Apèndix B

Documentació de l’API

B.0.1  UIParaverTrace Class Reference

Class interface read the values defined in the Paraver header.
#include <UIParaverTrace.h>

Public Member Functions

- UIParaverTrace (const string prvFile)
  
  This is the constructor. The system parses and store internally the data defined in the header of the Paraver trace file.

- virtual ~UIParaverTrace ()
  
  This is the destructor.

- string getTraceFile ()
  
  This method returns the path file to the Paraver trace file (.prv).

- unsigned int getNumberOfApplications () const
  
  This method returns the number of applications from the Paraver trace header.
• unsigned int getNumberOfTasks (const unsigned int appl) const
  *This method returns the number of Tasks for a certain Application.*

• unsigned int getNumberOfThreads (const unsigned int appl, const unsigned int task) const
  *This method returns the number of threads for a certain Application and Task.*

• unsigned int getNumberOfNodes () const
  *This method returns the number of nodes defined in the header of Paraver trace file (.prv).*

• unsigned int getNodeOfTask (const unsigned int appl, const unsigned int task) const
  *This method, for a certain task from a certain application, returns the node where has been executed the task.*

• unsigned int getNumberOfCPUFromNode (const unsigned int node) const
  *This method returns the number of CPUs belonging to a Node.*

• unsigned int getCPUFromNode (unsigned int node, unsigned int cpu) const
  *This method returns the CPU global ID from one CPU belonging to a node.*

B.0.1.1 Detailed Description

Class interface read the values defined in the Paraver header.

B.0.1.2 Constructor & Destructor Documentation

UIParaverTrace (const string *prvFile*)
This is the constructor. The system parses and store internally the data defined in the header of the Paraver trace file.
Parameters:

`prvFile` refers to the path of the Paraver trace file (.prv).

B.0.1.3 Member Function Documentation

`string getTraceFile ()`
This method returns the path file to the Paraver trace file (.prv).

Returns:
The path to the Paraver trace file (.prv).

`unsigned int getNumberOfApplications () const`
This method returns the number of applications from the Paraver trace header.

Returns:
The number of applications.

`unsigned int getNumberOfTasks (const unsigned int appl) const`
This method returns the number of Tasks for a certain Application.

Parameters:

`appl` This is the Application number to query.

Returns:
The number of Tasks for the application ‘appl’.
unsigned int getNumberOfThreads (const unsigned int appl, const unsigned int task) const

This method returns the number of threads for a certain Application and Task.

Parameters:

appl refers to the number of Application.
task refers to the Task to query.

Returns:

The number of threads for a certain application and task.

unsigned int getNumberOfNodes () const

This method returns the number of nodes defined in the header of Paraver trace file (.prv).

Returns:

The number of nodes defined in the trace.

unsigned int getNodeOfTask (const unsigned int appl, const unsigned int task) const

This method, for a certain task from a certain application, returns the node where has been executed the task.

Parameters:

appl The application to query.
task The task to query.

Returns:

The node where the task has been executed.
unsigned int getNumberOfCPUFromNode (const unsigned int node) const
This method returns the number of CPUs belonging to a Node.

Parameters:
    node The node to query.

Returns:
    The number of CPUs belonging to the node 'node'.

unsigned int getCPUFromNode (unsigned int node, unsigned int cpu) const
This method returns the CPU global ID from one CPU belonging to a node.

Parameters:
    node The node to query.
    cpu number from this node.

Returns:
    The number of CPU ID belonging to the node 'node'.

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

- UIParaverTrace.h

B.0.2 UIParaverTraceConfig Class Reference

Class interface to read the values defined in a Paraver Trace config file.

#include <UIParaverTraceConfig.h>
Public Member Functions

- `UIParaverTraceConfig (const string pcfFile)`
  
  This is the constructor. The system parses and store internally the data defined in the Paraver Config File.

- `virtual ~UIParaverTraceConfig ()`
  
  This is the destructor.

- `string getTraceConfigFile ()`
  
  This method returns the path to the Paraver Trace config file which this instance is working with.

- `int getEventType (const string eventTypeValue) const`
  
  This method returns the numeric key from a textual event type.

- `string getEventType (const int eventTypeKey) const`
  
  This method returns the textual description for event type key.

- `int getEventValue (const int eventTypeKey, const string evntValue) const`
  
  This method returns the numeric key for an event value that belongs to an event type key.

- `string getEventValue (const int eventTypeKey, const int eventValueKey) const`
  
  This method returns the textual description for an event value key that belongs to an event type key.

- `vector< unsigned int > getEventTypes () const`
  
  This method returns an integer vector with a list of event type keys.

- `vector< unsigned int > getEventValuesFromEventTypeKey (unsigned int eventTypeKey) const`
  
  This method returns an integer vector with a list of event value keys from an event type key.
B.0.2.1 Detailed Description

Class interface to read the values defined in a Paraver Trace config file.

B.0.2.2 Constructor & Destructor Documentation

UIParaverTraceConfig (const string pcfFile)
This is the constructor. The system parses and store internally the data
defined in the Paraver Config File.

Parameters:

pcfFile refers to the path to the Paraver Config File (.pcf).

B.0.2.3 Member Function Documentation

string getTraceConfigFile ()
This method returns the path to the Paraver Trace config file which this
instance is working with.

Returns:

The path to the Paraver Trace Config file (.pcf).

int getEventType (const string eventTypeValue) const
This method returns the numeric key from a textual event type.

Parameters:

eventTypeValue This is the textual description of the event type.
Returns:

The numeric key from the eventTypeValue.

```cpp
string getEventType (const int eventTypeKey) const
```
This method returns the textual description for event type key.

Parameters:

`eventTypeKey` This is the numerical key of the event type.

Returns:

The textual description from eventTypeValue.

```cpp
int getEventValue (const int eventTypeKey, const string eventNameValue) const
```
This method returns the numeric key for an event value that belongs to an event type key.

Parameters:

`eventTypeKey` This is the numerical key of the event type.
`eventNameValue` This is the textual description of the event value.

Returns:

The numeric key from an eventNameValue belonging to an eventTypeKey

```cpp
string getEventValue (const int eventTypeKey, const int eventNameValueKey) const
```
This method returns the textual description for an event value key that belongs to an event type key.
Parameters:

`eventTypeKey` This is the numerical ID of the event type.

`eventValueKey` This is the numeric ID of the event value.

Returns:

The textual description from the eventValue belonging to an eventTypeKey.

`vector<unsigned int> getEventTypes () const`

This method returns an integer vector with a list of event type keys.

Returns:

An integer vector with the list of event type keys.

`vector<unsigned int> getEventValuesFromEventTypeKey (unsigned int eventTypeKey) const`

This method returns an integer vector with a list of event value keys from an event type key.

Returns:

An integer vector with the list of event value keys.

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

- UIParaverTraceConfig.h

### B.0.3 UIParaverWindowConfig Class Reference

Class interface to extract data defined in a Paraver Window Config file.

```c++
#include <UIParaverWindowConfig.h>
```
Public Member Functions

- **UIParaverWindowConfig (string cfgFile)**
  
  _This is the constructor. The system parses and extract values defined in the Paraver window config file._

- **virtual ~UIParaverWindowConfig ()**
  
  _This is the destructor._

- **string getFile ()**
  
  _This method gets the path to the Paraver window config file that represents this object._

- **string getOutputFile ()**
  
  _This method returns a suggested path to an output file to be used by Paramedir._

- **unsigned int getNumberOfAnalyzers ()**
  
  _This method returns the number of analyzers defined in this Paraver window config file._

- **unsigned int getControlWindowIDFrom2D (unsigned int analyzer_id)**
  
  _This method gets the control window ID for a certain analyzer defined in this Paraver window config file._

- **unsigned int getDataWindowIDFrom2D (unsigned int analyzer_id)**
  
  _This method gets the data window ID for a certain analyzer defined in this Paraver window config file._

- **vector< unsigned int > getFilteredEventTypes (unsigned int window_id)**
  
  _This method gets the filtered event types from a display window._

### B.0.3.1 Detailed Description

Class interface to extract data defined in a Paraver Window Config file.
B.0.3.2 Constructor & Destructor Documentation

UIParaverWindowConfig (string *cfgFile*)
This is the constructor. The system parses and extract values defined in the Paraver window config file.

Parameters:

*cfgFile* refers to the path of a Paraver window config file (.cfg).

B.0.3.3 Member Function Documentation

string getFile ()
This method gets the path to the Paraver window config file that represents this object.

Returns:

Returns the path to the Paraver window config file (.cfg).

string getOutputFile ()
This method returns a suggested path to an output file to be used by Paramedir.

Returns:

The path to the Paramedir output file.
unsigned int getNumberOfAnalyzers ()
This method returns the number of analyzers defined in this Paraver window config file.

Returns:
The number of analyzers defined in this Paraver window config file.

unsigned int getControlWindowIDFrom2D (unsigned int analyzer_id)
This method gets the control window ID for a certain analyzer defined in this Paraver window config file.

Parameters:

analyzer_id This is the analyzer defined in this Paraver window config to make the query.

Returns:
The display window ID from the control window of the defined analyzer_id.

unsigned int getDataWindowIDFrom2D (unsigned int analyzer_id)
This methods gets the data window ID for a certain analyzer defined in this Paraver window config file.

Parameters:

analyzer_id This is the analyzer to make the query.

Returns:
The display window ID from the data window of the analyzer_id.
vector<unsigned int> getFilteredEventTypes (unsigned int window_id)

This method gets the filtered event types from a display window.

Parameters:

window_id This is the display window id to query.

Returns:

The event types that filtered in this window.

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

- UIParaverWindowConfig.h

B.0.4 UIParaver2DOutput Class Template Reference

Template Class to get the 2D data calculated by Paramedir.

#include <UIParaver2DOutput.h>

Public Member Functions

- double getThreadValue (string outputFilename, unsigned int appl, unsigned int task, unsigned int thread, unsigned int column)

  This method returns a single value from a Paramedir output file calculated at Thread level.

- vector<unsigned int> getThreadColumns (string outputFilename, unsigned int appl, unsigned int task, unsigned int thread)

  This method returns all the columns defined in a certain Paramedir output file at Thread level.

- vector<double> getThreadValues (string outputFilename, unsigned int appl, unsigned int task, unsigned int thread)

  This method returns all the values (without duplicates) in a certain Paramedir output file at Thread level.
• vector< double > getThreadFullValues (string outputFile, unsigned int appl, unsigned int task, unsigned int thread)

This method returns all the values in a certain Paramedir output file at Thread level.

• double getTaskValue (string outputFile, unsigned int appl, unsigned int task, unsigned int column)

This method returns a value in a Paramedir output file calculated at Task level.

• unsigned int getTaskColumns (string outputFile, unsigned int appl, unsigned int task)

This method returns all the columns defined in a certain Paramedir output file at Task level.

• double getApplValue (string outputFile, unsigned int appl, unsigned int column)

This method returns a value in a Paramedir output file calculated at Application level.

• vector< unsigned int > getApplColumns (string outputFile, unsigned int appl)

This method returns all the columns defined in a certain Paramedir output file at Application level.

• double getWorkloadValue (string outputFile, unsigned int column)

This method returns a value in a Paramedir output file calculated at Workload level.

• double getCpuValue (string outputFile, unsigned int node, unsigned int cpu, unsigned int column)

This method returns a value in a Paramedir output file calculated at CPU level.

• double getNodeValue (string outputFile, unsigned int node, unsigned int column)

This method returns a value in a Paramedir output file calculated at Node level.
• double getSystemValue (string outputFile, unsigned int column)

    This method returns a value in a Paramedir output file calculated at System level.

Static Public Member Functions

• static UIParaver2DOutput< DataManager > * getInstance ()

    This method initializes the DataManager and returns an instance of UIParaver2dOutput class.

B.0.4.1 Detailed Description

template<class DataManager> class domain::UIParaver2DOutput< DataManager >

Template Class to get the 2D data calculated by Paramedir.
This template class is instantiated by a class that implements the way to get the data. Currently just one DataManager has been developed:

• Paraver2DOutput The current version only provides basic functionalities.

B.0.4.2 Member Function Documentation

UIParaver2DOutput< DataManager > * getInstance () [inline, static]

This method initializes the DataManager and returns an instance of UIParaver2dOutput class.

Returns:

An instance of the UIParaver2dOutput class.
double getThreadValue (string outputFile, unsigned int appl, unsigned int task, unsigned int thread, unsigned int column) [inline]

This method returns a single value from a Paramedir output file calculated at Thread level.

Parameters:

outputFile Is the Paramedir output file where the data is stored.
appl Is the applications to query.
task Is the task to query.
thread Is the thread to query.
column Is the column to query.

Returns:

The value at thread level corresponding to the parameters appl, task, thread, column.

vector< unsigned int > getThreadColumns (string outputFile, unsigned int appl, unsigned int task, unsigned int thread) [inline]

This method returns all the columns defined in a certain Paramedir output file at Thread level.

Parameters:

outputFile Is the Paramedir output file where the data is stored.
appl Is the applications to query.
task Is the task to query.
thread Is the thread to ask for.

Returns:

A vector with the column values at thread level corresponding to the parameters appl, task, thread.
vector< double > getThreadValues (string outputFile, unsigned int appl, unsigned int task, unsigned int thread) [inline]

This method returns all the values (without duplicates) in a certain Paramedir output file at Thread level.

Parameters:

- **outputFile** Is the Paramedir output file where the data is stored.
- **appl** Is the applications to query.
- **task** Is the task to query.
- **thread** Is the thread to ask for.

Returns:

A vector with all the values at thread level corresponding to the parameters appl, task, thread. (Without duplicates).

vector< double > getThreadFullValues (string outputFile, unsigned int appl, unsigned int task, unsigned int thread) [inline]

This method returns all the values in a certain Paramedir output file at Thread level.

Parameters:

- **outputFile** Is the Paramedir output file where the data is stored.
- **appl** Is the applications to query.
- **task** Is the task to query.
- **thread** Is the thread to ask for.

Returns:

A vector with all the values at thread level corresponding to the parameters appl, task, thread.
double getTaskValue (string outputFile, unsigned int appl, unsigned int task, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at Task level.

Parameters:

  outputFile Is the Paramedir output file where the data is stored.
  appl Is the applications to query.
  task Is the task to query.
  column Is the column to query.

Returns:

  The value at Task level corresponding to the parameters appl, task, column.

unsigned int getTaskColumns (string outputFile, unsigned int appl, unsigned int task) [inline]

This method returns all the columns defined in a certain Paramedir output file at Task level.

Parameters:

  outputFile Is the Paramedir output file where the data is stored.
  appl Is the applications to query.
  task Is the task to query.

Returns:

  A vector with the column values at Task level corresponding to the parameters appl, task.
double getApplValue (string outputFile, unsigned int appl, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at Application level.

Parameters:

   outputFile Is the Paramedir output file where the data is stored.
   appl Is the applications to query.
   column Is the column to query.

Returns:

   The value at Application level corresponding to the parameters appl, column.

vector< unsigned int > getApplColumns (string outputFile, unsigned int appl) [inline]

This method returns all the columns defined in a certain Paramedir output file at Application level.

Parameters:

   outputFile Is the Paramedir output file where the data is stored.
   appl Is the applications to query.

Returns:

   A vector with the column values at Application level corresponding to the parameter appl.

double getWorkloadValue (string outputFile, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at Workload level.
Parameters:

- `outputFile` Is the Paramedir output file where the data is stored.
- `column` Is the column to query.

Returns:

The value at Workload level corresponding to the parameter column.

double getCpuValue (string outputFile, unsigned int node, unsigned int cpu, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at CPU level.

Parameters:

- `outputFile` Is the Paramedir output file where the data is stored.
- `node` Is the node ask for.
- `cpu` The value at CPU level corresponding to the parameter node.
- `column` Is the column to query.

Returns:

The value at CPU level corresponding to the parameters node, cpu, column.

double getNodeValue (string outputFile, unsigned int node, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at Node level.

Parameters:

- `outputFile` Is the Paramedir output file where the data is stored.
- `node` Is the node to query.
**column** Is the column to query.

**Returns:**

The value at Node level corresponding to the parameters node, column.

double getSystemValue (string outputFile, unsigned int column) [inline]

This method returns a value in a Paramedir output file calculated at System level.

**Parameters:**

outputFile Is the Paramedir output file where the data is stored.

column Is the column to query.

**Returns:**

The value at System level corresponding to the parameter column.

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

- UIParaver2DOutput.h

### B.0.5 UIParamedir Class Reference

Class interface to launch Paramedir with a set of Paraver window config files.

```c
#include <UIParamedir.h>
```

**Public Member Functions**

- UIParamedir (string prvFile)
  
  This is the constructor.
• virtual ~UIParamedir ()
  
  *This is the destructor.*

• void addCf{g} (pair< string, string > cf{g}File)
  
  *This method adds a cf{g} file to be calculated with Paramedir.*

• void addCf{g}s (vector< pair< string, string > > cf{g}s)
  
  *This method adds a vector of cf{g} files to be calculated with Paramedir.*

• vector< pair< string, string > > getCF{g}s ()
  
  *This method returns a copy the paths of the Paraver window config files that are going to be calculated with Paramedir, and its outputs paths.*

• void spawn ()
  
  *This method executes Paramedir with the Paraver config files defined previously.*

### B.0.5.1 Detailed Description

Class interface to launch Paramedir with a set of Paraver window config files. In order to call 'paramedir':

• The binary must be located in the $PATH environment variable.

• The environment variable $PARAVER_HOME must be defined correctly.

### B.0.5.2 Constructor & Destructor Documentation

**UIParamedir** (string *prvFile*)

This is the constructor.

**Parameters:**

*prvFile* Is the path to a Paraver trace file (.prv).
B.0.5.3 Member Function Documentation

void addCfg (pair< string, string > cfgFile)
This method adds a cfg file to be calculated with Paramedir.

Parameters:

    cfgFile  Is a pair of strings. The first one is the path to the cfg file,
             and the second is the path to the output file where Paramedir will
             write the results.

void addCfgs (vector< pair< string, string > > cfgs)
This method adds a vector of cfg files to be calculated with Paramedir.

Parameters:

    cfgs  This vector of pairs of strings will be added to the system to be
           executed by Paramedir.

vector<pair<string, string>> > getCfgs ()
This method returns a copy the paths of the Paraver window config files that
are going to be calculated with Paramedir, and its outputs paths.

Returns:

    A vector with pairs of Paraver window config files and its output files.

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

- UIParamedir.h
B.0.6 UIParaver Class Reference

Class interface to control some functionalities of Paraver.

#include <UIParaver.h>

Public Member Functions

- void sendSignal (string cfgFile, unsigned long long starttime, unsigned long long endtime)

  *This method gets the Paraver PID and creates the $HOME/paraloo.paraload.sig containing the parameters defined in this method. Afterwords, sends a signal to Paraver.*

Static Public Member Functions

- static UIParaver * getInstance ()

  *This method calls the constructor and return the current instance (singleton).*

Protected Member Functions

- UIParaver ()

  *This is the constructor.*

- virtual ~UIParaver ()

  *This is the destructor.*

B.0.6.1 Detailed Description

Class interface to control some functionalities of Paraver.

B.0.6.2 Constructor & Destructor Documentation
UIParaver () [protected]
This is the constructor.

Note:
Does not open an instance of Paraver.

B.0.6.3 Member Function Documentation

static UIParaver* getInstance () [static]
This method calls the constructor and return the current instance (singleton).

Note:
Does not open an instance of Paraver.

void sendSignal (string cfgFile, unsigned long long starttime, unsigned long long endtime)
This method gets the Paraver PID and creates the $HOME/paraload.sig containing the parameters defined in this method. Afterwards, sends a signal to Paraver.

Parameters:
- **cfgFile** Is the path to a Paraver window config file.
- **starttime** Is the start time to show in Paraver. (If it is unknown just write 0)
- **endtime** Is the end time to show in Paraver. (If it is unknown just write 0)

The documentation for this class was generated from the following file:
- UIParaver.h