

Intel® Cluster Checker API

Generated by Doxygen 1.6.1

Tue Oct 24 14:23:45 2017

Contents

1	Disclaimer and Legal Information	1
2	Class Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Class Documentation	7
4.1	clk::Layer::CollectConfig Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	8
4.1.2.1	cluster_permutations	8
4.1.2.2	database_close_delay	8
4.1.2.3	ignore_subclusters	8
4.1.2.4	network_interface	8
4.1.2.5	provider_async_output_dir	8
4.1.2.6	provider_aux_path	8
4.1.2.7	provider_config_params	8
4.1.2.8	provider_path	8
4.1.2.9	run_all_providers	8
4.1.2.10	run_async	9
4.1.2.11	scale	9
4.1.2.12	shared_dir	9

4.1.2.13	timeout	9
4.2	clk::Layer::Config Struct Reference	10
4.2.1	Detailed Description	10
4.2.2	Constructor & Destructor Documentation	10
4.2.2.1	Config	10
4.2.2.2	Config	11
4.2.3	Member Data Documentation	11
4.2.3.1	bin_path	11
4.2.3.2	collect_config	11
4.2.3.3	config_params	11
4.2.3.4	db_configs	11
4.2.3.5	expiration	11
4.2.3.6	fwd_file_list	11
4.2.3.7	fwds	11
4.2.3.8	install_path	12
4.2.3.9	language	12
4.2.3.10	node_source	12
4.2.3.11	nodes	12
4.2.3.12	now	12
4.2.3.13	provider_path	12
4.2.3.14	providers	12
4.3	clk::Layer::ConfigParam Struct Reference	13
4.3.1	Detailed Description	13
4.3.2	Member Data Documentation	13
4.3.2.1	key	13
4.3.2.2	module	13
4.3.2.3	provider	13
4.3.2.4	provider_optionset	13
4.3.2.5	values	13
4.4	clk::Layer::DatabaseConfig Struct Reference	14
4.4.1	Detailed Description	14

4.4.2	Member Data Documentation	14
4.4.2.1	name	14
4.4.2.2	source	14
4.4.2.3	type	14
4.5	clk::Diagnosis Class Reference	15
4.5.1	Detailed Description	15
4.6	clk::Fault Class Reference	16
4.6.1	Detailed Description	16
4.6.2	Member Data Documentation	16
4.6.2.1	confidence	16
4.6.2.2	id	16
4.6.2.3	msg	17
4.6.2.4	nodes	17
4.6.2.5	remedy	17
4.6.2.6	rowid	17
4.6.2.7	severity	17
4.6.2.8	suppressed	17
4.7	clk::Layer::Filter Struct Reference	18
4.7.1	Detailed Description	18
4.7.2	Member Data Documentation	18
4.7.2.1	confidence	18
4.7.2.2	ids	18
4.7.2.3	nodes	18
4.7.2.4	severity	18
4.7.2.5	state	19
4.7.2.6	suppressed	19
4.7.2.7	type	19
4.8	clk::Layer::Fwd Struct Reference	20
4.8.1	Detailed Description	20
4.8.2	Constructor & Destructor Documentation	20
4.8.2.1	Fwd	20

4.8.2.2	Fwd	20
4.8.3	Member Data Documentation	21
4.8.3.1	extension_mods	21
4.8.3.2	extension_path	21
4.8.3.3	kb_mods	21
4.8.3.4	kb_path	21
4.8.3.5	name	21
4.8.3.6	provider_aux_path	21
4.8.3.7	provider_path	21
4.8.3.8	providers	22
4.9	clk::Layer Class Reference	23
4.9.1	Detailed Description	24
4.9.2	Member Enumeration Documentation	24
4.9.2.1	ScaleType	24
4.9.3	Constructor & Destructor Documentation	25
4.9.3.1	Layer	25
4.9.3.2	~Layer	25
4.9.4	Member Function Documentation	25
4.9.4.1	analyze	25
4.9.4.2	collect	25
4.9.4.3	get_faults	26
4.9.4.4	get_messages	26
4.9.4.5	get_nodes	26
4.9.4.6	get_version_number	26
4.9.4.7	progress	27
4.9.5	Member Data Documentation	27
4.9.5.1	message	27
4.10	clk::Layer::Message Struct Reference	28
4.10.1	Detailed Description	28
4.10.2	Member Data Documentation	28
4.10.2.1	level	28

4.10.2.2	msg	28
4.11	clk::Node Class Reference	29
4.11.1	Detailed Description	29
4.11.2	Member Enumeration Documentation	29
4.11.2.1	role_t	29
4.11.3	Member Data Documentation	29
4.11.3.1	name	29
4.11.3.2	roles	29
4.11.3.3	subcluster	30
4.12	clk::Layer::ProviderConfigParam Struct Reference	31
4.12.1	Detailed Description	31
4.12.2	Member Data Documentation	31
4.12.2.1	provider	31
4.12.2.2	provider_optionsets	31
4.13	clk::Layer::ProviderOptionset Struct Reference	32
4.13.1	Detailed Description	32
4.13.2	Member Data Documentation	32
4.13.2.1	id	32
4.13.2.2	provider_option_vars	32
4.14	clk::Sign Class Reference	33
4.14.1	Detailed Description	33
4.14.2	Member Data Documentation	33
4.14.2.1	state	33
4.15	clk::Layer::Sorting Struct Reference	34
4.15.1	Detailed Description	34
4.15.2	Member Enumeration Documentation	34
4.15.2.1	FIELD	34
4.15.3	Constructor & Destructor Documentation	34
4.15.3.1	Sorting	34
4.15.4	Member Data Documentation	34
4.15.4.1	ascending	34

4.15.4.2	field	35
4.16	clk::Layer::Suppression Struct Reference	36
4.16.1	Detailed Description	36
4.16.2	Member Data Documentation	36
4.16.2.1	confidence	36
4.16.2.2	id	36
4.16.2.3	node	36
4.16.2.4	severity	36

Chapter 1

Disclaimer and Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps information.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Xeon, Intel Xeon Phi, Cilk, and VTune are trademarks of Intel Corporation in the U.S. and/or other countries.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

* Other names and brands may be claimed as the property of others.

Copyright ©2017 Intel Corporation.

Chapter 2

Class Index

2.1 Class Hierarchy

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

clk::Layer::CollectConfig	7
clk::Layer::Config	10
clk::Layer::ConfigParam	13
clk::Layer::DatabaseConfig	14
clk::Fault	16
clk::Diagnosis	15
clk::Sign	33
clk::Layer::Filter	18
clk::Layer::Fwd	20
clk::Layer	23
clk::Layer::Message	28
clk::Node	29
clk::Layer::ProviderConfigParam	31
clk::Layer::ProviderOptionset	32
clk::Layer::Sorting	34
clk::Layer::Suppression	36

Chapter 3

Class Index

3.1 Class List

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

clk::Layer::CollectConfig (Data collect configuration)	7
clk::Layer::Config (Layer configuration options)	10
clk::Layer::ConfigParam (Data analysis configuration parameter)	13
clk::Layer::DatabaseConfig (A database used for collection and analysis) . .	14
clk::Diagnosis (A diagnosis is the root cause of an issue. Diagnosis is derived from the Fault class)	15
clk::Fault (A fault is the basic analysis unit. A fault is either a sign (i.e., observation) or a diagnosis (i.e., root cause))	16
clk::Layer::Filter (Filter for the list of faults returned by get_faults())	18
clk::Layer::Fwd (Framework definition configuration options)	20
clk::Layer (The presentation layer)	23
clk::Layer::Message (Internal Layer messages for the caller to handle) . . .	28
clk::Node (The node container)	29
clk::Layer::ProviderConfigParam (Data collect configuration parameter) . .	31
clk::Layer::ProviderOptionset (Data collect configuration for option set) . .	32
clk::Sign (A sign is an observation of an issue. Sign is derived from the Fault class)	33
clk::Layer::Sorting (Sort order for the list of faults returned by get_faults())	34
clk::Layer::Suppression (Suppress faults matching the specified values) . .	36

Chapter 4

Class Documentation

4.1 `clk::Layer::CollectConfig` Struct Reference

Data collect configuration.

```
#include <clk.h>
```

Public Attributes

- bool `ignore_subclusters` = false
- bool `run_all_providers` = false
- bool `run_async` = false
- char `provider_async_output_dir` [PATH_MAX]
- char `provider_path` [PATH_MAX]
- char `provider_aux_path` [PATH_MAX]
- unsigned int `cluster_permutations` = 1
- int `database_close_delay` = 1
- enum `ScaleType scale` = `ScaleType::SCALE_CONSTANT`
- std::string `network_interface` = ""
- std::string `shared_dir`
- time_t `timeout` = 60
- std::vector< `ProviderConfigParam` > `provider_config_params`

4.1.1 Detailed Description

Data collect configuration.

4.1.2 Member Data Documentation

4.1.2.1 `unsigned int clk::Layer::CollectConfig::cluster_permutations = 1`

Cluster permutations, 0 means all.

4.1.2.2 `int clk::Layer::CollectConfig::database_close_delay = 1`

Time to delay the closing of database connection in seconds.

4.1.2.3 `bool clk::Layer::CollectConfig::ignore_subclusters = false`

Ignores the sub-clustering of nodes.

4.1.2.4 `std::string clk::Layer::CollectConfig::network_interface = ""`

Network interface.

4.1.2.5 `char clk::Layer::CollectConfig::provider_async_output_dir[PATH_MAX]`

Provider Asynchronous Output Directory.

4.1.2.6 `char clk::Layer::CollectConfig::provider_aux_path[PATH_MAX]`

Provider Auxiliary Path.

4.1.2.7 `std::vector<ProviderConfigParam>` `clk::Layer::CollectConfig::provider_config_params`

Data collect configuration parameters

4.1.2.8 `char clk::Layer::CollectConfig::provider_path[PATH_MAX]`

Provider [Config](#) Directory Path.

4.1.2.9 `bool clk::Layer::CollectConfig::run_all_providers = false`

Runs all providers specified at provider_config_dir location.

4.1.2.10 bool clk::Layer::CollectConfig::run_async = false

Runs asynchronously when set to true for ORCM.

4.1.2.11 enum ScaleType clk::Layer::CollectConfig::scale = ScaleType::SCALE_CONSTANT

The scale attribute specifies the rate at which the timeout value should increase based on the number of nodes. Default is set to constant.

4.1.2.12 std::string clk::Layer::CollectConfig::shared_dir

Shared Directory to store intermittent files.

Note: Currently, this variable is not configurable. Internally, the value is defaulted to \$HOME/.clk

4.1.2.13 time_t clk::Layer::CollectConfig::timeout = 60

Time out for the Provider. Default is 60 seconds.

4.2 clk::Layer::Config Struct Reference

[Layer](#) configuration options.

```
#include <clk.h>
```

Public Types

- enum { DATABASE, NODELIST, INTERSECTION, UNION }

Public Member Functions

- [Config](#) ()
- [Config](#) (const std::vector< [DatabaseConfig](#) > &db_configs)

Public Attributes

- [CollectConfig](#) collect_config
- std::vector< [Fwd](#) > fwds
- std::vector< std::string > fwd_file_list
- std::string bin_path
- std::vector< [DatabaseConfig](#) > db_configs
- time_t expiration = 0
- std::string install_path
- std::string language
- std::vector< [Node](#) > nodes
- enum clk::Layer::Config:: { ... } node_source
- time_t now = time(NULL)
- std::vector< std::string > providers
- std::string provider_path
- std::vector< [ConfigParam](#) > config_params

4.2.1 Detailed Description

[Layer](#) configuration options. Additional configuration options are likely to be added in the future, as needed.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 clk::Layer::Config::Config ()

Default constructor

4.2.2.2 clk::Layer::Config::Config (const std::vector< DatabaseConfig > & db_configs)

Constructor

Parameters:

db_configs List of database configurations

4.2.3 Member Data Documentation

4.2.3.1 std::string clk::Layer::Config::bin_path

Optional path to the binary directory. If it is not specified, then the bin path under the install path will be used.

4.2.3.2 CollectConfig clk::Layer::Config::collect_config

Collect configuration

4.2.3.3 std::vector<ConfigParam> clk::Layer::Config::config_params

Data analysis configuration parameters

4.2.3.4 std::vector<DatabaseConfig> clk::Layer::Config::db_configs

List of database configurations

4.2.3.5 time_t clk::Layer::Config::expiration = 0

Maximum allowable age of data, relative to the value of now. Data older than now minus expiration will be ignored. A value of 0 means to use all data, i.e., there is no expiration.

4.2.3.6 std::vector<std::string> clk::Layer::Config::fwd_file_list

List of Framework definition files to be loaded

4.2.3.7 std::vector<Fwd> clk::Layer::Config::fwds

List of Framework definitions

4.2.3.8 `std::string clk::Layer::Config::install_path`

Optional path of the Intel(R) Cluster Checker installation. If is is not specified, then the default install path will be used.

4.2.3.9 `std::string clk::Layer::Config::language`

String representing the message catalog language

4.2.3.10 `enum { ... } clk::Layer::Config::node_source`

Select the source for the list of nodes. Options are to use the database (exclusively), the input [nodes](#) vector (exclusively), or the intersection or union of the two.

4.2.3.11 `std::vector<Node> clk::Layer::Config::nodes`

List of nodes to be analyzed (see [node_source](#)).

4.2.3.12 `time_t clk::Layer::Config::now = time(NULL)`

Reference time to be used as the current time for analysis

4.2.3.13 `std::string clk::Layer::Config::provider_path`

Optional path to the provider XML files. If it is not specified then the provider path under the install path will be used.

4.2.3.14 `std::vector<std::string> clk::Layer::Config::providers`

List of providers to run. The providers are XML files that are located in the provider path. When specifying a provider, do not include the .xml extension. If a user intends to run a provider defined in dgemm.xml, then just append dgemm to the list of providers.

4.3 clk::Layer::ConfigParam Struct Reference

Data analysis configuration parameter.

```
#include <clk.h>
```

Public Attributes

- std::string [module](#)
- std::string [provider](#)
- std::string [provider_optionset](#)
- std::string [key](#)
- std::vector< std::string > [values](#)

4.3.1 Detailed Description

Data analysis configuration parameter. Note: For collect configuration, use [Provider-ConfigParam](#) struct

4.3.2 Member Data Documentation

4.3.2.1 std::string clk::Layer::ConfigParam::key

Parameter name

4.3.2.2 std::string clk::Layer::ConfigParam::module

Module name in the knowledge base

4.3.2.3 std::string clk::Layer::ConfigParam::provider

Provider name for data collection

4.3.2.4 std::string clk::Layer::ConfigParam::provider_optionset

Provider optionset

4.3.2.5 std::vector<std::string> clk::Layer::ConfigParam::values

Parameter value(s)

4.4 clk::Layer::DatabaseConfig Struct Reference

A database used for collection and analysis.

```
#include <clk.h>
```

Public Attributes

- [DatabaseType](#) type
- std::string [name](#)
- std::string [source](#)

4.4.1 Detailed Description

A database used for collection and analysis.

4.4.2 Member Data Documentation

4.4.2.1 std::string clk::Layer::DatabaseConfig::name

Database name

4.4.2.2 std::string clk::Layer::DatabaseConfig::source

Database source

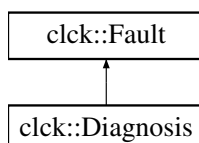
4.4.2.3 DatabaseType clk::Layer::DatabaseConfig::type

Database type

4.5 clk::Diagnosis Class Reference

A diagnosis is the root cause of an issue. [Diagnosis](#) is derived from the [Fault](#) class.

`#include <clk.h>`Inheritance diagram for `clk::Diagnosis`:



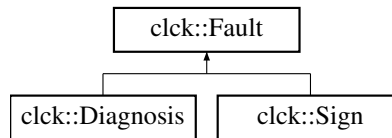
4.5.1 Detailed Description

A diagnosis is the root cause of an issue. [Diagnosis](#) is derived from the [Fault](#) class.

4.6 clk::Fault Class Reference

A fault is the basic analysis unit. A fault is either a sign (i.e., observation) or a diagnosis (i.e., root cause).

#include <clk.h> Inheritance diagram for clk::Fault:



Public Attributes

- int [confidence](#) = 0
- std::string [id](#)
- std::string [msg](#)
- std::vector< std::string > [nodes](#)
- std::string [remedy](#)
- int [severity](#) = 0
- bool [suppressed](#) = false
- std::set< int > [rowid](#)

4.6.1 Detailed Description

A fault is the basic analysis unit. A fault is either a sign (i.e., observation) or a diagnosis (i.e., root cause).

4.6.2 Member Data Documentation

4.6.2.1 int clk::Fault::confidence = 0

Confidence percentage (0 - 100)

4.6.2.2 std::string clk::Fault::id

Message catalog id

4.6.2.3 std::string clk::Fault::msg

Expanded message string

4.6.2.4 std::vector<std::string> clk::Fault::nodes

List of nodes

4.6.2.5 std::string clk::Fault::remedy

Expanded remedy string

4.6.2.6 std::set<int> clk::Fault::rowid

DB rows that provide the raw data upon which the sign/diagnosis is based

4.6.2.7 int clk::Fault::severity = 0

Severity percentage (0 - 100)

4.6.2.8 bool clk::Fault::suppressed = false

True if the diagnosis / sign is suppressed, false otherwise

4.7 clk::Layer::Filter Struct Reference

[Filter](#) for the list of faults returned by [get_faults\(\)](#).

```
#include <clk.h>
```

Public Attributes

- int [confidence](#) = 0
- std::vector< std::string > [ids](#)
- std::vector< std::string > [nodes](#)
- int [severity](#) = 0
- std::bitset< 2 > [state](#) = CLK_FAULT_STATE_DIAGNOSED | CLK_FAULT_STATE_OBSERVED
- std::bitset< 2 > [suppressed](#) = CLK_FAULT_SUPPRESSED_FALSE
- std::bitset< 2 > [type](#) = CLK_FAULT_TYPE_DIAGNOSIS | CLK_FAULT_TYPE_SIGN

4.7.1 Detailed Description

[Filter](#) for the list of faults returned by [get_faults\(\)](#).

4.7.2 Member Data Documentation

4.7.2.1 int clk::Layer::Filter::confidence = 0

Select faults with a greater than or equal to confidence value.

4.7.2.2 std::vector<std::string> clk::Layer::Filter::ids

Select faults corresponding to at least one of the ids. If empty, does not filter on id.

4.7.2.3 std::vector<std::string> clk::Layer::Filter::nodes

Selects faults corresponding to at least one of the nodes. If empty, does not filter on node.

4.7.2.4 int clk::Layer::Filter::severity = 0

Select faults with a greater than or equal to severity value.

**4.7.2.5 std::bitset<2> clk::Layer::Filter::state = CLCK_FAULT_STATE_-
DIAGNOSED | CLCK_FAULT_STATE_OBSERVED**

Select faults with a matching state. Only applies to signs, not diagnoses.

**4.7.2.6 std::bitset<2> clk::Layer::Filter::suppressed =
CLCK_FAULT_SUPPRESSED_FALSE**

Select faults with a matching suppression value.

**4.7.2.7 std::bitset<2> clk::Layer::Filter::type =
CLCK_FAULT_TYPE_DIAGNOSIS | CLCK_FAULT_TYPE_SIGN**

Select faults with a matching type.

4.8 clk::Layer::Fwd Struct Reference

Framework definition configuration options.

```
#include <clk.h>
```

Public Member Functions

- [Fwd](#) ()
- [Fwd](#) (const std::vector< std::string > &[providers](#), const std::set< std::string > &[extension_mods](#), const std::vector< std::string > &[kb_mods](#), const std::string &[provider_path](#)="", const std::string &[provider_aux_path](#)="", const std::string &[extension_path](#)="", const std::string &[kb_path](#)="")

Public Attributes

- std::string [name](#)
- std::vector< std::string > [providers](#)
- std::string [provider_path](#)
- std::string [provider_aux_path](#)
- std::set< std::string > [extension_mods](#)
- std::string [extension_path](#)
- std::vector< std::string > [kb_mods](#)
- std::string [kb_path](#)

4.8.1 Detailed Description

Framework definition configuration options.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 clk::Layer::Fwd::Fwd ()

Default constructor

4.8.2.2 clk::Layer::Fwd::Fwd (const std::vector< std::string > & *providers*, const std::set< std::string > & *extension_mods*, const std::vector< std::string > & *kb_mods*, const std::string & *provider_path* = "", const std::string & *provider_aux_path* = "", const std::string & *extension_path* = "", const std::string & *kb_path* = "")

Constructor

Parameters:

providers List of providers to be loaded
extension_mods List of connector extensions to be loaded
kb_mods List of knowledge base files to be loaded
provider_path Absolute path to the provider directory
provider_aux_path Absolute path to the provider auxiliary directory
extension_path Absolute path to the connector extension directory
kb_path Absolute path to the knowledge base directory

4.8.3 Member Data Documentation**4.8.3.1 std::set<std::string> clk::Layer::Fwd::extension_mods**

List of connector extensions to be loaded

4.8.3.2 std::string clk::Layer::Fwd::extension_path

Absolute path to the connector extension directory

4.8.3.3 std::vector<std::string> clk::Layer::Fwd::kb_mods

List of knowledge base files to be loaded

4.8.3.4 std::string clk::Layer::Fwd::kb_path

Absolute path to the knowledge base directory

4.8.3.5 std::string clk::Layer::Fwd::name

A unique Framework definition identifier.

4.8.3.6 std::string clk::Layer::Fwd::provider_aux_path

Absolute path to the provider auxiliary directory

4.8.3.7 std::string clk::Layer::Fwd::provider_path

Absolute path to the provider directory

4.8.3.8 `std::vector<std::string> clk::Layer::Fwd::providers`

List of providers to be loaded

4.9 clk::Layer Class Reference

The presentation layer.

```
#include <clk.h>
```

Classes

- struct [CollectConfig](#)
Data collect configuration.
- struct [Config](#)
Layer configuration options.
- struct [ConfigParam](#)
Data analysis configuration parameter.
- struct [DatabaseConfig](#)
A database used for collection and analysis.
- struct [Filter](#)
Filter for the list of faults returned by `get_faults()`.
- struct [Fwd](#)
Framework definition configuration options.
- struct [Message](#)
Internal Layer messages for the caller to handle.
- struct [ProviderConfigParam](#)
Data collect configuration parameter.
- struct [ProviderOptionset](#)
Data collect configuration for option set.
- struct [Sorting](#)
Sort order for the list of faults returned by `get_faults()`.
- struct [Suppression](#)
Suppress faults matching the specified values.

Public Types

- enum [ScaleType](#) { [SCALE_CONSTANT](#) = 0, [SCALE_LINEAR](#), [SCALE_SQUARED](#), [SCALE_LOGARITHMIC](#) }

The scale attribute specifies the rate at which the timeout value should increase based on the number of nodes. Valid options are: constant, linear, squared, logarithmic.

- enum [DatabaseType](#) { [SQLITE](#) = 0, [ODBC](#) }

Supported database types.

Public Member Functions

- [Layer](#) (const [Config](#) &config)
- [~Layer](#) ()
- bool [analyze](#) (const std::vector< [Suppression](#) > &suppressions)
- bool [collect](#) ()
- std::vector< std::shared_ptr< [Fault](#) > > [get_faults](#) (const [Filter](#) &filter, const std::vector< [Sorting](#) > &sorting)
- std::vector< [Message](#) > [get_messages](#) ()
- std::vector< [Node](#) > [get_nodes](#) ()
- int [get_version_number](#) ()
- bool [progress](#) (unsigned long &remaining, unsigned long &completed)

Public Attributes

- std::condition_variable [message](#)

4.9.1 Detailed Description

The presentation layer.

4.9.2 Member Enumeration Documentation

4.9.2.1 enum `clk::Layer::ScaleType`

The scale attribute specifies the rate at which the timeout value should increase based on the number of nodes. Valid options are: constant, linear, squared, logarithmic.

Enumerator:

SCALE_CONSTANT The "constant" attribute value does not scale the timeout with the number of nodes used.

SCALE_LINEAR The "linear" attribute value scales linearly with the number of nodes (e.g. $\text{base} * \text{num_nodes}$).

SCALE_SQUARED The "squared" attribute value scales with the number of nodes squared (e.g. $\text{base} * \text{num_nodes}^2$).

SCALE_LOGARITHMIC The "logarithmic" tag scales logarithmically with the number of nodes (e.g. $\text{base} * \ln((e-1) + \text{num_nodes})$).

4.9.3 Constructor & Destructor Documentation

4.9.3.1 clk::Layer::Layer (const Config & config)

Layer constructor

4.9.3.2 clk::Layer::~~Layer ()

Layer destructor

4.9.4 Member Function Documentation

4.9.4.1 bool clk::Layer::analyze (const std::vector< Suppression > & suppressions)

Start the analysis. Note: behavior is undefined if invoked more than once per Layer instance.

Returns:

True if the analysis was completed successfully, false otherwise. Note: this does NOT reflect the state of the cluster.

4.9.4.2 bool clk::Layer::collect ()

Collect data.

Returns:

True if the data collection was completed successfully, false otherwise.

4.9.4.3 `std::vector<std::shared_ptr<Fault> > clk::Layer::get_faults (const Filter &filter, const std::vector< Sorting > &sorting)`

Returns a list of faults. May be called multiple times. While [analyze\(\)](#) is active, presumably in another thread, the behavior is undefined. Use [progress\(\)](#) to determine current analysis status. If called before [analyze\(\)](#), returns an empty list.

Parameters:

filter Return only faults that match the filter

sorting List of sorting criteria. The first element is the primary sorting criterion, the second element the secondary sorting criterion, etc. If empty, the faults are returned unsorted.

Returns:

A list of faults

4.9.4.4 `std::vector<Message> clk::Layer::get_messages ()`

Returns a list of messages generated internal to [Layer](#) for the caller to handle.

Returns:

A list of messages

4.9.4.5 `std::vector<Node> clk::Layer::get_nodes ()`

Returns the list of nodes to be analyzed.

Returns:

A list of nodes.

4.9.4.6 `int clk::Layer::get_version_number ()`

Returns the version of the [Layer](#) API.

Returns:

The version number. Version X.Y.Z is represented as $(X*1000000 + Y*1000 + Z)$.

4.9.4.7 bool clk::Layer::progress (unsigned long & *remaining*, unsigned long & *completed*)

While [analyze\(\)](#) is active, presumably in another thread, returns the number of rules remaining to be fired and the number of rules already run. If called before [analyze\(\)](#), both values will be 0.

Parameters:

remaining The number of rules remaining to be fired. Not guaranteed to be monotonic. Returned by reference.

completed The number of rules that have been fired. Will be monotonic. Returned by reference.

Returns:

False if [analyze\(\)](#) has not yet been called, true otherwise.

4.9.5 Member Data Documentation

4.9.5.1 std::condition_variable clk::Layer::message

Notify when a new internal [Layer](#) message is available

4.10 clk::Layer::Message Struct Reference

Internal [Layer](#) messages for the caller to handle.

```
#include <clk.h>
```

Public Attributes

- int [level](#)
- std::string [msg](#)

4.10.1 Detailed Description

Internal [Layer](#) messages for the caller to handle.

4.10.2 Member Data Documentation

4.10.2.1 int clk::Layer::Message::level

[Message](#) level (priority). Inherits logging levels from syslog.

4.10.2.2 std::string clk::Layer::Message::msg

[Message](#) string

4.11 clk::Node Class Reference

The node container.

```
#include <clk.h>
```

Public Types

- enum [role_t](#) {
 ROLE_BOOT, ROLE_COMPUTE, ROLE_ENHANCED, ROLE_-
 EXTERNAL,
 ROLE_HEAD, ROLE_JOB_SCHEDULE, ROLE_LOGIN, ROLE_-
 NETWORK_ADDRESS,
 ROLE_STORAGE }

Public Attributes

- std::string [subcluster](#)
- std::string [name](#)
- std::vector< [role_t](#) > [roles](#)

4.11.1 Detailed Description

The node container.

4.11.2 Member Enumeration Documentation

4.11.2.1 enum clk::Node::role_t

Roles than a node can fulfill.

4.11.3 Member Data Documentation

4.11.3.1 std::string clk::Node::name

An unique node identifier, i.e., the hostname.

4.11.3.2 std::vector<role_t> clk::Node::roles

A list of roles that the node fulfills.

4.11.3.3 `std::string clk::Node::subcluster`

A subcluster to which the node belongs.

4.12 clk::Layer::ProviderConfigParam Struct Reference

Data collect configuration parameter.

```
#include <clk.h>
```

Public Attributes

- std::string [provider](#)
- std::vector< [ProviderOptionset](#) > [provider_optionsets](#)

4.12.1 Detailed Description

Data collect configuration parameter.

4.12.2 Member Data Documentation

4.12.2.1 std::string clk::Layer::ProviderConfigParam::provider

Provider name for data collection

4.12.2.2 std::vector<ProviderOptionset> clk::Layer::ProviderConfigParam::provider_ optionsets

Provider optionsets for provider

4.13 clk::Layer::ProviderOptionset Struct Reference

Data collect configuration for option set.

```
#include <clk.h>
```

Public Attributes

- `std::string` [id](#)
- `std::map< std::string, std::string >` [provider_option_vars](#)

4.13.1 Detailed Description

Data collect configuration for option set.

4.13.2 Member Data Documentation

4.13.2.1 `std::string` `clk::Layer::ProviderOptionset::id`

Provider Optionset

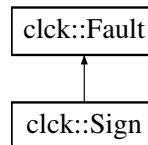
4.13.2.2 `std::map<std::string, std::string>` `clk::Layer::ProviderOptionset::provider_option_vars`

Provider Key=Value Parameters

4.14 clk::Sign Class Reference

A sign is an observation of an issue. [Sign](#) is derived from the [Fault](#) class.

`#include <clk.h>`Inheritance diagram for `clk::Sign`:



Public Types

- enum { **DIAGNOSED**, **OBSERVED** }

Public Attributes

- enum `clk::Sign::` { ... } [state](#)

4.14.1 Detailed Description

A sign is an observation of an issue. [Sign](#) is derived from the [Fault](#) class.

4.14.2 Member Data Documentation

4.14.2.1 enum { ... } `clk::Sign::state`

Either diagnosed (meaning used to make a diagnosis) or observed (undiagnosed).

4.15 clk::Layer::Sorting Struct Reference

Sort order for the list of faults returned by [get_faults\(\)](#).

```
#include <clk.h>
```

Public Types

- enum [FIELD](#) { CONFIDENCE, ID, NODE, SEVERITY }

Public Member Functions

- [Sorting](#) (bool [ascending](#), decltype([field](#)) [field](#))

Public Attributes

- bool [ascending](#) = true
- enum [clk::Layer::Sorting::FIELD](#) [field](#)

4.15.1 Detailed Description

Sort order for the list of faults returned by [get_faults\(\)](#).

4.15.2 Member Enumeration Documentation

4.15.2.1 enum clk::Layer::Sorting::FIELD

[Fault](#) Fields to sort on

4.15.3 Constructor & Destructor Documentation

4.15.3.1 clk::Layer::Sorting::Sorting (bool *ascending*, decltype(*field*) *field*)

[Sorting](#) constructor

4.15.4 Member Data Documentation

4.15.4.1 bool clk::Layer::Sorting::ascending = true

If true, sort in ascending order, otherwise sort in descending order.

4.15.4.2 enum clk::Layer::Sorting::FIELD clk::Layer::Sorting::field

Fault Fields to sort on The field attribute specifies the field to sort on

4.16 clk::Layer::Suppression Struct Reference

Suppress faults matching the specified values.

```
#include <clk.h>
```

Public Attributes

- int [confidence](#) = 0
- std::string [id](#)
- std::string [node](#)
- int [severity](#) = 0

4.16.1 Detailed Description

Suppress faults matching the specified values.

4.16.2 Member Data Documentation

4.16.2.1 int clk::Layer::Suppression::confidence = 0

Suppress all faults with a value less than the confidence value.

4.16.2.2 std::string clk::Layer::Suppression::id

Suppress all messages with a matching id. If empty, does not suppress on id.

4.16.2.3 std::string clk::Layer::Suppression::node

Suppress all messages containing the node value. If empty, does not suppress on node.

4.16.2.4 int clk::Layer::Suppression::severity = 0

Suppress all faults with a value less than the severity value.

Index

- ~Layer
 - clk::Layer, [25](#)
- analyze
 - clk::Layer, [25](#)
- ascending
 - clk::Layer::Sorting, [34](#)
- bin_path
 - clk::Layer::Config, [11](#)
- clk::Diagnosis, [15](#)
- clk::Fault, [16](#)
 - confidence, [16](#)
 - id, [16](#)
 - msg, [16](#)
 - nodes, [17](#)
 - remedy, [17](#)
 - rowid, [17](#)
 - severity, [17](#)
 - suppressed, [17](#)
- clk::Layer, [23](#)
 - ~Layer, [25](#)
 - analyze, [25](#)
 - collect, [25](#)
 - get_faults, [25](#)
 - get_messages, [26](#)
 - get_nodes, [26](#)
 - get_version_number, [26](#)
 - Layer, [25](#)
 - message, [27](#)
 - progress, [26](#)
 - SCALE_CONSTANT, [24](#)
 - SCALE_LINEAR, [24](#)
 - SCALE_LOGARITHMIC, [25](#)
 - SCALE_SQUARED, [25](#)
 - ScaleType, [24](#)
- clk::Layer::CollectConfig, [7](#)
 - cluster_permutations, [8](#)
 - database_close_delay, [8](#)
 - ignore_subclusters, [8](#)
 - network_interface, [8](#)
 - provider_async_output_dir, [8](#)
 - provider_aux_path, [8](#)
 - provider_config_params, [8](#)
 - provider_path, [8](#)
 - run_all_providers, [8](#)
 - run_async, [8](#)
 - scale, [9](#)
 - shared_dir, [9](#)
 - timeout, [9](#)
- clk::Layer::Config, [10](#)
 - bin_path, [11](#)
 - collect_config, [11](#)
 - Config, [10](#)
 - config_params, [11](#)
 - db_configs, [11](#)
 - expiration, [11](#)
 - fwd_file_list, [11](#)
 - fwds, [11](#)
 - install_path, [11](#)
 - language, [12](#)
 - node_source, [12](#)
 - nodes, [12](#)
 - now, [12](#)
 - provider_path, [12](#)
 - providers, [12](#)
- clk::Layer::ConfigParam, [13](#)
 - key, [13](#)
 - module, [13](#)
 - provider, [13](#)
 - provider_optionset, [13](#)
 - values, [13](#)
- clk::Layer::DatabaseConfig, [14](#)

- name, 14
- source, 14
- type, 14
- clk::Layer::Filter, 18
 - confidence, 18
 - ids, 18
 - nodes, 18
 - severity, 18
 - state, 18
 - suppressed, 19
 - type, 19
- clk::Layer::Fwd, 20
 - extension_mods, 21
 - extension_path, 21
 - Fwd, 20
 - kb_mods, 21
 - kb_path, 21
 - name, 21
 - provider_aux_path, 21
 - provider_path, 21
 - providers, 21
- clk::Layer::Message, 28
 - level, 28
 - msg, 28
- clk::Layer::ProviderConfigParam, 31
 - provider, 31
 - provider_optionsets, 31
- clk::Layer::ProviderOptionset, 32
 - id, 32
 - provider_option_vars, 32
- clk::Layer::Sorting, 34
 - ascending, 34
 - FIELD, 34
 - field, 34
 - Sorting, 34
- clk::Layer::Suppression, 36
 - confidence, 36
 - id, 36
 - node, 36
 - severity, 36
- clk::Node, 29
 - name, 29
 - role_t, 29
 - roles, 29
 - subcluster, 29
- clk::Sign, 33
 - state, 33
- cluster_permutations
 - clk::Layer::CollectConfig, 8
- collect
 - clk::Layer, 25
- collect_config
 - clk::Layer::Config, 11
- confidence
 - clk::Fault, 16
 - clk::Layer::Filter, 18
 - clk::Layer::Suppression, 36
- Config
 - clk::Layer::Config, 10
- config_params
 - clk::Layer::Config, 11
- database_close_delay
 - clk::Layer::CollectConfig, 8
- db_configs
 - clk::Layer::Config, 11
- expiration
 - clk::Layer::Config, 11
- extension_mods
 - clk::Layer::Fwd, 21
- extension_path
 - clk::Layer::Fwd, 21
- FIELD
 - clk::Layer::Sorting, 34
- field
 - clk::Layer::Sorting, 34
- Fwd
 - clk::Layer::Fwd, 20
- fwd_file_list
 - clk::Layer::Config, 11
- fwds
 - clk::Layer::Config, 11
- get_faults
 - clk::Layer, 25
- get_messages
 - clk::Layer, 26
- get_nodes
 - clk::Layer, 26
- get_version_number

- clk::Layer, 26
- id
 - clk::Fault, 16
 - clk::Layer::ProviderOptionset, 32
 - clk::Layer::Suppression, 36
- ids
 - clk::Layer::Filter, 18
- ignore_subclusters
 - clk::Layer::CollectConfig, 8
- install_path
 - clk::Layer::Config, 11
- kb_mods
 - clk::Layer::Fwd, 21
- kb_path
 - clk::Layer::Fwd, 21
- key
 - clk::Layer::ConfigParam, 13
- language
 - clk::Layer::Config, 12
- Layer
 - clk::Layer, 25
- level
 - clk::Layer::Message, 28
- message
 - clk::Layer, 27
- module
 - clk::Layer::ConfigParam, 13
- msg
 - clk::Fault, 16
 - clk::Layer::Message, 28
- name
 - clk::Layer::DatabaseConfig, 14
 - clk::Layer::Fwd, 21
 - clk::Node, 29
- network_interface
 - clk::Layer::CollectConfig, 8
- node
 - clk::Layer::Suppression, 36
- node_source
 - clk::Layer::Config, 12
- nodes
 - clk::Fault, 17
- clk::Layer::Config, 12
 - clk::Layer::Filter, 18
- now
 - clk::Layer::Config, 12
- progress
 - clk::Layer, 26
- provider
 - clk::Layer::ConfigParam, 13
 - clk::Layer::ProviderConfigParam, 31
- provider_async_output_dir
 - clk::Layer::CollectConfig, 8
- provider_aux_path
 - clk::Layer::CollectConfig, 8
 - clk::Layer::Fwd, 21
- provider_config_params
 - clk::Layer::CollectConfig, 8
- provider_option_vars
 - clk::Layer::ProviderOptionset, 32
- provider_optionset
 - clk::Layer::ConfigParam, 13
- provider_optionsets
 - clk::Layer::ProviderConfigParam, 31
- provider_path
 - clk::Layer::CollectConfig, 8
 - clk::Layer::Config, 12
 - clk::Layer::Fwd, 21
- providers
 - clk::Layer::Config, 12
 - clk::Layer::Fwd, 21
- remedy
 - clk::Fault, 17
- role_t
 - clk::Node, 29
- roles
 - clk::Node, 29
- rowid
 - clk::Fault, 17
- run_all_providers
 - clk::Layer::CollectConfig, 8
- run_async
 - clk::Layer::CollectConfig, 8
- scale

- clk::Layer::CollectConfig, [9](#)
- SCALE_CONSTANT
 - clk::Layer, [24](#)
- SCALE_LINEAR
 - clk::Layer, [24](#)
- SCALE_LOGARITHMIC
 - clk::Layer, [25](#)
- SCALE_SQUARED
 - clk::Layer, [25](#)
- ScaleType
 - clk::Layer, [24](#)
- severity
 - clk::Fault, [17](#)
 - clk::Layer::Filter, [18](#)
 - clk::Layer::Suppression, [36](#)
- shared_dir
 - clk::Layer::CollectConfig, [9](#)
- Sorting
 - clk::Layer::Sorting, [34](#)
- source
 - clk::Layer::DatabaseConfig, [14](#)
- state
 - clk::Layer::Filter, [18](#)
 - clk::Sign, [33](#)
- subcluster
 - clk::Node, [29](#)
- suppressed
 - clk::Fault, [17](#)
 - clk::Layer::Filter, [19](#)
- timeout
 - clk::Layer::CollectConfig, [9](#)
- type
 - clk::Layer::DatabaseConfig, [14](#)
 - clk::Layer::Filter, [19](#)
- values
 - clk::Layer::ConfigParam, [13](#)