RTI Cloud Discovery Service

Version 7.7.0

1 RTI Cloud Discovery Service **1**

1.1 Feedback and Support for this Release. . . . 1

2 Module Index **3**

2.1 Modules . . . 3

3 Data Structure Index **5**

3.1 Data Structures . . . 5

4 File Index **7**

4.1 File List . . . 7

5 Module Documentation **9**

5.1 Logging . . . 9

5.1.1 Detailed Description . . . 10

5.1.2 Macro Definition Documentation . . . 10

5.1.2.1 CDS_SUBMODULE_MASK_COMMON . . . 10

5.1.2.2 CDS_SUBMODULE_MASK_CONFIG . . . 10

5.1.2.3 CDS_SUBMODULE_MASK_ADAPTERS . . . 10

5.1.2.4 CDS_SUBMODULE_MASK_PROCESSORS . . . 11

5.1.2.5 CDS_SUBMODULE_MASK_MONITORING . . . 11

5.1.2.6 CDS_SUBMODULE_MASK_SERVICE . . . 11

5.1.2.7 CDS_SUBMODULE_MASK_SERVICEAPP . . . 11

5.1.2.8 CDS_SUBMODULE_MASK_SERVICELIB . . . 11

5.1.2.9 CDS_SUBMODULE_MASK_ALL . . . 11

5.1.3 Function Documentation . . . 11

5.1.3.1 RTI_CDS_Logger_log() . . . 12

5.1.4 Variable Documentation . . . 12

5.1.4.1 RTI_CDS_LOG_VERBOSITY_DEBUG . . . 12

5.1.4.2 RTI_CDS_LOG_VERBOSITY_ALL . . . 12

5.1.4.3 RTI_CDS_LOG_VERBOSITY_INFO . . . 13

5.1.4.4 RTI_CDS_LOG_VERBOSITY_WARNINGS . . . 13

5.1.4.5 RTI_CDS_LOG_VERBOSITY_EXCEPTIONS . . . 13

5.1.4.6 RTI_CDS_LOG_VERBOSITY_SILENT . . . 13

5.2 Service Property . . . 13

5.2.1 Detailed Description . . . 14

5.2.2 Macro Definition Documentation . . . 14

5.2.2.1 RTI_CDS_DomainListProperty_INITIALIZER . . . 14

5.2.2.2 RTI_CDS_TransportSelectionProperty_INITIALIZER . . . 14

5.2.3 Function Documentation . . . 15

5.2.3.1 RTI_CDS_Property_initialize() . . . 15

5.2.3.2 RTI_CDS_Property_copy() . . . 15
5.2.3.3 RTI_CDS_Property_finalize() . . . 15
5.3 Library API . . . 16
5.3.1 Detailed Description . . . 17
5.3.2 Function Documentation . . . 17
5.3.2.1 RTI_CDS_Service_new() . . . 17
5.3.2.2 RTI_CDS_Service_delete() . . . 18
5.3.2.3 RTI_CDS_Service_start() . . . 18
5.3.2.4 RTI_CDS_Service_stop() . . . 19
5.3.2.5 RTI_CDS_Service_finalize_globals() . . . 19
5.3.2.6 RTI_CDS_Service_is_started() . . . 19
5.3.2.7 RTI_CDS_Service_update_rtps_psk_secret_passphrase() . . . 20

6 Data Structure Documentation **21**

6.1 RTI_CDS_ForwarderEventProperty Struct Reference . . . 21
6.1.1 Detailed Description . . . 21
6.2 RTI_CDS_Property Struct Reference . . . 21
6.2.1 Detailed Description . . . 22
6.2.2 Field Documentation . . . 22
6.2.2.1 cfg_file . . . 22
6.2.2.2 cfg_strings . . . 23
6.2.2.3 cfg_strings_count . . . 23
6.2.2.4 service_name . . . 23
6.2.2.5 application_name . . . 24
6.2.2.6 enforce_xsd_validation . . . 24
6.2.2.7 skip_default_files . . . 24
6.2.2.8 service_verbosity . . . 24
6.2.2.9 dds_verbosity . . . 25
6.2.2.10 transport_selection . . . 25
6.2.2.11 domain_list . . . 25
6.2.2.12 enable_administration . . . 25
6.2.2.13 administration_domain_id . . . 26
6.2.2.14 enable_monitoring . . . 26
6.2.2.15 monitoring_domain_id . . . 26
6.2.2.16 license_file_name . . . 26
6.2.2.17 user_environment . . . 26
6.3 RTI_CDS_Service Struct Reference . . . 27
6.3.1 Detailed Description . . . 27

7 File Documentation **29**

7.1 clouddiscoveryservice_builtin.h 29
7.2 clouddiscoveryservice_dll.h 30
7.3 clouddiscoveryservice_infrastructure.h File Reference 31
7.3.1 Detailed Description 31
7.4 clouddiscoveryservice_infrastructure.h 32
7.5 clouddiscoveryservice_log.h File Reference 34
7.5.1 Detailed Description 34
7.6 clouddiscoveryservice_log.h 34
7.7 clouddiscoveryservice_service.h File Reference 36
7.7.1 Detailed Description 37
7.8 clouddiscoveryservice_service.h 37

Index **45**

Chapter 1

RTI Cloud Discovery Service

RTI Cloud Discovery Service is a stand-alone application that bootstraps discovery for RTI Connext DDS applications in dynamic environments where UDP/IP multicast is not available. This is typical of wide-area networks or some cloud-based environments where the routers and switches may disable IP multicast forwarding. RTI Cloud Discovery Service also works in conjunction with the RTI Real-Time WAN Transport to provide peer-to-peer communication between DomainParticipants situated behind Network Address Translators (NATs).

You can learn how to use RTI Cloud Discovery Service through the `RTI Cloud Discovery Service User's Manual`.

RTI Cloud Discovery Service can be run as a standalone application, or linked into your application.

A reference to the Service (Library) API can be found here:

- **Library API** (p. 16)

1.1 Feedback and Support for this Release.

For more information, visit our knowledge base, accessible from `https://community.rti.com/` to see sample code, general information on RTI Connext, performance information, troubleshooting tips, and technical details.

By its very nature, the knowledge base is continuously evolving and improving. We hope that you will find it helpful. If there are questions that you would like to see addressed or comments you would like to share, please send email to `support@rti.com`. We can only guarantee a response for customers with a current maintenance contract or subscription. To purchase a maintenance contract or subscription, contact your local RTI representative (see `http://www.rti.com/company/contact.html`), send an email request to `sales@rti.com`, or call +1 (408) 990-7400.

Please do not hesitate to contact RTI with questions or comments about this release. We welcome any input on how to improve RTI Cloud Discovery Service and RTI Connext to suit your needs.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Logging 9
Library API 16
    Service Property 13

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

RTI_CDS_ForwarderEventProperty
Data structure to configure the forwarder resend event mechanism. It allows CDS to resend a new or update announcement so that discovery can occur quickly in a lossy WAN environment 21
RTI_CDS_Property
Configuration property for RTI Cloud Discovery Service 21
RTI_CDS_Service
RTI Cloud Discovery Service 27

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

clouddiscoveryservice_builtin.h 29
clouddiscoveryservice_dll.h 30
clouddiscoveryservice_infrastructure.h
    Service infrastructure 31
clouddiscoveryservice_log.h
    Logging facilities for RTI Cloud Discovery Service 34
clouddiscoveryservice_service.h
    Implements the RTI Cloud Discovery Service 36

Chapter 5

Module Documentation

5.1 Logging

Provides control over the logging facilities for RTI Cloud Discovery Service.

Macros

- #define **CDS_SUBMODULE_MASK_COMMON** (0x0002)
- #define **CDS_SUBMODULE_MASK_CONFIG** (0x0004)
- #define **CDS_SUBMODULE_MASK_ADAPTERS** (0x0008)
- #define **CDS_SUBMODULE_MASK_PROCESSORS** (0x0010)
- #define **CDS_SUBMODULE_MASK_MONITORING** (0x0012)
- #define **CDS_SUBMODULE_MASK_SERVICE** (0x0020)
- #define **CDS_SUBMODULE_MASK_SERVICEAPP** (0x0040)
- #define **CDS_SUBMODULE_MASK_SERVICELIB** (0x0060)
- #define **CDS_SUBMODULE_MASK_ALL** RTI_LOG_SUBMODULE_MASK_ALL

Functions

- void **RTI_CDS_Logger_log** (NDDS_Config_LogLevel log_level, const char *format,...)

  Logs a message with the specified level within the running instance of RTI Cloud Discovery Service. The message will be printed under the log of the currently executing RTI Cloud Discovery Service, if the instance was started with `log_level` *or higher verbosity.*

Variables

- const int **RTI_CDS_LOG_VERBOSITY_DEBUG**

  Log verbosity level: exceptions + warnings + info + periodic + content.
- const int **RTI_CDS_LOG_VERBOSITY_ALL**

  Log verbosity level: exceptions + warnings + info + periodic.
- const int **RTI_CDS_LOG_VERBOSITY_INFO**

  Log verbosity level: exceptions + warnings + info.
- const int **RTI_CDS_LOG_VERBOSITY_WARNINGS**

  Log verbosity level: exceptions + warnings.
- const int **RTI_CDS_LOG_VERBOSITY_EXCEPTIONS**

  Log verbosity level: exceptions.
- const int **RTI_CDS_LOG_VERBOSITY_SILENT**

  Log verbosity level: silent.

5.1.1 Detailed Description

Provides control over the logging facilities for RTI Cloud Discovery Service.

5.1.2 Macro Definition Documentation

5.1.2.1 CDS_SUBMODULE_MASK_COMMON

```
#define CDS_SUBMODULE_MASK_COMMON (0x0002)
```

Mask for turning on logging for only the common submodule

5.1.2.2 CDS_SUBMODULE_MASK_CONFIG

```
#define CDS_SUBMODULE_MASK_CONFIG (0x0004)
```

Mask for turning on logging for only the config submodule

5.1.2.3 CDS_SUBMODULE_MASK_ADAPTERS

```
#define CDS_SUBMODULE_MASK_ADAPTERS (0x0008)
```

Mask for turning on logging for only the adapter submodule

5.1.2.4 CDS_SUBMODULE_MASK_PROCESSORS

```
#define CDS_SUBMODULE_MASK_PROCESSORS (0x0010)
```

Mask for turning on logging for only the processor submodule

5.1.2.5 CDS_SUBMODULE_MASK_MONITORING

```
#define CDS_SUBMODULE_MASK_MONITORING (0x0012)
```

Mask for turning on logging for only the monitoring submodule

5.1.2.6 CDS_SUBMODULE_MASK_SERVICE

```
#define CDS_SUBMODULE_MASK_SERVICE (0x0020)
```

Mask for turning on logging for only the service submodule

5.1.2.7 CDS_SUBMODULE_MASK_SERVICEAPP

```
#define CDS_SUBMODULE_MASK_SERVICEAPP (0x0040)
```

Mask for turning on logging for only the service app submodule

5.1.2.8 CDS_SUBMODULE_MASK_SERVICELIB

```
#define CDS_SUBMODULE_MASK_SERVICELIB (0x0060)
```

Mask for turning on logging for only the service lib submodule

5.1.2.9 CDS_SUBMODULE_MASK_ALL

```
#define CDS_SUBMODULE_MASK_ALL RTI_LOG_SUBMODULE_MASK_ALL
```

Mask for turning on logging for all submodules

5.1.3 Function Documentation

5.1.3.1 RTI_CDS_Logger_log()

```
void RTI_CDS_Logger_log (
            NDDS_Config_LogLevel log_level,
            const char * format,
             ... )
```

Logs a message with the specified level within the running instance of RTI Cloud Discovery Service. The message will be printed under the log of the currently executing RTI Cloud Discovery Service, if the instance was started with `log_level` or higher verbosity.

The message is specified with a format and a format parameter, in a similar fashion to the standard C printf() operation.

The generated log will be part of logging stream of the running RTI Cloud Discovery Service, if the `log_level` is part of the configured verbosity. The result log message may include additional information according to the Connext logging configuration, such as Advlog Context, thread ID, line number, etc. Additionally, the result log message will contain a newline character at the end, so the format does not need to contain it.

Parameters

in	*log_level*	Log level associated to the message
in	*format*	message format specification
in	*...*	variablelength argument (stdarg)

5.1.4 Variable Documentation

5.1.4.1 RTI_CDS_LOG_VERBOSITY_DEBUG

```
const int RTI_CDS_LOG_VERBOSITY_DEBUG  [extern]
```

Log verbosity level: exceptions + warnings + info + periodic + content.

5.1.4.2 RTI_CDS_LOG_VERBOSITY_ALL

```
const int RTI_CDS_LOG_VERBOSITY_ALL  [extern]
```

Log verbosity level: exceptions + warnings + info + periodic.

5.1.4.3 RTI_CDS_LOG_VERBOSITY_INFO

```
const int RTI_CDS_LOG_VERBOSITY_INFO  [extern]
```

Log verbosity level: exceptions + warnings + info.

5.1.4.4 RTI_CDS_LOG_VERBOSITY_WARNINGS

```
const int RTI_CDS_LOG_VERBOSITY_WARNINGS  [extern]
```

Log verbosity level: exceptions + warnings.

5.1.4.5 RTI_CDS_LOG_VERBOSITY_EXCEPTIONS

```
const int RTI_CDS_LOG_VERBOSITY_EXCEPTIONS  [extern]
```

Log verbosity level: exceptions.

5.1.4.6 RTI_CDS_LOG_VERBOSITY_SILENT

```
const int RTI_CDS_LOG_VERBOSITY_SILENT  [extern]
```

Log verbosity level: silent.

5.2 Service Property

Service Property for RTI Cloud Discovery Service. Use this in conjunction with **Library API** (p. 16), to configure RTI Cloud Discovery Service in your code.

Data Structures

- struct **RTI_CDS_Property**

  Configuration property for RTI Cloud Discovery Service.

Macros

- #define **RTI_CDS_DomainListProperty_INITIALIZER**

  Allows all domain IDs for RTI Cloud Discovery Service.
- #define **RTI_CDS_TransportSelectionProperty_INITIALIZER**

  Empty transport selection list. Use value from XML configuration.

Functions

- DDS_Boolean **RTI_CDS_Property_initialize** (struct **RTI_CDS_Property** ∗self)

  *Initialize an instance of the **Service Property** (p. 13) variable with empty values.*
- struct **RTI_CDS_Property** ∗ **RTI_CDS_Property_copy** (struct **RTI_CDS_Property** ∗to, const struct **RTI_↩ CDS_Property** ∗from)

  *Copy the values of a **Service Property** (p. 13) variable into another.*
- void **RTI_CDS_Property_finalize** (struct **RTI_CDS_Property** ∗self)

  *Finalize an instance of the **Service Property** (p. 13) variable.*

5.2.1 Detailed Description

Service Property for RTI Cloud Discovery Service. Use this in conjunction with **Library API** (p. 16), to configure RTI Cloud Discovery Service in your code.

5.2.2 Macro Definition Documentation

5.2.2.1 RTI_CDS_DomainListProperty_INITIALIZER

```
#define RTI_CDS_DomainListProperty_INITIALIZER
```

Value:

```
{ \
    NULL, NULL \
}
```

Allows all domain IDs for RTI Cloud Discovery Service.

5.2.2.2 RTI_CDS_TransportSelectionProperty_INITIALIZER

```
#define RTI_CDS_TransportSelectionProperty_INITIALIZER
```

Value:

```
{ \
    { RTI_CDS_TransportUnicastProperty_INITIALIZER }, 0 \
}
```

Empty transport selection list. Use value from XML configuration.

5.2.3 Function Documentation

5.2.3.1 RTI_CDS_Property_initialize()

```
DDS_Boolean RTI_CDS_Property_initialize (
            struct  RTI_CDS_Property * self )
```

Initialize an instance of the **Service Property** (p. 13) variable with empty values.

e

Parameters

self	A **Service Property** (p. 13) instance

5.2.3.2 RTI_CDS_Property_copy()

```
struct  RTI_CDS_Property * RTI_CDS_Property_copy (
            struct  RTI_CDS_Property * to,
            const struct  RTI_CDS_Property * from )
```

Copy the values of a **Service Property** (p. 13) variable into another.

e

Parameters

to	A **Service Property** (p. 13) instance to copy the values to
from	A **Service Property** (p. 13) instance to copy the values from

Returns

The pointer to **Service Property** (p. 13) containing the copied values. NULL otherwise.

5.2.3.3 RTI_CDS_Property_finalize()

```
void RTI_CDS_Property_finalize (
            struct  RTI_CDS_Property * self )
```

Finalize an instance of the **Service Property** (p. 13) variable.

e

Parameters

self	A **Service Property** (p. 13) instance

5.3 Library API

RTI Cloud Discovery Service can be deployed as a C library linked into your application on select architectures.

Modules

- **Service Property**

  *Service Property for RTI Cloud Discovery Service. Use this in conjunction with **Library API** (p. 16), to configure RTI Cloud Discovery Service in your code.*

Data Structures

- struct **RTI_CDS_Service**

  RTI Cloud Discovery Service.

Functions

- struct **RTI_CDS_Service** * **RTI_CDS_Service_new** (const struct **RTI_CDS_Property** *property)

  Create a new RTI Cloud Discovery Service instance.
- void **RTI_CDS_Service_delete** (struct **RTI_CDS_Service** *self)

  Stop and delete an RTI Cloud Discovery Service instance.
- DDS_Boolean **RTI_CDS_Service_start** (struct **RTI_CDS_Service** *self)

  Start RTI Cloud Discovery Service.
- DDS_Boolean **RTI_CDS_Service_stop** (struct **RTI_CDS_Service** *self)

  Stop RTI Cloud Discovery Service.
- DDS_Boolean **RTI_CDS_Service_finalize_globals** (void)
- DDS_Boolean **RTI_CDS_Service_is_started** (struct **RTI_CDS_Service** *self)

  Query whether this RTI Cloud Discovery Service is currently started.
- DDS_Boolean **RTI_CDS_Service_update_rtps_psk_secret_passphrase** (struct **RTI_CDS_Service** *self, const char *rtps_psk_secret_passphrase)

  *Sets the dds.sec.crypto.rtps_psk_secret_passphrase property for the RTI Cloud Discovery Service instance started with **RTI_CDS_Service_start** (p. 18).*

5.3.1 Detailed Description

RTI Cloud Discovery Service can be deployed as a C library linked into your application on select architectures.

This API allows you to create, configure and start RTI Cloud Discovery Service instances from your application.

The following code shows the typical use of the API:

```
struct RTI_CDS_Property property = RTI_CDS_Property_INITIALIZER;
struct RTI_CDS_Service *service = NULL;
property.cfg_file = "my_cloud_discovery_service_cfg.xml";
property.service_name = "my_cloud_discovery_service";
...
service = RTI_CDS_Service_new(&property);
if (service == NULL) {
   printf("Error ...");
   return -1;
}
if (!RTI_CDS_Service_start(service)) {
   printf("Error ...");
   RTI_CDS_Service_delete(service);
   return -1;
}
while (keep_running) {
    sleep();
    ...
}
RTI_CDS_Service_delete(service);
return 0;
```

Instead of a file, you can use XML strings to configure RTI Cloud Discovery Service. See **Service Property** (p. 13) for more information.

To build your application you need to link with the RTI Cloud Discovery Service library in <**RTI Connext home**>/**bin**/<**architecture**>/

An example is provided at **https://github.com/rticommunity/rticonnextdds-examples/tree/master/exampl _discovery_service/library_api**.

5.3.1.0.1 Development Requirements | | Linux Systems | Windows Systems | | ----------:|:----------:|:----------:| | Link Libraries (Compile Time) | librticlouddiscoveryservice.so | rticlouddiscoveryservice.dll | | Shared Libraries (Run Time) | librtiroutingservice.so | rtiroutingservice.dll | | ^ | librtidlc.so | rtidlc.dll | | ^ | librticonnextmsgc.so | rticonnextmsgc.dll | | ^ | libnddsmetp.so | nddsmetp.dll | | ^ | librtiapputilsc.so | rtiapputilsc.dll | | ^ | librtixml2.so | rtixml2.dll | | ^ | libnddsc.so | nddsc.dll | | ^ | libnddscore.so | nddscore.dll | | Static Libraries | librticlouddiscoveryservicez.a | rticlouddiscoveryservicez.lib | | ^ | librtiroutingservicez.a | rtiroutingservicez.lib | | ^ | librtidlcz.a | rtidlcz.lib | | ^ | librticonnextmsgcz.a | rticonnextmsgcz.lib | | ^ | libnddsmetpz.a | nddsmetpz.lib | | ^ | librtiapputilscz.a | rtiapputilscz.lib | | ^ | librtixml2z.a | rtixml2z.lib | | ^ | libnddscz.a | nddscz.lib | | ^ | libnddscorez.a | nddscorez.lib | | Headers | **clouddiscoveryservice/clouddiscoveryservice_service.h** (p. 36) | |

NOTE: If you are using debug libraries, remember to add the 'd' suffix to the library name. For shared libraries on macOS systems, the library name is the same as that for Linux except the extension is '.dylib' instead of '.so'.

5.3.2 Function Documentation

5.3.2.1 RTI_CDS_Service_new()

```
struct  RTI_CDS_Service * RTI_CDS_Service_new (
             const struct  RTI_CDS_Property * property )
```

Create a new RTI Cloud Discovery Service instance.

e

Parameters

property	The properties to configure RTI Cloud Discovery Service This parameter is copied internally, so the user is responsible for releasing any memory allocated inside this structure.

MT Safety:

On non-Linux, non-Windows systems (i.e. VxWorks): UNSAFE for multiple threads to simultaneously make the FIRST call to **RTI_CDS_Service_new()** (p. 17). Subsequent calls are thread safe. On Windows and Linux systems, these calls are thread-safe.

5.3.2.2 RTI_CDS_Service_delete()

```
void RTI_CDS_Service_delete (
            struct  RTI_CDS_Service * self )
```

Stop and delete an RTI Cloud Discovery Service instance.

e

See also

RTI_CDS_Service_stop (p. 19)

Parameters

self	An **RTI_CDS_Service** (p. 27) instance created with **RTI_CDS_Service_new** (p. 17)

5.3.2.3 RTI_CDS_Service_start()

```
DDS_Boolean RTI_CDS_Service_start (
            struct  RTI_CDS_Service * self )
```

Start RTI Cloud Discovery Service.

e

This is a non-blocking operation. RTI Cloud Discovery Service will create its own set of threads to perform its tasks.

Parameters

self	An **RTI_CDS_Service** (p. 27) instance created with **RTI_CDS_Service_new** (p. 17)

Returns

True on success; False otherwise

5.3.2.4 RTI_CDS_Service_stop()

```
DDS_Boolean RTI_CDS_Service_stop (
            struct  RTI_CDS_Service * self )
```

Stop RTI Cloud Discovery Service.

e

This function won't return the execution control until the instance is fully stopped.

Parameters

self	An **RTI_CDS_Service** (p. 27) instance created with **RTI_CDS_Service_new** (p. 17)

Returns

True on success; False otherwise

5.3.2.5 RTI_CDS_Service_finalize_globals()

```
DDS_Boolean RTI_CDS_Service_finalize_globals (
            void  )
```

e

[DEPRECATED]

Calling **RTI_CDS_Service_finalize_globals()** (p. 19) is no longer necessary and the function will be removed in future versions.

5.3.2.6 RTI_CDS_Service_is_started()

```
DDS_Boolean RTI_CDS_Service_is_started (
            struct  RTI_CDS_Service * self )
```

Query whether this RTI Cloud Discovery Service is currently started.

e

5.3.2.7 RTI_CDS_Service_update_rtps_psk_secret_passphrase()

```
DDS_Boolean RTI_CDS_Service_update_rtps_psk_secret_passphrase (
            struct  RTI_CDS_Service * self,
            const char * rtps_psk_secret_passphrase )
```

Sets the dds.sec.crypto.rtps_psk_secret_passphrase property for the RTI Cloud Discovery Service instance started with **RTI_CDS_Service_start** (p. 18).

e

This operation performs an update; if there was no value for the dds.sec.crypto.rtps_psk_secret_passphrase in the XML configuration when RTI Cloud Discovery Service was started by calling **RTI_CDS_Service_start** (p. 18), the operation would fail.

Parameters

self	An **RTI_CDS_Service** (p. 27) instance started with **RTI_CDS_Service_start** (p. 18)
rtps_psk_secret_passphrase	The new value for the dds.sec.crypto.rtps_psk_secret_passphrase property

Chapter 6

Data Structure Documentation

6.1 RTI_CDS_ForwarderEventProperty Struct Reference

Data structure to configure the forwarder resend event mechanism. It allows CDS to resend a new or update announcement so that discovery can occur quickly in a lossy WAN environment.

`#include <clouddiscoveryservice_infrastructure.h>`

6.1.1 Detailed Description

Data structure to configure the forwarder resend event mechanism. It allows CDS to resend a new or update announcement so that discovery can occur quickly in a lossy WAN environment.

6.2 RTI_CDS_Property Struct Reference

Configuration property for RTI Cloud Discovery Service.

`#include <clouddiscoveryservice_service.h>`

Data Fields

- char * **cfg_file**

  Path to an RTI Cloud Discovery Service configuration file.
- const char ** **cfg_strings**

  XML configuration represented as strings.
- unsigned int **cfg_strings_count**

  *Size of the array **cfg_strings** (p. 22).*
- char * **service_name**

  The name of the RTI Cloud Discovery Service instance to run.

- char * **application_name**

  Assigns a name to the execution of the RTI Cloud Discovery Service.
- DDS_Boolean **enforce_xsd_validation**

  Controls whether the service applies XSD validation to the loaded configuration.
- DDS_Boolean **skip_default_files**

  Set it to true to avoid loading the standard files usually loaded by RTI Cloud Discovery Service.
- int **service_verbosity**

  The verbosity of the service.
- int **dds_verbosity**

  The verbosity of RTI Connext core libraries.
- struct RTI_CDS_TransportSelectionProperty **transport_selection**

  Specifies the transport and receive ports used to receive and send discovery traffic. Specifically, a receive resource is uniquely identified by a transport class-receive port pair. Overrides XML configuration if specified.
- struct RTI_CDS_DomainListProperty **domain_list**

  Specifies the set of domain IDs the service will accept announcements from.
- DDS_Boolean **enable_administration**

  Set this to true to enable remote administration or false to disable it.
- int **administration_domain_id**

  *If **enable_administration** (p. 25) is true, this is the domain ID to use for remote administration.*
- DDS_Boolean **enable_monitoring**

  Set this to true to enable monitoring or false to disable it.
- int **monitoring_domain_id**

  *If **enable_monitoring** (p. 26) is true, this is the domain ID to use for publishing monitoring topics.*
- char * **license_file_name**

  Path to an RTI Cloud Discovery Service license file.
- struct RTI_RoutingServiceProperties **user_environment**

  Dictionary of user variables. The dictionary provides a parallel way to expand XML configuration variables in the form , when they are not defined in the environment.

6.2.1 Detailed Description

Configuration property for RTI Cloud Discovery Service.

6.2.2 Field Documentation

6.2.2.1 cfg_file

```
char* RTI_CDS_Property::cfg_file
```

Path to an RTI Cloud Discovery Service configuration file.

If not `NULL`, this file is loaded; otherwise, if `cfg_strings` is not `NULL`, that XML code is loaded

[default] NULL.

6.2.2.2 cfg_strings

```
const char** RTI_CDS_Property::cfg_strings
```

XML configuration represented as strings.

An array of strings that altogether make up an XML document to configure RTI Cloud Discovery Service. This parameter is used only if **cfg_file** (p. 22) is `NULL`.

For example:

```
int MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE = 3;
const char *
MY_CLOUD_DISCOVERY_SERVICE_CFG[MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE] =
     {"<dds>
            <cloud_discovery_service name=\"MyServiceName\">"
             ...
            </cloud_discovery_service>
      </dds>"};

property.cfg_strings = MY_CLOUD_DISCOVERY_SERVICE_CFG;
property.cfg_strings_count = MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE;
```

The reason for using an array instead of one single string is to get around the limited size of literal strings in C. In general, if you create the XML string dynamically using one single string in the array, setting **cfg_strings_count** (p. 23) to 1 is enough:

```
property.cfg_strings = malloc(sizeof(char *));
property.cfg_strings[0] = "<dds><cloud_discovery_service>...</dds>";
property.cfg_strings_count = 1;
```

If your target system doesn't support a file system, you can use XML strings to configure the service. To ease this process, a utility is shipped to generate a C string array from a text file. You can find this utility in [RTI Cloud Discovery Service installation directory]/resource/perl/cStringifyFile.pl

[default] NULL.

6.2.2.3 cfg_strings_count

```
unsigned int RTI_CDS_Property::cfg_strings_count
```

Size of the array **cfg_strings** (p. 22).

[default] 0.

6.2.2.4 service_name

```
char* RTI_CDS_Property::service_name
```

The name of the RTI Cloud Discovery Service instance to run.

This is the name used to find the

```
<cloud_discovery_service>
```

XML tag in the configuration file; this is the default name that will be used to refer to this execution in remote administration and monitoring if application_name isn't specified.

[default] NULL (use built-in configuration)

6.2.2.5 application_name

```
char* RTI_CDS_Property::application_name
```

Assigns a name to the execution of the RTI Cloud Discovery Service.

Remote commands and status information will refer to the RTI Cloud Discovery Service using this name.

[default] service_name if this value is NULL.

6.2.2.6 enforce_xsd_validation

```
DDS_Boolean RTI_CDS_Property::enforce_xsd_validation
```

Controls whether the service applies XSD validation to the loaded configuration.

[default] DDS_BOOLEAN_TRUE

6.2.2.7 skip_default_files

```
DDS_Boolean RTI_CDS_Property::skip_default_files
```

Set it to true to avoid loading the standard files usually loaded by RTI Cloud Discovery Service.

Only the configuration in **cfg_file** (p. 22) or **cfg_strings** (p. 22) will be loaded.

[default] DDS_BOOLEAN_FALSE

6.2.2.8 service_verbosity

```
int RTI_CDS_Property::service_verbosity
```

The verbosity of the service.

Values:

- **RTI_CDS_LOG_VERBOSITY_SILENT** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_EXCEPTIONS** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_WARNINGS** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_INFO** (p. 12)
- **RTI_CDS_LOG_VERBOSITY_ALL** (p. 12)
- **RTI_CDS_LOG_VERBOSITY_DEBUG** (p. 12)

[default] RTI_CDS_LOG_VERBOSITY_EXCEPTIONS

6.2.2.9 dds_verbosity

```
int RTI_CDS_Property::dds_verbosity
```

The verbosity of RTI Connext core libraries.

Values:

- **RTI_CDS_LOG_VERBOSITY_SILENT** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_EXCEPTIONS** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_WARNINGS** (p. 13)
- **RTI_CDS_LOG_VERBOSITY_INFO** (p. 12)
- **RTI_CDS_LOG_VERBOSITY_ALL** (p. 12)
- **RTI_CDS_LOG_VERBOSITY_DEBUG** (p. 12)

[default] RTI_CDS_LOG_VERBOSITY_EXCEPTIONS

6.2.2.10 transport_selection

```
struct RTI_CDS_TransportSelectionProperty RTI_CDS_Property::transport_selection
```

Specifies the transport and receive ports used to receive and send discovery traffic. Specifically, a receive resource is uniquely identified by a transport class-receive port pair. Overrides XML configuration if specified.

6.2.2.11 domain_list

```
struct RTI_CDS_DomainListProperty RTI_CDS_Property::domain_list
```

Specifies the set of domain IDs the service will accept announcements from.

[default] Unspecified. Use value from XML configuration.

6.2.2.12 enable_administration

```
DDS_Boolean RTI_CDS_Property::enable_administration
```

Set this to true to enable remote administration or false to disable it.

[default] DDS_BOOLEAN_FALSE

6.2.2.13 administration_domain_id

```
int RTI_CDS_Property::administration_domain_id
```

If **enable_administration** (p. 25) is true, this is the domain ID to use for remote administration.

Takes precedence over the XML configuration. If **enable_administration** (p. 25) is false, this value is not used even if remote administration is enabled in the XML configuration.

[default] 0

6.2.2.14 enable_monitoring

```
DDS_Boolean RTI_CDS_Property::enable_monitoring
```

Set this to true to enable monitoring or false to disable it.

[default] DDS_BOOLEAN_FALSE

6.2.2.15 monitoring_domain_id

```
int RTI_CDS_Property::monitoring_domain_id
```

If **enable_monitoring** (p. 26) is true, this is the domain ID to use for publishing monitoring topics.

Takes precedence over the XML configuration. If **enable_monitoring** (p. 26) is false, this value is not used even if monitoring is enabled in the XML configuration.

[default] 0

6.2.2.16 license_file_name

```
char* RTI_CDS_Property::license_file_name
```

Path to an RTI Cloud Discovery Service license file.

If not `NULL`, this file is checked for a valid license; otherwise, default location will be used. This parameter is only used if your installation requires a license file.

[default] NULL.

6.2.2.17 user_environment

```
struct RTI_RoutingServiceProperties RTI_CDS_Property::user_environment
```

Dictionary of user variables. The dictionary provides a parallel way to expand XML configuration variables in the form , when they are not defined in the environment.

[default] empty

6.3 RTI_CDS_Service Struct Reference

RTI Cloud Discovery Service.

`#include <clouddiscoveryservice_service.h>`

6.3.1 Detailed Description

RTI Cloud Discovery Service.

e

Chapter 7

File Documentation

7.1 clouddiscoveryservice_builtin.h

```
1 /*
2  * Copyright (c) 2016-2025 Real-Time Innovations, Inc. All rights reserved.
3  * No duplications, whole or partial, manual or electronic, may be made
4  * without express written permission.  Any such copies, or revisions thereof,
5  * must display this notice unaltered.
6  * This code contains trade secrets of Real-Time Innovations, Inc.
7  */
8
9 #ifndef clouddiscoveryservice_builtin_h
10 #define clouddiscoveryservice_builtin_h
11
12
13 #ifndef cds_dll_h
14   #include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
15   #include "clouddiscoveryservice/clouddiscoveryservice_service.h"
16 #endif
17
18 #ifndef dds_c_domain_h
19   #include "dds_c/dds_c_domain.h"
20 #endif
21
22 #ifndef disc_builtin_h
23   #include "disc/disc_builtin.h"
24 #endif
25
26 #ifdef __cplusplus
27 extern "C" {
28 #endif
29
30
31 typedef enum {
32     CDS_DISCOVERY_PROTOCOL_MODE_SPDP,
33     CDS_DISCOVERY_PROTOCOL_MODE_SPDP2,
34     CDS_DISCOVERY_PROTOCOL_MODE_INVALID
35 } CDSDiscoveryProtocolMode;
36
37
38 typedef struct CDSParticipantDiscoveryImpl CDSParticipantDiscovery;
39
40
41 typedef struct CDSParticipantDetectorImpl CDSParticipantDetector;
42
43
44 typedef struct CDSParticipantAnnouncerImpl CDSParticipantAnnouncer;
45
46 extern CDSDllExport int CDSParticipantDiscovery_get_unicast_port_for_domain(
47         DDS_DomainId_t domain_id,
48         const struct DDS_WireProtocolQosPolicy *wire_protocol);
49
50 extern CDSDllExport CDSParticipantDiscovery *CDSParticipantDiscovery_new(
51         DDS_DomainParticipantFactory *factory,
```

```
52         const struct DDS_DomainParticipantQos *participant_qos,
53         CDSDiscoveryProtocolMode protocolMode);
54
55 extern CDSDllExport void CDSParticipantDiscovery_delete(
56         CDSParticipantDiscovery *self);
57
58 extern CDSDllExport CDSParticipantDetector *
59 CDSParticipantDiscovery_assert_detector(
60         CDSParticipantDiscovery *self,
61         struct PRESPsReaderListener *listener);
62
63 extern CDSDllExport CDSParticipantAnnouncer *
64 CDSParticipantDiscovery_assert_announcer(
65         CDSParticipantDiscovery *participantDiscovery);
66
67 extern CDSDllExport void CDSParticipantDiscovery_delete_announcer(
68         CDSParticipantDiscovery *participantDiscovery);
69
70 extern CDSDllExport void CDSParticipantDiscovery_delete_detector(
71         CDSParticipantDiscovery *participantDiscovery);
72
77 extern CDSDllExport DDS_Boolean CDSParticipantDetector_take(
78         CDSParticipantDetector *self,
79         struct DISCBuiltinTopicParticipantData ***participant_data_seq,
80         struct DDS_SampleInfo ***info_seq,
81         int *count);
82
87 extern CDSDllExport void CDSParticipantDetector_return_loan(
88         CDSParticipantDetector *self,
89         struct DISCBuiltinTopicParticipantData **participant_data_seq,
90         struct DDS_SampleInfo **info_seq,
91         int count);
92
99 extern CDSDllExport DDS_Boolean CDSParticipantAnnouncer_set_destinations(
100         CDSParticipantAnnouncer *self,
101         const struct PRESLocatorQosPolicy *locatorQos);
102
107 extern CDSDllExport DDS_Boolean CDSParticipantAnnouncer_write(
108         CDSParticipantAnnouncer *self,
109         struct DISCBuiltinTopicParticipantData *participant_data);
110
115 extern CDSDllExport DDS_Boolean CDSParticipantAnnouncer_dispose(
116         CDSParticipantAnnouncer *self,
117         const DDS_InstanceHandle_t *participant_handle);
118
119
120 #ifdef __cplusplus
121 } /* extern "C" */
122 #endif
123
124 #endif /* clouddiscoveryservice_builtin_h */
```

7.2 clouddiscoveryservice_dll.h

```
1 /*
2  * Copyright (c) 2016-2025 Real-Time Innovations, Inc. All rights reserved.
3  * No duplications, whole or partial, manual or electronic, may be made
4  * without express written permission.  Any such copies, or revisions thereof,
5  * must display this notice unaltered.
6  * This code contains trade secrets of Real-Time Innovations, Inc.
7  */
8
9 #ifndef clouddiscoveryservice_dll_h
10 #define clouddiscoveryservice_dll_h
11
12
13 #if defined(RTI_WIN32) || defined(RTI_WINCE) || defined(RTI_INTIME)
14   #if defined(RTI_clouddiscoveryservice_core_DLL_EXPORT)
15     #define CDSDllExport __declspec(dllexport)
16   #else
17     #define CDSDllExport
18   #endif /* RTI_clouddiscoveryservice_core_DLL_EXPORT */
19
20   #if defined(RTI_clouddiscoveryservice_core_DLL_VARIABLE)
21     #if defined(RTI_clouddiscoveryservice_core_DLL_EXPORT)
22       #define CDSDllVariable __declspec(dllexport)
23     #else
```

```
24       #define CDSDllVariable __declspec(dllimport)
25     #endif /* RTI_clouddiscoveryservice_core_DLL_EXPORT */
26   #else
27     #define CDSDllVariable
28   #endif /* RTI_clouddiscoveryservice_core_DLL_VARIABLE */
29 #elif defined(RTI_VISIBILITY_HIDDEN)
30   #define CDSDllExport __attribute__((visibility("default")))
31   #define CDSDllVariable __attribute__((visibility("default")))
32 #else
33   #define CDSDllExport
34   #define CDSDllVariable
35 #endif /* RTI_WIN32 || RTI_WINCE || RTI_INTIME */
36
37
38 #endif /* clouddiscoveryservice_dll_h */
```

7.3 clouddiscoveryservice_infrastructure.h File Reference

Service infrastructure.

```
#include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
#include "ndds/ndds_c.h"
#include "core_version/core_version_version.h"
```

Data Structures

- struct **RTI_CDS_ForwarderEventProperty**

  Data structure to configure the forwarder resend event mechanism. It allows CDS to resend a new or update announcement so that discovery can occur quickly in a lossy WAN environment.

Macros

- #define **RTI_CDS_DomainListProperty_INITIALIZER**

  Allows all domain IDs for RTI Cloud Discovery Service.
- #define **RTI_CDS_TransportSelectionProperty_INITIALIZER**

  Empty transport selection list. Use value from XML configuration.

7.3.1 Detailed Description

Service infrastructure.

7.4 clouddiscoveryservice_infrastructure.h

Go to the documentation of this file.

```
1 /*
2  * Copyright (c) 2016-2025 Real-Time Innovations, Inc. All rights reserved.
3  * No duplications, whole or partial, manual or electronic, may be made
4  * without express written permission.  Any such copies, or revisions thereof,
5  * must display this notice unaltered.
6  * This code contains trade secrets of Real-Time Innovations, Inc.
7  */
8
9 #ifndef clouddiscoveryservice_infrastructure_h
10 #define clouddiscoveryservice_infrastructure_h
11
12
13 #ifndef cds_dll_h
14   #include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
15 #endif
16 #ifndef ndds_c_h
17   #include "ndds/ndds_c.h"
18 #endif
19 #ifndef core_version_version_h
20   #include "core_version/core_version_version.h"
21 #endif
22
23 #ifdef __cplusplus
24 extern "C" {
25 #endif
26
27 /*e \file
28     \brief Service infrastructure
29 */
30
31
32 #define RTI_CDS_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT(T) \
33     struct T; \
34     CDSDllExport void T##_initialize(struct T *self); \
35     CDSDllExport void T##_finalize(struct T *self); \
36     CDSDllExport struct T *T##_copy(struct T *self, const struct T *from)
37
38 #define RTI_CDS_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT_W_ERROR_CHECKING(T) \
39     struct T; \
40     CDSDllExport RTIBool T##_initialize(struct T *self); \
41     CDSDllExport void T##_finalize(struct T *self); \
42     CDSDllExport RTIBool T##_copy(struct T *self, const struct T *from)
43
44
45 /*e \ingroup RTI_CDS_ServiceProperty
46  *
47  * @brief Allows all domain IDs for @product.
48  *
49  */
50 #define RTI_CDS_DomainListProperty_INITIALIZER \
51     { \
52         NULL, NULL \
53     }
54
55 #define RTI_CDS_DOMAIN_LIST_PROPERTY_ALL_VALUE ("DOMAIN_LIST_ALL")
56 #define RTI_CDS_DOMAIN_LIST_PROPERTY_EMPTY_VALUE ("")
57
58 RTI_CDS_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT(RTI_CDS_DomainListProperty);
59
60 struct RTI_CDS_DomainListProperty {
61     /*e
62      * @brief The list of domain IDs where @product processes the incoming
63      * participant announcements.
64      *
65      * @default NULL. Use value from XML configuration.
66      */
67     char *allow_domain_ids;
68
69     /*e
70      * @brief The list of domain IDs to deny or ignore. Overrides
71      * \ref allow_domain_ids.
72      *
73      * @default NULL. Use value from XML configuration.
74      */
75     char *deny_domain_ids;
76 };
```

```
77
78 #define RTI_CDS_TRANSPORT_UNICAST_PROPERTY_ALIAS_LENGTH_MAX (128)
79
80 #define RTI_CDS_TRANSPORT_SELECTION_LENGTH_MAX (8)
81
82 #define RTI_CDS_TransportUnicastProperty_INITIALIZER \
83     { \
84         { '\0' }, 0 \
85     }
86
87 /*e \ingroup RTI_CDS_ServiceProperty
88  *
89  * @brief Empty transport selection list. Use value from XML configuration.
90  *
91  */
92 #define RTI_CDS_TransportSelectionProperty_INITIALIZER \
93     { \
94         { RTI_CDS_TransportUnicastProperty_INITIALIZER }, 0 \
95     }
96
97 RTI_CDS_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT(RTI_CDS_TransportSelectionProperty);
98
99 struct RTI_CDS_TransportUnicastProperty {
100     /*e
101      * @brief Transport alias name.
102      */
103     char alias[RTI_CDS_TRANSPORT_UNICAST_PROPERTY_ALIAS_LENGTH_MAX + 1];
104     /*e
105      * @brief Receive port.
106      */
107     int receive_port;
108 };
109
110 struct RTI_CDS_TransportSelectionProperty {
111     /*e
112      * @brief \ref RTI_CDS_TransportUnicastProperty sequence.
113      */
114     struct RTI_CDS_TransportUnicastProperty
115             transports[RTI_CDS_TRANSPORT_SELECTION_LENGTH_MAX];
116     /*e
117      * @brief Indicates how many slots in the array are used.
118      */
119     int count;
120 };
121
122 #define RTI_CDS_FLOW_CONTROLLER_JOBS_UNLIMITED (0)
123 #define RTI_CDS_FlowControllerProperty_INITIALIZER \
124     { \
125         RTI_CDS_FLOW_CONTROLLER_JOBS_UNLIMITED, 0, { 0, 0, 0 }, \
126                 DDS_DURATION_INFINITE_VALUE \
127     }
128
129 RTI_CDS_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT(RTI_CDS_FlowControllerProperty);
130
131 typedef enum {
132     RTI_CDS_NEW_ANNOUNCEMENT_FLOW_CONTROLLER_OUTPUT_SLOT,
133     RTI_CDS_UPDATE_ANNOUNCEMENT_FLOW_CONTROLLER_OUTPUT_SLOT,
134     RTI_CDS_REFRESH_ANNOUNCEMENT_FLOW_CONTROLLER_OUTPUT_SLOT,
135     RTI_CDS_FlowControllerOutputSlotKind_COUNT
136 } RTI_CDS_FlowControllerOutputSlotKind;
137
138 struct RTI_CDS_FlowControllerProperty {
139     RTI_UINT64 output_jobs_per_second;
140     RTI_UINT64 max_burst_jobs;
141     RTI_INT32 output_slot_capacity_allocation
142             [RTI_CDS_FlowControllerOutputSlotKind_COUNT];
143     struct DDS_Duration_t flush_period;
144 };
145
151 struct RTI_CDS_ForwarderEventProperty {
152     /* Indicates how many times CDS resends a new or change announcement. */
153     DDS_UnsignedLong new_or_change_participant_announcements;
154     /* Minimum duration for the random interval between a resend. */
155     struct DDS_Duration_t min_new_or_change_participant_announcements_period;
156     /* Maximum duration for the random interval between a resend. */
157     struct DDS_Duration_t max_new_or_change_participant_announcements_period;
158 };
159
160 #define RTI_CDS_ForwarderEventProperty_INITIALIZER \
161     { \
162         0, /* new_or_change_participant_announcements */ \
```

```
163                 { 0, \
164                   0 }, /* min_new_or_change_participant_announcements_period \
165                         */ \
166         { \
167             0, 0 \
168         } /* max_new_or_change_participant_announcements_period */ \
169     }
170
171 DDSC_VARIABLE_LENGTH_VALUE_TYPE_SUPPORT(RTI_CDS_ForwarderEventProperty);
172
173
174 #ifdef __cplusplus
175 } /* extern "C" */
176 #endif
177
178 #endif /* clouddiscoveryservice_infrastructure_h */
```

7.5 clouddiscoveryservice_log.h File Reference

Logging facilities for RTI Cloud Discovery Service.

```
#include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
#include "log/log_common.h"
```

Macros

- #define **CDS_SUBMODULE_MASK_COMMON** (0x0002)
- #define **CDS_SUBMODULE_MASK_CONFIG** (0x0004)
- #define **CDS_SUBMODULE_MASK_ADAPTERS** (0x0008)
- #define **CDS_SUBMODULE_MASK_PROCESSORS** (0x0010)
- #define **CDS_SUBMODULE_MASK_MONITORING** (0x0012)
- #define **CDS_SUBMODULE_MASK_SERVICE** (0x0020)
- #define **CDS_SUBMODULE_MASK_SERVICEAPP** (0x0040)
- #define **CDS_SUBMODULE_MASK_SERVICELIB** (0x0060)
- #define **CDS_SUBMODULE_MASK_ALL** RTI_LOG_SUBMODULE_MASK_ALL

7.5.1 Detailed Description

Logging facilities for RTI Cloud Discovery Service.

7.6 clouddiscoveryservice_log.h

Go to the documentation of this file.

```
1 /*
2  * Copyright (c) 2016-2025 Real-Time Innovations, Inc. All rights reserved.
3  * No duplications, whole or partial, manual or electronic, may be made
4  * without express written permission.  Any such copies, or revisions thereof,
5  * must display this notice unaltered.
6  * This code contains trade secrets of Real-Time Innovations, Inc.
7  */
8
9 #ifndef clouddiscoveryservice_log_h
```

```
10 #define clouddiscoveryservice_log_h
11
12
13 #ifndef cds_dll_h
14   #include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
15 #endif
16 #ifndef log_common_h
17   #include "log/log_common.h"
18 #endif
19
20 #ifdef __cplusplus
21 extern "C" {
22 #endif
23
24 /*e \file
25   @brief Logging facilities for @product
26 */
27
28
29 /*e \defgroup CDSLogModule Logging
30   \ingroup CDS
31   @brief Provides control over the logging facilities for @product.
32 */
33
34
35 /*e \ingroup CDSLogModule
36   Mask for turning on logging for only the common submodule
37 */
38 #define CDS_SUBMODULE_MASK_COMMON (0x0002)
39 /*e \ingroup CDSLogModule
40   Mask for turning on logging for only the config submodule
41 */
42 #define CDS_SUBMODULE_MASK_CONFIG (0x0004)
43 /*e \ingroup CDSLogModule
44   Mask for turning on logging for only the adapter submodule
45 */
46 #define CDS_SUBMODULE_MASK_ADAPTERS (0x0008)
47
48 /*e \ingroup CDSLogModule
49   Mask for turning on logging for only the processor submodule
50 */
51 #define CDS_SUBMODULE_MASK_PROCESSORS (0x0010)
52
53 /*e \ingroup CDSLogModule
54   Mask for turning on logging for only the monitoring submodule
55 */
56 #define CDS_SUBMODULE_MASK_MONITORING (0x0012)
57
58 /*e \ingroup CDSLogModule
59   Mask for turning on logging for only the service submodule
60 */
61 #define CDS_SUBMODULE_MASK_SERVICE (0x0020)
62
63 /*e \ingroup CDSLogModule
64   Mask for turning on logging for only the service app submodule
65 */
66 #define CDS_SUBMODULE_MASK_SERVICEAPP (0x0040)
67
68 /*e \ingroup CDSLogModule
69   Mask for turning on logging for only the service lib submodule
70 */
71 #define CDS_SUBMODULE_MASK_SERVICELIB (0x0060)
72
73 /*e \ingroup CDSLogModule
74   Mask for turning on logging for all submodules
75 */
76 #define CDS_SUBMODULE_MASK_ALL RTI_LOG_SUBMODULE_MASK_ALL
77
78
79 #ifdef __cplusplus
80 } /* extern "C" */
81 #endif
82
83 #endif /* clouddiscoveryservice_log_h */
```

7.7 clouddiscoveryservice_service.h File Reference

Implements the RTI Cloud Discovery Service.

```
#include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
#include "clouddiscoveryservice/clouddiscoveryservice_infrastructure.h"
#include "routingservice/routingservice_infrastructure.h"
#include "ndds/ndds_c.h"
#include "core_version/core_version_version.h"
#include "ndds/ndds_config_c.h"
```

Data Structures

- struct **RTI_CDS_Property**

  Configuration property for RTI Cloud Discovery Service.
- struct **RTI_CDS_Service**

  RTI Cloud Discovery Service.

Functions

- DDS_Boolean **RTI_CDS_Property_initialize** (struct **RTI_CDS_Property** *self)

  *Initialize an instance of the **Service Property** (p. 13) variable with empty values.*
- struct **RTI_CDS_Property** * **RTI_CDS_Property_copy** (struct **RTI_CDS_Property** *to, const struct **RTI_↩ CDS_Property** *from)

  *Copy the values of a **Service Property** (p. 13) variable into another.*
- void **RTI_CDS_Property_finalize** (struct **RTI_CDS_Property** *self)

  *Finalize an instance of the **Service Property** (p. 13) variable.*
- struct **RTI_CDS_Service** * **RTI_CDS_Service_new** (const struct **RTI_CDS_Property** *property)

  Create a new RTI Cloud Discovery Service instance.
- void **RTI_CDS_Service_delete** (struct **RTI_CDS_Service** *self)

  Stop and delete an RTI Cloud Discovery Service instance.
- DDS_Boolean **RTI_CDS_Service_start** (struct **RTI_CDS_Service** *self)

  Start RTI Cloud Discovery Service.
- DDS_Boolean **RTI_CDS_Service_stop** (struct **RTI_CDS_Service** *self)

  Stop RTI Cloud Discovery Service.
- DDS_Boolean **RTI_CDS_Service_finalize_globals** (void)
- DDS_Boolean **RTI_CDS_Service_is_started** (struct **RTI_CDS_Service** *self)

  Query whether this RTI Cloud Discovery Service is currently started.
- DDS_Boolean **RTI_CDS_Service_update_rtps_psk_secret_passphrase** (struct **RTI_CDS_Service** *self, const char *rtps_psk_secret_passphrase)

  *Sets the dds.sec.crypto.rtps_psk_secret_passphrase property for the RTI Cloud Discovery Service instance started with **RTI_CDS_Service_start** (p. 18).*
- void **RTI_CDS_Logger_log** (NDDS_Config_LogLevel log_level, const char *format,...)

  Logs a message with the specified level within the running instance of RTI Cloud Discovery Service. The message will be printed under the log of the currently executing RTI Cloud Discovery Service, if the instance was started with `log_level` or higher verbosity.

Variables

- const int **RTI_CDS_LOG_VERBOSITY_DEBUG**

  Log verbosity level: exceptions + warnings + info + periodic + content.
- const int **RTI_CDS_LOG_VERBOSITY_ALL**

  Log verbosity level: exceptions + warnings + info + periodic.
- const int **RTI_CDS_LOG_VERBOSITY_INFO**

  Log verbosity level: exceptions + warnings + info.
- const int **RTI_CDS_LOG_VERBOSITY_WARNINGS**

  Log verbosity level: exceptions + warnings.
- const int **RTI_CDS_LOG_VERBOSITY_EXCEPTIONS**

  Log verbosity level: exceptions.
- const int **RTI_CDS_LOG_VERBOSITY_SILENT**

  Log verbosity level: silent.

7.7.1 Detailed Description

Implements the RTI Cloud Discovery Service.

7.8 clouddiscoveryservice_service.h

Go to the documentation of this file.

```
1 /*
2  * Copyright (c) 2016-2025 Real-Time Innovations, Inc. All rights reserved.
3  * No duplications, whole or partial, manual or electronic, may be made
4  * without express written permission.  Any such copies, or revisions thereof,
5  * must display this notice unaltered.
6  * This code contains trade secrets of Real-Time Innovations, Inc.
7  */
8
9 #ifndef clouddiscoveryservice_service_h
10 #define clouddiscoveryservice_service_h
11
12
13 #ifndef clouddiscoveryservice_dll_h
14   #include "clouddiscoveryservice/clouddiscoveryservice_dll.h"
15 #endif
16 #ifndef clouddiscoveryservice_infrastructure_h
17   #include "clouddiscoveryservice/clouddiscoveryservice_infrastructure.h"
18 #endif
19 #ifndef routingservice_infrastructure_h
20   #include "routingservice/routingservice_infrastructure.h"
21 #endif
22 #ifndef ndds_c_h
23   #include "ndds/ndds_c.h"
24 #endif
25 #ifndef core_version_version_h
26   #include "core_version/core_version_version.h"
27 #endif
28 #ifndef ndds_config_c_h
29   #include "ndds/ndds_config_c.h"
30 #endif
31
32 #ifdef __cplusplus
33 extern "C" {
34 #endif
35
36 /*e \file
37     \brief Implements the @product.
38 */
39
```

```
40 extern CDSDllVariable const struct RTI_CDS_DomainListProperty
41         RTI_CDS_DOMAIN_LIST_DEFAULT;
42
43 extern CDSDllVariable const char *RTI_CDS_TRANSPORT_UNICAST_ALIAS_DEFAULT;
44
45 extern CDSDllVariable int RTI_CDS_TRANSPORT_UNICAST_RECEIVE_PORT_DEFAULT;
46
47 extern CDSDllVariable struct RTI_CDS_TransportSelectionProperty
48         RTI_CDS_TRANSPORT_SELECTION_DEFAULT;
49
50 extern CDSDllVariable const RTI_UINT64
51         CDS_FLOW_CONTROLLER_OUTPUT_BANDWITH_UNLIMITED;
52
53 extern CDSDllVariable const struct RTI_CDS_FlowControllerProperty
54         RTI_CDS_FLOW_CONTROLLER_PROPERTY_DEFAULT;
55
56 extern CDSDllVariable const struct RTI_CDS_ForwarderEventProperty
57         RTI_CDS_FORWARDER_EVENT_PROPERTY_DEFAULT;
58
59
60 #define RTI_CDS_FLOW_CONTROLLER_PROPERTY_DEFAULT_VALUE \
61     { \
62         RTI_CDS_FLOW_CONTROLLER_JOBS_UNLIMITED, \
63                 RTI_CDS_FLOW_CONTROLLER_JOBS_UNLIMITED, { 40, 30, 30 }, \
64                 DDS_DURATION_INFINITE_VALUE \
65     }
66
67 /*e \defgroup RTI_CDS_ServiceProperty Service Property
68  *
69  * \ingroup RTI_CDS_ServiceLibModule
70  *
71  * @brief Service Property for @product. Use this in conjunction with
72  * \ref RTI_CDS_ServiceLibModule, to configure @product in your code.
73  */
74
75 /*e \ingroup RTI_CDS_ServiceProperty
76  *
77  * @brief Configuration property for @product.
78  */
79 struct RTI_CDS_Property {
80     /*e
81      * @brief Path to an @product configuration file
82      *
83      * If not \c NULL, this file is loaded; otherwise,
84      * if \c cfg_strings is not \c NULL, that XML code is loaded
85      *
86      * \default NULL.
87      */
88     char *cfg_file;
89
90     /*e
91      * @brief XML configuration represented as strings
92      *
93      * An array of strings that altogether make up an
94      * XML document to configure @product. This parameter is used
95      * only if \ref cfg_file is \c NULL.
96      *
97      * For example:
98      *
99      * \code
100      * int MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE = 3;
101      * const char *
102      * MY_CLOUD_DISCOVERY_SERVICE_CFG[MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE] =
103      *       {"<dds>
104      *              <cloud_discovery_service name=\"MyServiceName\">"
105      *                ...
106      *              </cloud_discovery_service>
107      *        </dds>"};
108      *
109      * property.cfg_strings = MY_CLOUD_DISCOVERY_SERVICE_CFG;
110      * property.cfg_strings_count = MY_CLOUD_DISCOVERY_SERVICE_CFG_SIZE;
111      * \endcode
112      *
113      * The reason for using an array instead of one single string is to
114      * get around the limited size of literal strings in C.
115      * In general, if you create the XML string dynamically
116      * using one single string in the array,
117      * setting \ref cfg_strings_count to 1 is enough:
118      *
119      * \code
120      * property.cfg_strings = malloc(sizeof(char *));
```

```
121      * property.cfg_strings[0] = "<dds><cloud_discovery_service>...</dds>";
122      * property.cfg_strings_count = 1;
123      * \endcode
124      *
125      * If your target system doesn't support a file system, you can use
126      * XML strings to configure the service. To ease this process, a
127      * utility is shipped to generate a C string array from a text file.
128      * You can find this utility in
129      * [@product installation directory]/resource/perl/cStringifyFile.pl
130      *
131      * \default NULL.
132      */
133     const char **cfg_strings;
134
135     /*e
136      * @brief Size of the array \ref cfg_strings
137      *
138      * \default 0.
139      */
140     unsigned int cfg_strings_count;
141
142     /*e
143      * @brief The name of the @product instance to run
144      *
145      * This is the name used to find the
146      * \verbatim <cloud_discovery_service> \endverbatim
147      * XML tag in the configuration file; this is the default name that will
148      * be used to refer to this execution in remote administration and
149      * monitoring if application_name isn't specified.
150      *
151      * \default NULL (use built-in configuration)
152      */
153     char *service_name;
154
155     /*e
156      * @brief Assigns a name to the execution of the @product
157      *
158      * Remote commands and status information will refer to the RTI Cloud
159      * Discovery Service using this name.
160      *
161      * \default service_name if this value is NULL.
162      */
163     char *application_name;
164
165     /*e
166      * @brief Controls whether the service applies XSD validation to the loaded
167      * configuration.
168      *
169      * \default DDS_BOOLEAN_TRUE
170      */
171     DDS_Boolean enforce_xsd_validation;
172
173     /*e
174      * @brief Set it to true to avoid loading the standard files usually loaded
175      * by @product
176      *
177      * Only the configuration in \ref cfg_file or \ref cfg_strings will be
178      * loaded.
179      *
180      * \default DDS_BOOLEAN_FALSE
181      */
182     DDS_Boolean skip_default_files;
183
184     /*e
185      * @brief The verbosity of the service
186      *
187      * Values:
188      *  <ul>
189      *  <li> \ref RTI_CDS_LOG_VERBOSITY_SILENT
190      *  <li> \ref RTI_CDS_LOG_VERBOSITY_EXCEPTIONS
191      *  <li> \ref RTI_CDS_LOG_VERBOSITY_WARNINGS
192      *  <li> \ref RTI_CDS_LOG_VERBOSITY_INFO
193      *  <li> \ref RTI_CDS_LOG_VERBOSITY_ALL
194      *  <li> \ref RTI_CDS_LOG_VERBOSITY_DEBUG
195      *  </ul>
196      *
197      * \default RTI_CDS_LOG_VERBOSITY_EXCEPTIONS
198      */
199     int service_verbosity;
200
201     /*e
```

```
202      * @brief The verbosity of \ndds core libraries
203      *
204      * Values:
205      *  <ul>
206      *  <li> \ref RTI_CDS_LOG_VERBOSITY_SILENT
207      *  <li> \ref RTI_CDS_LOG_VERBOSITY_EXCEPTIONS
208      *  <li> \ref RTI_CDS_LOG_VERBOSITY_WARNINGS
209      *  <li> \ref RTI_CDS_LOG_VERBOSITY_INFO
210      *  <li> \ref RTI_CDS_LOG_VERBOSITY_ALL
211      *  <li> \ref RTI_CDS_LOG_VERBOSITY_DEBUG
212      *  </ul>
213      *
214      * \default RTI_CDS_LOG_VERBOSITY_EXCEPTIONS
215      */
216     int dds_verbosity;
217
218     /*e
219      * @brief Specifies the transport and receive ports used to receive and
220      * send discovery traffic. Specifically, a receive resource is uniquely
221      * identified by a transport class-receive port pair.
222      * Overrides XML configuration if specified.
223      */
224     struct RTI_CDS_TransportSelectionProperty transport_selection;
225
226     /*e
227      * @brief Specifies the set of domain IDs the service will accept
228      *        announcements from.
229      *
230      * \default Unspecified. Use value from XML configuration.
231      */
232     struct RTI_CDS_DomainListProperty domain_list;
233
234     /*e
235      * @brief Set this to true to enable remote administration
236      * or false to disable it.
237      *
238      * \default DDS_BOOLEAN_FALSE
239      */
240     DDS_Boolean enable_administration;
241
242     /*e
243      * @brief If \ref enable_administration is true, this is
244      * the domain ID to use for remote administration.
245      *
246      * Takes precedence over the XML configuration.
247      * If \ref enable_administration is false, this value is not used
248      * even if remote administration is enabled in the XML configuration.
249      *
250      * \default 0
251      */
252     int administration_domain_id;
253
254     /*e
255      * @brief Set this to true to enable monitoring or false to disable it.
256      *
257      * \default DDS_BOOLEAN_FALSE
258      */
259     DDS_Boolean enable_monitoring;
260
261     /*e
262      * @brief If \ref enable_monitoring is true, this is
263      * the domain ID to use for publishing monitoring topics.
264      *
265      * Takes precedence over the XML configuration.
266      * If \ref enable_monitoring is false, this value is not used
267      * even if monitoring is enabled in the XML configuration.
268      *
269      * \default 0
270      */
271     int monitoring_domain_id;
272
273     /*e
274      * @brief Path to an @product license file
275      *
276      * If not \c NULL, this file is checked for a valid license; otherwise,
277      * default location will be used. This parameter is only used if your
278      * installation requires a license file.
279      *
280      * \default NULL.
281      */
282     char *license_file_name;
```

```
283
284     /*e
285      * @brief Dictionary of user variables.
286      * The dictionary provides a parallel way to expand XML configuration
287      * variables in the form $(my_var), when they are not defined in the
288      * environment.
289      *
290      * \default empty
291      */
292     struct RTI_RoutingServiceProperties user_environment;
293 };
294
295 #define RTI_CDS_Property_INITIALIZER \
296     { \
297         NULL,                       /* cfg_file */ \
298                 NULL,               /* cfg_strings */ \
299                 0,                  /* cfg_strings_count */ \
300                 NULL,               /* service_name */ \
301                 NULL,               /* application_name */ \
302                 DDS_BOOLEAN_FALSE, /* enforce_xsd_validation */ \
303                 DDS_BOOLEAN_FALSE, /* skip_default_files */ \
304                 0,                  /* service_verbosity */ \
305                 0,                  /* dds_verbosity */ \
306                 RTI_CDS_TransportSelectionProperty_INITIALIZER, /* transport_selection \
307                                                                    */ \
308                 RTI_CDS_DomainListProperty_INITIALIZER, /* domain_list */ \
309                 0,    /* enable_administration */ \
310                 0,    /* administration_domain_id */ \
311                 0,    /* enable_monitoring */ \
312                 0,    /* monitoring_domain_id */ \
313                 NULL, /* license_file_name */ \
314         { \
315             NULL, 0, 1 \
316         } /* user_environment */ \
317     }
318
319 extern CDSDllVariable const struct RTI_CDS_Property RTI_CDS_PROPERTY_DEFAULT;
320
331 extern CDSDllExport DDS_Boolean
332 RTI_CDS_Property_initialize(struct RTI_CDS_Property *self);
333
349 extern CDSDllExport struct RTI_CDS_Property *RTI_CDS_Property_copy(
350         struct RTI_CDS_Property *to,
351         const struct RTI_CDS_Property *from);
352
362 extern CDSDllExport void RTI_CDS_Property_finalize(
363         struct RTI_CDS_Property *self);
364
365 /*e \defgroup RTI_CDS_ServiceLibModule Library API
366  * \ingroup CDS
367  * @brief @product can be deployed as a C library linked into your application
368  * on select architectures.
369  *
370  * This API allows you to create, configure and start @product instances from
371  * your application.
372  *
373  * The following code shows the typical use of the API:
374  *
375  * \code
376  *
377  * struct RTI_CDS_Property property = RTI_CDS_Property_INITIALIZER;
378  * struct RTI_CDS_Service *service = NULL;
379  *
380  * property.cfg_file = "my_cloud_discovery_service_cfg.xml";
381  * property.service_name = "my_cloud_discovery_service";
382  * ...
383  *
384  * service = RTI_CDS_Service_new(&property);
385  * if (service == NULL) {
386  *    printf("Error ...");
387  *    return -1;
388  * }
389  *
390  * if (!RTI_CDS_Service_start(service)) {
391  *    printf("Error ...");
392  *    RTI_CDS_Service_delete(service);
393  *    return -1;
394  * }
395  *
396  * while (keep_running) {
397  *     sleep();
```

```
398  *     ...
399  * }
400  *
401  * RTI_CDS_Service_delete(service);
402  *
403  * return 0;
404  *
405  * \endcode
406  *
407  * Instead of a file, you can use XML strings to configure @product.
408  * See \ref RTI_CDS_ServiceProperty for more information.
409  * <p>
410  * To build your application you need to link with the @product library
411  * in <b> &lt;@ndds home&gt;/bin/&lt;architecture&gt;/ </b>
412  *
413  * An example is provided at
414  *
       <b>https://github.com/rticommunity/rticonnextdds-examples/tree/master/examples/cloud_discovery_service/library_api</b>.
415  *
416  * ### Development Requirements
417  *
418  * |                                | Linux Systems                  | Windows
419  * Systems               | | ---------------------------: |
420  * :----------------------------: | :----------------------------: | | Link
421  * Libraries (Compile Time) | librticlouddiscoveryservice.so |
422  * rticlouddiscoveryservice.dll  | | Shared Libraries (Run Time)   |
423  * librtiroutingservice.so        | rtiroutingservice.dll         | | ^ |
424  * librtidlc.so                   | rtidlc.dll                    | | ^ |
425  * librticonnextmsgc.so           | rticonnextmsgc.dll            | | ^ |
426  * libnddsmetp.so                 | nddsmetp.dll                  | | ^ |
427  * librtiapputilsc.so             | rtiapputilsc.dll              | | ^ |
428  * librtixml2.so                  | rtixml2.dll                   | | ^ |
429  * libnddsc.so                    | nddsc.dll                     | | ^ |
430  * libnddscore.so                 | nddscore.dll                  | | Static
431  * Libraries              | librticlouddiscoveryservicez.a |
432  * rticlouddiscoveryservicez.lib | | ^                            |
433  * librtiroutingservicez.a        | rtiroutingservicez.lib        | | ^ |
434  * librtidlcz.a                   | rtidlcz.lib                   | | ^ |
435  * librticonnextmsgcz.a           | rticonnextmsgcz.lib           | | ^ |
436  * libnddsmetpz.a                 | nddsmetpz.lib                 | | ^ |
437  * librtiapputilscz.a             | rtiapputilscz.lib             | | ^ |
438  * librtixml2z.a                  | rtixml2z.lib                  | | ^ |
439  * libnddscz.a                    | nddscz.lib                    | | ^ |
440  * libnddscorez.a                 | nddscorez.lib                 | | Headers |
441  * clouddiscoveryservice/clouddiscoveryservice_service.h | |
442  *
443  * <b>NOTE:</b> If you are using debug libraries, remember to add the 'd' suffix
444  * to the library name. For shared libraries on macOS systems, the library name
445  * is the same as that for Linux except the extension is '.dylib' instead of
446  * '.so'.
447  */
448
449 #ifdef DOXYGEN_DOCUMENTATION_ONLY
455 struct RTI_CDS_Service { };
456 #endif
457
458 struct RTI_CDS_Service;
459
475 extern CDSDllExport struct RTI_CDS_Service *RTI_CDS_Service_new(
476         const struct RTI_CDS_Property *property);
477
489 extern CDSDllExport void RTI_CDS_Service_delete(struct RTI_CDS_Service *self);
490
504 extern CDSDllExport DDS_Boolean
505 RTI_CDS_Service_start(struct RTI_CDS_Service *self);
506
521 extern CDSDllExport DDS_Boolean
522 RTI_CDS_Service_stop(struct RTI_CDS_Service *self);
523
532 extern CDSDllExport DDS_Boolean RTI_CDS_Service_finalize_globals(void);
533
539 extern CDSDllExport DDS_Boolean
540 RTI_CDS_Service_is_started(struct RTI_CDS_Service *self);
541
558 extern CDSDllExport DDS_Boolean
559 RTI_CDS_Service_update_rtps_psk_secret_passphrase(
560         struct RTI_CDS_Service *self,
561         const char *rtps_psk_secret_passphrase);
562
563 /*e \ingroup CDSLogModule
564  * @brief Logs a message with the specified level within the running instance of
```

```
565  * @product. The message will be printed under the log of the currently
566  * executing
567  * @product, if the instance was started with \p log_level or higher verbosity.
568  *
569  * The message is specified with a format and a format parameter, in a similar
570  * fashion to the standard C printf() operation.
571  *
572  * The generated log will be part of logging stream of the running
573  * @product, if the \p log_level is part of the configured verbosity.
574  * The result log message may include additional information according to the
575  * Connext logging configuration, such as Advlog Context, thread ID, line
576  * number, etc. Additionally, the result log message will contain a newline
577  * character at the end, so the format does not need to contain it.
578  *
579  * @param [in] log_level Log level associated to the message
580  * @param [in] format message format specification
581  * @param [in] ... variablelength argument (stdarg)
582  */
583 extern CDSDllExport void RTI_CDS_Logger_log(
584         NDDS_Config_LogLevel log_level,
585         const char *format,
586         ...);
587
588 /*e \ingroup CDSLogModule
589  * @brief Log verbosity level: exceptions + warnings + info + periodic + content
590  */
591 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_DEBUG;
592
593 /*e \ingroup CDSLogModule
594  * @brief Log verbosity level: exceptions + warnings + info + periodic
595  */
596 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_ALL;
597
598 /*e \ingroup CDSLogModule
599  * @brief Log verbosity level: exceptions + warnings + info
600  */
601 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_INFO;
602
603 /*e \ingroup CDSLogModule
604  * @brief Log verbosity level: exceptions + warnings
605  */
606 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_WARNINGS;
607
608 /*e \ingroup CDSLogModule
609  * @brief Log verbosity level: exceptions
610  */
611 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_EXCEPTIONS;
612
613 /*e \ingroup CDSLogModule
614  * @brief Log verbosity level: silent
615  */
616 extern CDSDllVariable const int RTI_CDS_LOG_VERBOSITY_SILENT;
617
618 extern CDSDllExport void RTI_CDS_Service_print_available_configI(
619         struct RTI_CDS_Service *self);
620
621
622 #ifdef __cplusplus
623 } /* extern "C" */
624 #endif
625
626 #endif /* clouddiscoveryservice_service_h */
```

Index

administration_domain_id
    RTI_CDS_Property, 25
application_name
    RTI_CDS_Property, 23

CDS_SUBMODULE_MASK_ADAPTERS
    Logging, 10
CDS_SUBMODULE_MASK_ALL
    Logging, 11
CDS_SUBMODULE_MASK_COMMON
    Logging, 10
CDS_SUBMODULE_MASK_CONFIG
    Logging, 10
CDS_SUBMODULE_MASK_MONITORING
    Logging, 11
CDS_SUBMODULE_MASK_PROCESSORS
    Logging, 10
CDS_SUBMODULE_MASK_SERVICE
    Logging, 11
CDS_SUBMODULE_MASK_SERVICEAPP
    Logging, 11
CDS_SUBMODULE_MASK_SERVICELIB
    Logging, 11
cfg_file
    RTI_CDS_Property, 22
cfg_strings
    RTI_CDS_Property, 22
cfg_strings_count
    RTI_CDS_Property, 23
clouddiscoveryservice_builtin.h, 29
clouddiscoveryservice_dll.h, 30
clouddiscoveryservice_infrastructure.h, 31, 32
clouddiscoveryservice_log.h, 34
clouddiscoveryservice_service.h, 36, 37

dds_verbosity
    RTI_CDS_Property, 24
domain_list
    RTI_CDS_Property, 25

enable_administration
    RTI_CDS_Property, 25
enable_monitoring
    RTI_CDS_Property, 26
enforce_xsd_validation
    RTI_CDS_Property, 24

Library API, 16
    RTI_CDS_Service_delete, 18
    RTI_CDS_Service_finalize_globals, 19
    RTI_CDS_Service_is_started, 19
    RTI_CDS_Service_new, 17
    RTI_CDS_Service_start, 18
    RTI_CDS_Service_stop, 19
    RTI_CDS_Service_update_rtps_psk_secret_passphrase, 19
license_file_name
    RTI_CDS_Property, 26
Logging, 9
    CDS_SUBMODULE_MASK_ADAPTERS, 10
    CDS_SUBMODULE_MASK_ALL, 11
    CDS_SUBMODULE_MASK_COMMON, 10
    CDS_SUBMODULE_MASK_CONFIG, 10
    CDS_SUBMODULE_MASK_MONITORING, 11
    CDS_SUBMODULE_MASK_PROCESSORS, 10
    CDS_SUBMODULE_MASK_SERVICE, 11
    CDS_SUBMODULE_MASK_SERVICEAPP, 11
    CDS_SUBMODULE_MASK_SERVICELIB, 11
    RTI_CDS_LOG_VERBOSITY_ALL, 12
    RTI_CDS_LOG_VERBOSITY_DEBUG, 12
    RTI_CDS_LOG_VERBOSITY_EXCEPTIONS, 13
    RTI_CDS_LOG_VERBOSITY_INFO, 12
    RTI_CDS_LOG_VERBOSITY_SILENT, 13
    RTI_CDS_LOG_VERBOSITY_WARNINGS, 13
    RTI_CDS_Logger_log, 11

monitoring_domain_id
    RTI_CDS_Property, 26

RTI_CDS_DomainListProperty_INITIALIZER
    Service Property, 14
RTI_CDS_ForwarderEventProperty, 21
RTI_CDS_LOG_VERBOSITY_ALL
    Logging, 12
RTI_CDS_LOG_VERBOSITY_DEBUG
    Logging, 12
RTI_CDS_LOG_VERBOSITY_EXCEPTIONS
    Logging, 13
RTI_CDS_LOG_VERBOSITY_INFO
    Logging, 12
RTI_CDS_LOG_VERBOSITY_SILENT
    Logging, 13
RTI_CDS_LOG_VERBOSITY_WARNINGS

Logging, 13
RTI_CDS_Logger_log
Logging, 11
RTI_CDS_Property, 21
administration_domain_id, 25
application_name, 23
cfg_file, 22
cfg_strings, 22
cfg_strings_count, 23
dds_verbosity, 24
domain_list, 25
enable_administration, 25
enable_monitoring, 26
enforce_xsd_validation, 24
license_file_name, 26
monitoring_domain_id, 26
service_name, 23
service_verbosity, 24
skip_default_files, 24
transport_selection, 25
user_environment, 26
RTI_CDS_Property_copy
Service Property, 15
RTI_CDS_Property_finalize
Service Property, 15
RTI_CDS_Property_initialize
Service Property, 15
RTI_CDS_Service, 27
RTI_CDS_Service_delete
Library API, 18
RTI_CDS_Service_finalize_globals
Library API, 19
RTI_CDS_Service_is_started
Library API, 19
RTI_CDS_Service_new
Library API, 17
RTI_CDS_Service_start
Library API, 18
RTI_CDS_Service_stop
Library API, 19
RTI_CDS_Service_update_rtps_psk_secret_passphrase
Library API, 19
RTI_CDS_TransportSelectionProperty_INITIALIZER
Service Property, 14

Service Property, 13
RTI_CDS_DomainListProperty_INITIALIZER, 14
RTI_CDS_Property_copy, 15
RTI_CDS_Property_finalize, 15
RTI_CDS_Property_initialize, 15
RTI_CDS_TransportSelectionProperty_INITIALIZER, 14
service_name
RTI_CDS_Property, 23
service_verbosity
RTI_CDS_Property, 24
skip_default_files
RTI_CDS_Property, 24

transport_selection
RTI_CDS_Property, 25

user_environment
RTI_CDS_Property, 26