)
- void **SetTimestampNow** (void)

Public Attributes

- Blackboard * Board

  Blackboard where the object resides.
- SADBu8 * Name

  Human Readable Object Name.
- SADBu8 * Description

  Human Readable Object Description.
- void * Data

  Pointer to Memory Location of the data.
- SADBu32 Dimensionality

  Dimensionality of the data (number of dimensiosns)
- SADBu32 * Dimensions

  Dimensions of the data.
- int TotalNumberOfElements

  Total number of elements (Product of all dimensions)
- enum BODataType DataType

  Data Type.
- int ElementSize

  Element Size (in bytes)
- SADBu64 ID

  Object ID on the blackboard.
- Timestamp Time

  Timestamp of the data.

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

- BlackboardObject.h

2.4 SADB::BlackboardObject Class Reference

Public Member Functions

- BlackboardObject (void)
- ~BlackboardObject (void)
- void Initialize (void)

  Initializes object.
- void Free (void)

  Frees allocated memory.
- void Create (Blackboard *board, const char *name)
- void Create (Blackboard *board, const char *name, const char *description)
- void Create (Blackboard *board, const char *name, const char *description, enum BODataType type)
- void Create (Blackboard *board, const char *name, enum BODataType)
- void SetBlackboard (Blackboard *board)
- void AddToBlackboard (void)
- int Send (void)
- void AddToCategory (SADBu64 CUID)
- void AddToCategory (const char *Cat)
- void RemoveFromCategory (SADBu64 CUID)
- void RemoveFromCategory (const char *Cat)
- void CreateIfNonExistant (const char *name)
- void GetByName (const char *name)
- int GetLatestValue (void)
- int GetValueAtNearestTimestamp (Timestamp *Time)
- int GetValueAtTimestampNoInterpolation (Timestamp *Time)
- int GetValueAtTimestamp (Timestamp *Time)
- int GetValueAtTimestamp (Timestamp *Time, int Interpolation)
- int GetValueAtTimestamp (Timestamp *Time, int Interpolation, int nPoints)
- int GetValueAtNextTimestamp (Timestamp *Time)
- int GetValueAtPreviousTimestamp (Timestamp *Time)
- int GetLatestTimestamp (Timestamp *Time)
- int GetPreviousTimestamp (Timestamp *Current, Timestamp *Previous)
- int GetNextTimestamp (Timestamp *Current, Timestamp *Next)
- int GetNearestTimestamp (Timestamp *Target, Timestamp *Result)
- void SetDataType (enum BODataType)
- int SetDimensionsf (SADBu32 D,...)
- int OnlySetDimensionsf (SADBu32 D,...)
- int SetDimensions (SADBu32 D, SADBu32 *Ds)
- SADBu32 GetDimension (SADBu32 D)
- int SetName (const char *name)
- void GetName (void)

  Get Human Readable name of the object from the blackboard.
- int SetDescription (const char *desc)
- void GetDescription (void)

  Get Human Readable description of the object from the blackboard.
- void CreateWithInitialValueFromCSV (Timestamp *Time, const char *Filename, enum BODataType DataType, char Delimiter)
- void CreateWithInitialValueFromCSV (const char *Filename, enum BODataType DataType)
- void CreateWithInitialValueFromCSV (const char *Filename, enum BODataType DataType, char Delimiter)
- void CreateWithValuesFromCSV (const char *Filename, enum BODataType DataType)
- void CreateWithValuesFromCSV (const char *Filename, enum BODataType DataType, char Delimiter)
- void ExportAllValuesToCSV (const char *Filename)
- void ExportRaw (const char *Filename)

- void ImportRaw (const char *Filename)
- void ImportRaw (const char *Filename, float A, float B)
- void SetByte (SADBu8 Value,...)
- void SetInteger (SADBu32 Value,...)
- void SetFloat (SADBfl Value,...)
- void SetDouble (SADBdfl Value,...)
- void SetString (const char *str)
- void SetStringElement (const char *str,...)
- void SetBoolean (bool Value,...)
- SADBu8 GetByte (int X,...)
- SADBu32 GetInteger (int X,...)
- SADBfl GetFloat (int X,...)
- SADBdfl GetDouble (int X,...)
- const char * GetString (void)
- char * GetNewString (void)
- bool GetBoolean (int X,...)
- char * GetStringElement (int X,...)
- void RemoveOldestValue (void)

  Remove oldest value from the blackboard.
- void KeepOnlyLatestValues (int nValues)
- int CheckForNewValue (void)
- void Information (void)

  Print Human Readable information about the object.
- void SetTimestamp (Timestamp *time)
- void SetTimestampNow (void)

  Set the timestamp to the current time.

Public Attributes

- Blackboard * Board

  Blackboard where the object resides.
- SADBu8 * Name

  Human Readable Object Name.
- SADBu8 * Description

  Human Readable Object Description.
- void * Data

  Pointer to Memory Location of the data.
- SADBu32 Dimensionality

  Dimensionality of the data (number of dimensiosns)
- SADBu32 * Dimensions

  Dimensions of the data.
- int TotalNumberOfElements

  Total number of elements (Product of all dimensions)
- enum BODataType DataType

  Data Type.
- int ElementSize

  Element Size (in bytes)
- SADBu64 ID

  Object ID on the blackboard.
- Timestamp Time

  Timestamp of the data.

2.4.1 Constructor & Destructor Documentation

2.4.1.1 BlackboardObject::BlackboardObject (void)

Constructor, initializes object

See also

BlackboardObject::Initialize

2.4.1.2 BlackboardObject::~BlackboardObject (void)

Destructor, deinitializes object

See also

BlackboardObject::Free

2.4.2 Member Function Documentation

2.4.2.1 void BlackboardObject::AddToBlackboard (void)

Add the object to a blackboard

Note

an object may only be associated with a single blackboard

Parameters

board	Pointer to the target blackboard object

2.4.2.2 void BlackboardObject::AddToCategory (SADBu64 *CUID*)

Add the object to a category

Parameters

CUID	ID of the category

2.4.2.3 void BlackboardObject::AddToCategory (const char * *Cat*)

Add the object to a category

Parameters

Cat	Name of the category

2.4.2.4 int BlackboardObject::CheckForNewValue (void)

Check to see if there is a new value on the SADB server, without retrieving it

Returns

1 if a new value is present, 0 if no new value is available, and 2 on error

2.4.2.5 void BlackboardObject::Create (Blackboard * *board,* const char * *name*)

Create an object (convenience function)

Parameters

board	Pointer to the target blackboard object
name	Name of the object

2.4.2.6 void BlackboardObject::Create (Blackboard * *board*, const char * *name*, const char * *description*)

Create an object (convenience function)

Parameters

board	Pointer to the target blackboard object
name	Name of the object
description	Description of the object

2.4.2.7 void BlackboardObject::Create (Blackboard * *board*, const char * *name*, const char * *description*, enum BODataType *type*)

Create an object (convenience function)

Parameters

board	Pointer to the target blackboard object
name	Name of the object
description	Description of the object
type	Datatype of the object

2.4.2.8 void BlackboardObject::Create (Blackboard * *board*, const char * *name*, enum BODataType *type*)

Create an object (convenience function)

Parameters

board	Pointer to the target blackboard object
name	Name of the object
type	Datatype of the object

2.4.2.9 void BlackboardObject::CreateIfNonExistant (const char * *name*)

Get an object from the blackboard or create it if it doesn't already exist

Parameters

name	Name of the object

2.4.2.10 void BlackboardObject::CreateWithInitialValueFromCSV (Timestamp * *time*, const char * *Filename*, enum BODataType *DataType*, char *Delimiter*)

Initialize the object with data from a comma separated values (CSV) file

Note

The entire CSV file gets loaded as the value of the object for a given timestamp

Parameters

Time	Pointer to a timestamp object
Filename	Path to the CSV file
DataType	Type of the data in the CSV File
Delimiter	Delimiter (normally ',')

2.4.2.11 void BlackboardObject::CreateWithInitialValueFromCSV (const char * *Filename,* enum BODataType *DataType*)

Initialize the object with data from a comma separated values (CSV) file

Note

The entire CSV file gets loaded as the value of the object for a given timestamp
Uses the current time as the timestamp

Parameters

Filename	Path to the CSV file
DataType	Type of the data in the CSV File

2.4.2.12 void BlackboardObject::CreateWithInitialValueFromCSV (const char * *Filename,* enum BODataType *DataType,* char *Delimiter*)

Initialize the object with data from a comma separated values (CSV) file

Note

The entire CSV file gets loaded as the value of the object for a given timestamp
Uses the current time as the timestamp

Parameters

Filename	Path to the CSV file
DataType	Type of the data in the CSV File
Delimiter	Delimiter (normally ',')

2.4.2.13 void BlackboardObject::CreateWithValuesFromCSV (const char * *Filename,* enum BODataType *DataType*)

Initialize the object with data from a comma separated values (CSV) file

Note

the first column of each row should contain a timestamp encoded as a double, each row of data will be loaded on the blackboard as a different value at a distinct timestamp

Parameters

Filename	Path to the CSV file
DataType	Type of the data in the CSV File (other than the first column, which MUST be a double/float)

2.4.2.14 void BlackboardObject::CreateWithValuesFromCSV (const char * *Filename,* enum BODataType *DataType,* char *Delimiter*)

Initialize the object with data from a comma separated values (CSV) file

Note

the first column of each row should contain a timestamp encoded as a double, each row of data will be loaded on the blackboard as a different value at a distinct timestamp

Parameters

Filename	Path to the CSV file
DataType	Type of the data in the CSV File (other than the first column, which MUST be a double/float)
Delimiter	Delimiter (normally ',')

2.4.2.15 void BlackboardObject::ExportAllValuesToCSV (const char * *Filename*)

Write the values of the object to a CSV file. The first element of each line is the timestamp and the remaining cells of the line contain the value.

Note

For the time being, each value is vectorized to fit on a line, dimensions information is discarded. This may cause problems with objects of higher dimensions.
If the dimensions of the object changes over time, the lines of the CSV file will have different lengths in order to export the entire data.

Parameters

Filename	Path to the CSV file

2.4.2.16 void BlackboardObject::ExportRaw (const char * *Filename*)

Write the values of the object to a raw file. Similar to the exportCSV function, but this one also stores information about the dimensions of objects. The first element of each line is the timestamp, the second element is the data type, the third column is the number of dimensions, the next few columns are the dimensions, and the remaining cells contain the value.

Parameters

Filename	Path to the output file

2.4.2.17 bool BlackboardObject::GetBoolean (int *X, ...*)

Get the value of a boolean in the object at a given location

Note

Booleans are stored as bytes (false is stored as 0, true as 1)

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.18 void BlackboardObject::GetByName (const char * *name*)

Get the object Id from the associated blackboard from the name

Parameters

name	Name of the object on the blackboard

2.4.2.19 SADBu8 BlackboardObject::GetByte (int *X*, ...)

Get the value of a byte in the object at a given location

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.20 SADBu32 BlackboardObject::GetDimension (SADBu32 *D*)

The the size in the nth dimension

Parameters

D	Index of the desired dimension (starts at 0)

Returns

The size, or 0 if dimension doesn't exist

2.4.2.21 SADBdfl BlackboardObject::GetDouble (int *X*, ...)

Get the value of a double precision float in the object at a given location

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.22 SADBfl BlackboardObject::GetFloat (int *X*, ...)

Get the value of a float in the object at a given location

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.23 SADBu32 BlackboardObject::GetInteger (int *X*, ...)

Get the value of an integer in the object at a given location

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.24 int BlackboardObject::GetLatestTimestamp (Timestamp * *Time*)

Get the latest timestamp on the server

Note

This function only gets the timestamp, not the acutal data

Parameters

out	*Time*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.4.2.25 int BlackboardObject::GetLatestValue (void)

Get the latest value form the SADB server

Returns

0 on success, 1 otherwise

2.4.2.26 int BlackboardObject::GetNearestTimestamp (Timestamp * *Target,* Timestamp * *Result*)

Get the timestamp on server closest to the rquested one

Note

This function only gets the timestamp, not the acutal data

Parameters

	Target	Pointer to a Timestamp object
out	*Result*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.4.2.27 char * BlackboardObject::GetNewString (void)

Get the content of the object as a string

Note

this function assumes that the object contains a single string, for string tables, look a the SADBStringTable class
This functions allocates memory, you mustt free the returned pointer yourself

Returns

Pointer to the start of the string

2.4.2.28 int BlackboardObject::GetNextTimestamp (Timestamp * *Current,* Timestamp * *Next*)

Get the next timestamp on the server

Note

This function only gets the timestamp, not the acutal data

Parameters

	Current	Pointer to a Timestamp object
out	*Next*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.4.2.29 int BlackboardObject::GetPreviousTimestamp (Timestamp * *Current*, Timestamp * *Previous*)

Get the previous timestamp on the server

Note

This function only gets the timestamp, not the acutal data

Parameters

	Current	Pointer to a Timestamp object
out	*Previous*	Pointer to a Timestamp object to store result

Returns

0 on success, 1 otherwise

2.4.2.30 const char * BlackboardObject::GetString (void)

Get the content of the object as a string

Note

this function assumes that the object contains a single string, for string tables, look a the SADBStringTable class

Returns

Pointer to the start of the string

2.4.2.31 char * BlackboardObject::GetStringElement (int *X*, ...)

Get the value of a float in the object at a given location

Parameters

X	Position in the first dimension (required as a first argument to va_start)
...	Position of the element in the other dimensions

2.4.2.32 int BlackboardObject::GetValueAtNearestTimestamp (Timestamp * *Time*)

Get the value at the nearest timestamp

Parameters

Time	Requested Timestamp

Returns

0 on success, 1 otherwise

2.4.2.33 int BlackboardObject::GetValueAtNextTimestamp (Timestamp * *Time*)

Get the value at the next timestamp

Parameters

Time	Target Timestamp

Returns

0 on success, 1 otherwise

2.4.2.34 int BlackboardObject::GetValueAtPreviousTimestamp (Timestamp * *Time*)

Get the value at the previous timestamp

Parameters

Time	Target Timestamp

Returns

0 on success, 1 otherwise

2.4.2.35 int BlackboardObject::GetValueAtTimestamp (Timestamp * *Time*)

Get the value at the exact timestamp with default interpolation method (linear)

Parameters

Time	Requested Timestamp

Returns

0 on success, 1 otherwise

2.4.2.36 int BlackboardObject::GetValueAtTimestamp (Timestamp * *Time*, int *Interpolation*)

Get the value at the nearest timestamp

Parameters

Time	Requested Timestamp

Interpolation	Requested Interpolation method

Returns

0 on success, 1 otherwise

2.4.2.37 int BlackboardObject::GetValueAtTimestamp (Timestamp * *Time*, int *Interpolation*, int *nPoints*)

Get the value at the nearest timestamp

Parameters

Time	Requested Timestamp
Interpolation	Requested Interpolation method
nPoints	Number of points to use for interpolation

Returns

0 on success, 1 otherwise

2.4.2.38 int BlackboardObject::GetValueAtTimestampNoInterpolation (Timestamp * *Time*)

Get the value at a given timestamp, if such a value exists

Parameters

Time	Requested Timestamp

Returns

0 on success, 1 otherwise

2.4.2.39 void BlackboardObject::ImportRaw (const char * *Filename*)

Initialize the object with data from a raw data file (exported with exportraw)

Parameters

Filename	Path to the raw file

2.4.2.40 void BlackboardObject::ImportRaw (const char * *Filename*, float *A*, float *B*)

Initialize the object with data from a raw data file (exported with exportraw)

Parameters

Filename	Path to the raw file
A	Beginning Percentage
A	End Percentage

2.4.2.41 void BlackboardObject::KeepOnlyLatestValues (int *nValues*)

Discard earliest values and keep only the n most recent

Parameters

nValues	Number of Values to be kept

2.4.2.42 int BlackboardObject::OnlySetDimensionsf (SADBu32 *D*, ...)

Sets the size of the object by defining its dimensionality and its dimensions, without allocating memory

Parameters

D	Number of Dimensions
...	Dimensions (Height, Width, etc)

Returns

0 on success

2.4.2.43 void BlackboardObject::RemoveFromCategory (SADBu64 *CUID*)

Add the object to a category

Parameters

CUID	ID of the category

2.4.2.44 void BlackboardObject::RemoveFromCategory (const char * *Cat*)

Add the object to a category

Parameters

Cat	Name of the category

2.4.2.45 int BlackboardObject::Send (void)

Send the object data to the blackboard

Returns

Returns 0 on success, will return 1 if either the board is not defined or the transfer fails

2.4.2.46 void BlackboardObject::SetBlackboard (Blackboard * *board*)

Associate the object to a blackboard

Note

an object may only be associated with a single blackboard

Parameters

board	Pointer to the target blackboard object

2.4.2.47 void BlackboardObject::SetBoolean (bool *Value,* ...)

Sets the value of an element in the object

Note

Booleans are stored as bytes (false is stored as 0, true as 1)

Parameters

Value	Value to set
...	Position of the element to receive the value (X,Y,Z,...)

2.4.2.48 void BlackboardObject::SetByte (SADBu8 *Value,* ...)

Sets the value of an element in the object

Parameters

Value	Value to set
...	Position of the element to receive the value (X,Y,Z,...)

2.4.2.49 void BlackboardObject::SetDataType (enum BODataType *Type*)

Sets the type of data that the object will contain

Parameters

Type	Data Type

2.4.2.50 int BlackboardObject::SetDescription (const char * *desc*)

Set Human Readable description of the object

Parameters

desc	Description

Returns

0 on success

2.4.2.51 int BlackboardObject::SetDimensions (SADBu32 *D,* SADBu32 * *Ds*)

Sets the size of the object by defining its dimensionality and its dimensions

Parameters

D	Number of Dimensions
Ds	list of Dimensions (Height, Width, etc)

Returns

0 on success

2.4.2.52 int BlackboardObject::SetDimensionsf (SADBu32 *D,* ...)

Sets the size of the object by defining its dimensionality and its dimensions

Parameters

D	Number of Dimensions
...	Dimensions (Height, Width, etc)

Returns

0 on success

2.4.2.53 void BlackboardObject::SetDouble (SADBdfl *Value,* ...)

Sets the value of an element in the object

Parameters

Value	Value to set
...	Position of the element to receive the value (X,Y,Z,...)

2.4.2.54 void BlackboardObject::SetFloat (SADBfl *Value,* ...)

Sets the value of an element in the object

Parameters

Value	Value to set
...	Position of the element to receive the value (X,Y,Z,...)

2.4.2.55 void BlackboardObject::SetInteger (SADBu32 *Value,* ...)

Sets the value of an element in the object

Parameters

Value	Value to set
...	Position of the element to receive the value (X,Y,Z,...)

2.4.2.56 int BlackboardObject::SetName (const char * *name*)

Set Human Readable name of the object

Parameters

name	Name

Returns

0 on success

2.4.2.57 void BlackboardObject::SetString (const char * *str*)

Set a string as the content of the object.

Note

this function assumes that the object contains a single string, for string tables, look a the SADBStringTable class
Allocates the required memory

Parameters

str	Desired string

2.4.2.58 void BlackboardObject::SetStringElement (const char * *str, ...*)

Sets a string in an object

Parameters

str	String
...	Position of the element to receive the value (X,Y,Z,...), the last dimension stores the string

2.4.2.59 void BlackboardObject::SetTimestamp (**Timestamp** * *time*)

Set the timestamp

Parameters

time	Desired timestamp

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

- SADB.h
- BlackboardObject.cpp

2.5 SADB::BlackboardObjectListener Class Reference

Public Member Functions

- BlackboardObjectListener (void)
- ~BlackboardObjectListener (void)
- void Initialize (BlackboardObject *Obj, SADBu64 period, void(*callback)(BlackboardObject *))
- void Free (void)

  Frees allocated memory.
- void Start (void)

Start the main thread for the Object Listener.

- void Stop (void)

  Stop the main thread for the Object Listener.

Public Attributes

- SADBu64 **Period**

2.5.1 Constructor & Destructor Documentation

2.5.1.1 BlackboardObjectListener::BlackboardObjectListener (void)

Constructor, initializes object

See also

Blackboard::Initialize

2.5.1.2 BlackboardObjectListener::~BlackboardObjectListener (void)

Destructor, deinitializes object

See also

Blackboard::Free

2.5.2 Member Function Documentation

2.5.2.1 void BlackboardObjectListener::Initialize (BlackboardObject * *Obj*, SADBu64 *period*, void(*)(BlackboardObject *) *callback*)

Initializes object

Parameters

Obj	Object to be periodically checked
period	Period of the check (in us)
callback	Callback function, which should have prototype "void functionname(BlackboardObject *)

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

- SADB.h
- BlackboardObjectListener.cpp

2.6 BlackboardObjectListener Class Reference

Public Member Functions

- void **Initialize** (BlackboardObject *Obj, SADBu64 period, void(*callback)(BlackboardObject *))
- void **Free** (void)
- void **Start** (void)
- void **Stop** (void)

Public Attributes

- SADBu64 **Period**

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

- BlackboardObjectListener.h

2.7 NanoTimer Class Reference

Public Member Functions

- void Start (void)

  Start the timer.
- long Stop (void)
- void Print (void)

  Prints the number of nanoseconds elapsed since the timer was started.
- void PrintInverse (void)

  Prints the inverse (1/x) of number of nanoseconds elapsed since the timer was started, good to measure frequencies.
- void WaitUntil (long time)

2.7.1 Member Function Documentation

2.7.1.1 long NanoTimer::Stop (void)

Returns the number of nanoseconds elapsed since the timer was started

Returns

nanoseconds

2.7.1.2 void NanoTimer::WaitUntil (long *time*)

Prints the inverse (1/x) of number of nanoseconds elapsed since the timer was started, good to measure frequencies

Parameters

time	Wait until this amount of time has elapsed since styarting the timer

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

- NanoTimer.h
- NanoTimer.cpp

2.8 Network Class Reference

Public Member Functions

- void **Close** (void)
- void **Server** (int port, int MaxQueue)
- int **ConnectTo** (const char *address, int port)
- int **WaitForConnection** (void)

- int **Receive** (unsigned char *Buffer, int MaxLength)
- int **ReceiveExactly** (unsigned char *Buffer, int Length)
- int **Send** (unsigned char *Buffer)
- int **Send** (unsigned char *Buffer, int Length)
- int **Sendf** (unsigned char *format,...)

Public Attributes

- int Socket

  Socket Descriptor.
- int Port

  Port Number.

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

- Network.h

2.9 SADB::Network Class Reference

Public Member Functions

- void Close (void)

  Close a Socket.
- void Server (int port, int MaxQueue)
- int ConnectTo (const char *address, int port)
- int WaitForConnection (void)

  Wait for a connection.
- int Receive (unsigned char *Buffer, int MaxLength)
- int ReceiveExactly (unsigned char *Buffer, int Length)
- int Send (unsigned char *Buffer)
- int Send (unsigned char *Buffer, int Length)
- int Sendf (unsigned char *format,...)

Public Attributes

- int Socket

  Socket Descriptor.
- int Port

  Port Number.

2.9.1 Member Function Documentation

2.9.1.1 int Network::ConnectTo (const char * *address*, int *port*)

Connect to a server

Parameters

in	*address*	Address to connect to
in	*port*	Port to use

2.9.1.2 int Network::Receive (unsigned char * *Buffer,* int *MaxLength*)

Receive data from the socket

Parameters

in	*Socket*	Socket to use
out	*Buffer*	Pointer to memory location where data read is to be stored
in	*MaxLength*	Maximum data to be read

Returns

Number of bytes read

2.9.1.3 int Network::ReceiveExactly (unsigned char * *Buffer,* int *Length*)

Receive a complete Packet from the socket

Parameters

in	*Socket*	Socket to use
out	*Buffer*	Pointer to memory location where data read is to be stored
in	*Length*	Maximum data to be read

Returns

Number of bytes read

2.9.1.4 int Network::Send (unsigned char * *Buffer*)

Send a string

Parameters

in	*Socket*	Socket to use
out	*Buffer*	Pointer to memory location where data to be sent is stored

2.9.1.5 int Network::Send (unsigned char * *Buffer,* int *Length*)

Send Data

Parameters

in	*Socket*	Socket to use
in	*Buffer*	Pointer to memory location where data to be sent is stored
in	*Length*	Number of bytes to be sent

2.9.1.6 int Network::Sendf (unsigned char * *format, ...*)

Send a Formatted String

Parameters

in	*Socket*	Socket to use
out	*format*	Format string
in	...	Additional variables

2.9.1.7 void Network::Server (int *port*, int *MaxQueue*)

Open a server socket

Parameters

in	*port*	Port to use

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

- SADB.h
- Network.cpp

2.10 SADB::SADBStringTable Class Reference

`#include <SADB.h>`

Public Member Functions

- SADBStringTable (void)
- ~SADBStringTable (void)
- void Initialize (void)

  Initialize Model.
- void Free (void)

  Free all allocated memory.
- void SetName (const char *name)
- void GetByName (const char *name)
- void SetDescription (const char *desc)
- void SetBlackboard (Blackboard *board)
- void SetDimensions (int X, int Y)
- void SetString (int X, int Y, const char *Str)
- const char * GetString (int X, int Y)
- char * GetNewString (int X, int Y)
- void LoadCSV (const char *Filename)

  Load a csv file containing strings.
- void GetLatestValues (void)

  Get the latest version of the table from the board.
- void GetLatestValue (int X, int Y)

Public Attributes

- SADBu8 * Name

  Human Readable Object Name.
- BlackboardObject Object

  Object to store the list of indices.
- int nCols

Number of Columns.

- int nRows

  Number of Rows.

- Blackboard * Board

  Blackboard.

2.10.1 Detailed Description

Class used to store tables of strings in SADB

Note

If you change the dimensions of the table to a smaller size (shrink), some object will remain on the blackboard without being linked to a string table. If the table size is incremented at a later time, the objects will be linked again. This is to preserve history.

2.10.2 Constructor & Destructor Documentation

2.10.2.1 SADBStringTable::SADBStringTable (void)

Constructor, initializes object

See also

SADBStringTable::Initialize

2.10.2.2 SADBStringTable::~SADBStringTable (void)

Destructor, deinitializes object

See also

SADBStringTable::Free

2.10.3 Member Function Documentation

2.10.3.1 void SADBStringTable::GetByName (const char * *name*)

Get a stringtable object from the SADB server

Note

Just a wrapper for SetName, to be consistent with the GetByName function of standard blackboard objects.

Parameters

name	Desired name

2.10.3.2 void SADBStringTable::GetLatestValue (int *X*, int *Y*)

Get the latest value of a cell

Parameters

X	Column index
Y	Row index

2.10.3.3 char * SADBStringTable::GetNewString (int *X*, int *Y*)

Get a string in the table

Parameters

X	Column position
Y	Row Position

Returns

Pointer to the string (don't forget to free it) String to be set

2.10.3.4 const char * SADBStringTable::GetString (int *X*, int *Y*)

Get a string in the table

Parameters

X	Column position
Y	Row Position

Returns

Pointer to the string (don't forget to free it) String to be set

2.10.3.5 void SADBStringTable::SetBlackboard (Blackboard * *board*)

Attach the string table to a blackboard

Parameters

board	Blackboard to be used

2.10.3.6 void SADBStringTable::SetDescription (const char * *desc*)

Passtrough function that sets the description of the underlying object

Parameters

desc	Description

2.10.3.7 void SADBStringTable::SetDimensions (int *X*, int *Y*)

Set the size fo the string table (always 2D)

Note

Must set blackboard before allocating memory

Parameters

X	Number of columns
Y	Number of Rows

2.10.3.8 void SADBStringTable::SetName (const char * *name*)

Passtrough function that sets the name of the underlying object

Parameters

name	Name of the object

2.10.3.9 void SADBStringTable::SetString (int *X*, int *Y*, const char * *Str*)

Set a string in the table

Parameters

X	Column position
Y	Row Position
Str	String to be set

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

- SADB.h
- SADBStringTable.cpp

2.11 SADBStringTable Class Reference

`#include <SADBStringTable.h>`

Public Member Functions

- void **Initialize** (void)
- void **Free** (void)
- void **SetName** (const char *name)
- void **GetByName** (const char *name)
- void **SetDescription** (const char *desc)
- void **SetBlackboard** (Blackboard *board)
- void **SetDimensions** (int X, int Y)
- void **SetString** (int X, int Y, const char *Str)
- const char * **GetString** (int X, int Y)
- char * **GetNewString** (int X, int Y)
- void **LoadCSV** (const char *Filename)
- void **GetLatestValues** (void)
- void **GetLatestValue** (int X, int Y)

Public Attributes

- SADBu8 * Name

  Human Readable Object Name.
- BlackboardObject Object

Object to store the list of indices.

- int nCols

  Number of Columns.

- int nRows

  Number of Rows.

- Blackboard * Board

  Blackboard.

2.11.1 Detailed Description

Class used to store tables of strings in SADB

Note

If you change the dimensions of the table to a smaller size (shrink), some object will remain on the blackboard without being linked to a string table. If the table size is incremented at a later time, the objects will be linked again. This is to preserve history.

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

- SADBStringTable.h

2.12 Timestamp Class Reference

Public Member Functions

- void **Initialize** (void)
- void **Free** (void)
- void **Print** (void)
- void **Now** (void)
- void **Set** (SADBu64 second, SADBu16 millisecond)
- SADBdfl **SecondDifference** (Timestamp *B)
- SADBs64 **MillisecondDifference** (Timestamp *B)
- SADBdfl **Get** (void)
- void **Set** (SADBdfl time)

Public Attributes

- SADBu64 **Second**
- SADBu16 **Millisecond**

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

- Timestamp.h

2.13 SADB::Timestamp Class Reference

Public Member Functions

- Timestamp (void)
- ~Timestamp (void)

- void Initialize (void)

  Initializes object to 0.
- void Free (void)

  Frees allocated memory.
- void Print (void)

  Print the timestamp.
- void Now (void)

  Set the timestamp to the current time.
- void Set (SADBu64 second, SADBu16 millisecond)
- SADBdfl SecondDifference (Timestamp *B)
- SADBs64 MillisecondDifference (Timestamp *B)
- SADBdfl Get (void)

  Get the timestamp in SMC double format.
- void Set (SADBdfl time)

Public Attributes

- SADBu64 **Second**
- SADBu16 **Millisecond**

2.13.1 Constructor & Destructor Documentation

2.13.1.1 Timestamp::Timestamp (void)

Constructor

See also

Timestamp::Initialize

2.13.1.2 Timestamp::~Timestamp (void)

Destructor

See also

Timestamp::Free

2.13.2 Member Function Documentation

2.13.2.1 SADBs64 Timestamp::MillisecondDifference (Timestamp * *B*)

Get the number of seconds between the current timestamp object and another timestamp object

Parameters

B	The other timestamp

Note

If you have 2 timestamp objects A and B, A.SecondDifference(&B) will return B-A.
If the distance between timestamps is very large, there is a possibility of overflow. This should only happen if the number of seconds between the two timestamps is greater than ~ 1.8e16.

2.13.2.2 SADBdfl Timestamp::SecondDifference (Timestamp * *B*)

Get the number of seconds between the current timestamp object and another timestamp object

Parameters

B	The other timestamp

Note

If you have 2 timestamp objects A and B, A.SecondDifference(&B) will return B-A.
If the distance between timestamps is very large, there is a possibility of overflow. This should only happen if the number of seconds between the two timestamps is greater than $\sim$ 9e15.

2.13.2.3 void Timestamp::Set (SADBu64 *second,* SADBu16 *millisecond*)

Manually set the timestamp

Parameters

second	Target Time (second part)
millisecond	Target Time (millisecond part)

2.13.2.4 void Timestamp::Set (SADBdfl *time*)

Manually set the timestamp, in SMC double format

Parameters

time	Target Time

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

- SADB.h
- Timestamp.cpp

Index